

**INFORMATICA  
APLICADA A LA  
CIENCIA Y A LA  
INDUSTRIA**

**LIC. EDUARDO  
BUSTOS  
FARIAS**



**1999-2000**

# INDICE

<b>INTRODUCCION.....</b>	<b>5</b>
<b>OBJETIVOS DEL CURSO.....</b>	<b>7</b>
<b>FORMA DE ACREDITACIÓN DEL CURSO.....</b>	<b>7</b>
<b>PROGRAMA DE ESTUDIOS DE LA ASIGNATURA DE: INFORMÁTICA APLICADA A LA CIENCIA Y LA INDUSTRIA.....</b>	<b>8</b>
CONTENIDO GENERAL DEL PROGRAMA.....	8
UNIDAD 1 DISEÑO GRAFICO.....	8
UNIDAD 2 PROGRAMACIÓN.....	8
UNIDAD 3 REDES LOCALES Y SERVICIOS DE RED.....	9
UNIDAD 4 LA COMPUTADORA Y EL ANÁLISIS ESTADÍSTICO.....	10
BIBLIOGRAFÍA.....	11
<b>CORELDRAW.....</b>	<b>12</b>
<b>VISUAL BASIC.....</b>	<b>36</b>
INTRODUCCIÓN.....	36
SU PRIMERA APLICACIÓN EN VISUAL BASIC.....	37
PRIMEROS PASOS CON VISUAL BASIC.....	37
ELEMENTOS DE LA INTERFAZ.....	38
PASOS PARA LA CREACIÓN DE UNA APLICACIÓN CON VISUAL BASIC.....	40
CREACIÓN DE LA INTERFAZ.....	41
DEFINICIÓN DE PROPIEDADES.....	43
ESCRITURA DE CÓDIGO.....	44
EJECUCIÓN DE LA APLICACIÓN.....	47
TRABAJAR CON APLICACIONES EJEMPLO.....	47
ANIMACIÓN SIMPLE.....	47
UN VISOR DE IMÁGENES.....	51
DISTRIBUCIÓN DE SUS APLICACIONES.....	55
<b>UNIDAD 2. CREACIÓN Y USO DE LOS CONTROLES.....</b>	<b>56</b>
CONTROLES DE VISUAL BASIC.....	56
CATEGORÍA DE CONTROLES.....	58
CONVENCIONES PARA NOMBRAR OBJETOS.....	60
USO DEL VALOR DE UN CONTROL.....	61
HACIENDO CLIC EN LOS BOTONES PARA EJECUTAR ACCIONES.....	62
USO DE LOS BOTONES DE COMANDO.....	63
UN CONTROL IMAGE PARA INTERACTUAR CON EL USUARIO.....	64
PRESENTACIÓN E INTRODUCCIÓN DE TEXTO.....	64
USO DE ETIQUETAS PARA PRESENTAR TEXTO.....	65
TRABAJAR CON CUADROS DE TEXTO.....	67
CONTROLES QUE PRESENTAN OPCIONES AL USUARIO.....	69
SELECCIÓN DE OPCIONES INDIVIDUALES CON CASILLAS DE VERIFICACIÓN.....	70
AGRUPAR OPCIONES CON OPTION BUTTONS.....	72
USO DE CUADRO DE LISTA Y CUADROS COMBINADOS.....	77
USO DE LAS BARRAS DE DESPLAZAMIENTO COMO DISPOSITIVOS DE ENTRADA.....	91
OBTENER Y ESTABLECER EL ENFOQUE DE UN OBJETO.....	96
DEFINICIÓN DEL ORDEN DE TABULACION.....	97
ELIMINAR UN CONTROL DEL ORDEN DE TABULACIÓN.....	99

HABILITACIÓN, DESHABILITACIÓN Y CONTROL DE LA VISIBILIDAD EN TIEMPO DE EJECUCIÓN .....	99
<b>ADMINISTRACIÓN DE PROYECTOS.....</b>	<b>101</b>
COMPRESIÓN DE LOS PROYECTOS.....	101
VENTANA PROYECTO.....	101
EL ARCHIVO DE PROYECTO.....	102
ELEMENTOS DE UN PROYECTO.....	102
<i>Módulos estándar</i> .....	102
ADICIÓN, ELIMINACIÓN Y ALMACENAMIENTO DE ARCHIVOS.....	105
COMBINACIÓN DE TEXTO.....	106
<b>FUNDAMENTOS DE PROGRAMACIÓN.....</b>	<b>107</b>
ESTRUCTURA DE UNA APLICACIÓN VISUAL BASIC.....	107
COMO FUNCIONA UNA APLICACIÓN CONTROLADA POR EVENTOS.....	107
VARIABLES.....	110
ALMACENAMIENTO Y RECUPERACIÓN DE DATOS EN VARIABLES.....	111
TIPOS DE DATOS.....	111
MÓDULOS.....	112
MÓDULOS DE FORMULARIO.....	112
MÓDULOS ESTÁNDAR.....	113
MÓDULOS DE CLASE.....	113
PROCEDIMIENTOS.....	113
PROCEDIMIENTOS SUB.....	114
PROCEDIMIENTOS FUNCIÓN.....	116
PROCEDIMIENTOS PROPERTY.....	117
CREACIÓN DE PROCEDIMIENTOS NUEVOS.....	126
SELECCIÓN DE PROCEDIMIENTOS EXISTENTES.....	126
LLAMADA A PROCEDIMIENTOS.....	127
TRANSFERENCIA DE ARGUMENTOS A PROCEDIMIENTOS.....	131
ESTRUCTURAS DE CONTROL.....	136
ESTRUCTURAS DE DECISIÓN.....	136
ESTRUCTURAS DE BUCLES.....	140
ESTRUCTURAS DE CONTROL ANIDADAS.....	144
SALIDA DE UNA ESTRUCTURA DE CONTROL.....	145
SALIDA DE UN PROCEDIMIENTO SUB O FUNCTION.....	146
DIVISIÓN DE UNA INSTRUCCIÓN EN MÚLTIPLES LÍNEAS.....	147
ADICIÓN DE COMENTARIOS AL CÓDIGO.....	147
COMPRESIÓN DE SISTEMAS DE NUMERACIÓN.....	148
CONVENCIÓN DE NOMBRES EN VISUAL BASIC.....	148
<b>REDES Y COMUNICACIONES ELECTRONICAS.....</b>	<b>150</b>
<b>INTERNET.....</b>	<b>168</b>
FUNDAMENTOS DEL EXPLORADOR.....	168
CONECTARSE.....	183
<b>COREL CHART.....</b>	<b>207</b>
<b>CREACIÓN DE GRÁFICOS.....</b>	<b>207</b>
<b>INTRODUCCIÓN DE INFORMACIÓN EN EL GESTOR DE DATOS.....</b>	<b>208</b>
<b>UTILIZACIÓN DE PICTOGRAMAS EN LOS GRÁFICOS.....</b>	<b>209</b>
<b>INTERCAMBIO DE INFORMACIÓN CON OTRAS APLICACIONES.....</b>	<b>209</b>

EXPORTACIÓN DE GRÁFICOS PARA SU UTILIZACIÓN EN OTROS PROGRAMAS .....	210
IMPORTACIÓN DE GRÁFICOS DE OTROS PROGRAMAS .....	211
COPIADO DE GRÁFICOS E IMÁGENES EN EL PORTAPAPELES DE WINDOWS.....	211
VINCULACIÓN DE GRÁFICOS CON ARCHIVOS DE CORELDRAW .....	212
ESTABLECIMIENTO DE VÍNCULOS DE INTERCAMBIO DINÁMICO CON HOJAS DE CÁLCULO .....	212
HERRAMIENTAS DE CORELCHART .....	212
EL BOTÓN GESTOR DE DATOS.....	213
HERRAMIENTA SELECCIÓN .....	213
HERRAMIENTA DE ACCESO RÁPIDO .....	213
HERRAMIENTA ZOOM .....	214
HERRAMIENTA LÁPIZ.....	214
HERRAMIENTA RECTÁNGULO .....	214
HERRAMIENTA ELIPSE .....	215
HERRAMIENTA TEXTO .....	215
HERRAMIENTA FILETE .....	216
LA HERRAMIENTA RELLENO .....	217
PERSIANA RELLENO .....	218
PERSIANA DE HERRAMIENTA DE VISIÓN EN 3D.....	221
VENTANA DE DIÁLOGO PATRÓN A TODO COLOR .....	222
<b>STADISTIC .....</b>	<b>224</b>
<i>Introduction</i> .....	224
<i>What are observations</i> .....	224
<i>What are Data ?</i> .....	225
<i>Data are coded observations</i> .....	225
<i>Data are coded observations</i> .....	225
<i>Statistics studies statistical techniques</i> .....	225
VARIABLES .....	225
WHAT IS A VARIABLE? (STATISTICAL VARIABLES).....	225
RANDOM VARIABLES .....	226
MODELING SAMPLES WITH RANDOM VARIABLES .....	227
<i>Conventional</i> .....	227
<i>Robustness</i> .....	227
THE VARIANCE .....	227
<i>Discussion</i> .....	227
<i>Definition</i> .....	227
SAMPLING .....	227
POPULATIONS AND SAMPLES .....	228
FINITE POPULATONS .....	228
INFINITE POPULATIONS .....	228
PARAMETERS AND STATISTICS.....	229
SAMPLING FROM INFINITE POPULATONS .....	229
SAMPLING FROM FINITE POPULATONS .....	229
SAMPLING DESIGN .....	230
SAMPLING WITH AND WITHOUT REPLACEMENT .....	230
SIMPLE RANDOM SAMPLING .....	230
SIMPLE RANDOM SAMPLING WITH AND WITHOUT REPLACEMENT .....	231
MODELS .....	231
MODELING POPULATIONS .....	231
MODELING SAMPLES.....	231
INFERENCE .....	232
INDUCTIVE VERSUS DEDUCTIVE INFERENCE .....	232
INFERENCE STABILITY .....	232
ESTIMATION .....	233
WHAT IS A SAMPLING DISTRIBUTION?.....	233
CONFIDENCE INTERVALS .....	234

HYPOTHESIS TESTING .....	235
BASIC DEFINITIONS.....	237
<i>Contents</i> .....	237

## INTRODUCCION

El Centro Universitario del Valle se esfuerza continuamente por mejorar el nivel y la calidad de los cursos que imparte y del material didáctico que utiliza. Es elaborado bajo la política del colegio de contar con material actualizado que sirva para mejorar la calidad en la impartición de la enseñanza de todas y cada y una de las materia.

El presente manual no pretende sustituir los manuales del usuario de cada uno de los programas que se explican, mas bien sirve de complemento a la materia de Informática que ésta institución imparte como parte del plan de estudios del cuarto grado de preparatoria, según el plan de estudios de la UNAM, y ha sido elaborado de acuerdo a su programa académico. Se considera que el material puede ser útil para apoyar otros planes de capacitación en computación.

Los temas incluidos en este manual dan a conocer al alumno un panorama general y actual de la computación a la vez que presentan los elementos necesarios para un buen aprovechamiento de cursos posteriores.

Esta materia se sitúa en el sexto año del bachillerato. Es optativa y de carácter teórico-práctica.

Materias antecedentes: directamente con Introducción a la Informática en la que los alumnos fueron introducidos al manejo básico de un equipo de cómputo y su potencialidad. De manera indirecta esta materia fortalece sus bases con la lógica y las matemáticas principalmente.

Materias paralelas: Se relaciona desde luego con las matemáticas, la estadística y probabilidad, aquí planeamos el uso de la computadora a mitad del ciclo cuando los alumnos ya cuentan con las bases de estadística; De forma en general se relaciona con las ciencias naturales (física química y biología) en las que los instrumentos de medición de esta área están ahora automatizados, de lo que se desprende una mayor precisión, lo importante es que los alumnos manejen las tarjetas de interfase y sus dispositivos en la medición, no sólo de forma mecanizada sino que entiendan su funcionamiento para así poder resolver cualquier problema que se les pueda presentar..

Materias consecuentes: es una materia propedéutica y sus materias que le precedan serán las propias de la licenciatura, y estamos seguros de que con esta asignatura estamos fortaleciendo a los jóvenes del bachillerato prontos al ingreso a facultad.

Este curso tendrá como carácter principal el manejo y el análisis de la información que se genera en las ciencias naturales como Química, Física y Biología, de tal forma que los equipos de cómputo tengan un carácter integrador de los conocimientos. Mediante la optimización del manejo de la computadora los alumnos reducirán los niveles de abstracción necesarios en la comprensión de los fenómenos naturales a través de las modelos que pueden hacerse con estos

instrumentos. Así mismo, la interconexión de experimentos de laboratorio a las computadoras facilitará la obtención de datos, el control y análisis de los mismos. Dado el carácter teórico-práctico de la asignatura y teniendo una gran trascendencia el desarrollo práctico de la misma, se buscará reforzar ésta, mediante una serie de actividades de aprendizaje que llevará a los alumnos a destinar tiempo fuera de su horario de clase y acudir a los centros de cómputo.

Desde tiempos remotos el hombre en su afán de conocer y explicarse los fenómenos naturales y sociales ha generado grandes volúmenes de información, Parte de ésta, transformándose en dispositivos e instrumentos tecnológicos que le brindan supervivencia y comodidad; como parte de ellos están las computadoras, que ofrecen a los usuarios versatilidad, rapidez, precisión y calidad en el manejo de la información.

El propósito del curso es que el alumno pueda obtener, procesar y analizar la información de las ciencias naturales, así como manejar instrumentos de laboratorio de manera más precisa y fácil, dando como resultado un contexto más amplio en el que circunscriben estas áreas. El propósito general de la materia es que los alumnos incorporen nuevas tecnologías como herramientas en el aprendizaje, no sólo la computadora, sino también otro tipo de instrumentos que pueden conectarse a ellas como, microscopios, potenciómetros, termómetros, VCR, CCD.

México, D.F. a julio de 1999.  
Lic. Eduardo Bustos Farías.

## **OBJETIVOS DEL CURSO.**

La asignatura de Informática aplicada a la ciencia y a la industria contribuye a la construcción del perfil general del egresado de la siguiente manera, que el alumno

- Aprecie la vinculación existente entre las ciencias experimentales y el cómputo.
- Sea capaz de aplicar la metodología de la solución de problemas a situaciones concretas en aulas y laboratorios vía computadora.
- Identifique las ventajas y desventajas del empleo del equipo de cómputo en la metodología de la solución de problemas.
- Sea capaz de emplear algunos sensores de medición para señales analógicas y lleve a cabo la adquisición de datos vía computadora que le ayuden a diseñar pequeños experimentos que involucren algunos elementos del protocolo de investigación.
- Identifique las partes de la física y química que se involucran en estos procesos.
- Sea capaz de aplicar sus conocimientos en otras disciplinas para llevar a cabo el análisis de los datos experimentales, mediante el uso de software que facilite su tarea.
- Sea capaz de identificar conceptos básicos sobre redes y su aplicación.
- Visualice a la computadora como fuente de información a nivel mundial y sea capaz de utilizarla de manera racional
- Posea conocimientos, métodos y técnicas básicas inherentes a la informática.

## **FORMA DE ACREDITACIÓN DEL CURSO.**

### **a) Actividades o factores**

Prácticas en los laboratorios de creatividad y LACE.  
Prácticas.  
Realización de tareas.  
Exámenes.

### **b) Carácter de la actividad**

El carácter de las actividades de evaluación serán individuales.

### **c) Periodicidad.**

El proceso de evaluación será continuo, y la aplicación de los exámenes será por unidad.

### **d) Porcentaje sobre la calificación sugerido.**



Exámenes 30%  
Prácticas en computadora 50%  
Tareas 20%

**PROGRAMA DE ESTUDIOS DE LA ASIGNATURA DE: INFORMÁTICA  
APLICADA A LA CIENCIA Y LA INDUSTRIA**

AÑO ESCOLAR EN QUE SE IMPARTE: **SEXTO**  
CATEGORÍA DE LA ASIGNATURA: **PROPEDÉUTICA**  
CARÁCTER DE LA ASIGNATURA: **TEÓRICO-PRÁCTICA**

**CONTENIDO GENERAL DEL PROGRAMA.**

UNIDAD 1. DISEÑO GRAFICO.  
UNIDAD 2. PROGRAMACIÓN.  
UNIDAD 3. REDES LOCALES Y SERVICIOS DE RED.  
UNIDAD 4. LA COMPUTADORA Y EL ANÁLISIS ESTADÍSTICO.

**UNIDAD 1 DISEÑO GRAFICO.**

b) Propósitos de la unidad

Que el alumno pueda elaborar logotipos y diseños gráficos de calidad profesional.

HORAS	CONTENIDO	DESCRIPCIÓN DEL CONTENIDO	ESTRATEGIAS DIDÁCTICAS (actividades de aprendizaje)
10	Herramientas de diseño gráfico.  Efectos especiales.	Lápiz. Herramientas de relleno. Texturas. Filete  Mantener la copia durante el giro. Perspectiva. Siluetas	Utilizar el programa Corel Draw para diseñar graficos de alta calidad.

**UNIDAD 2 PROGRAMACIÓN**

b) Propósitos de la unidad

Que el alumno pueda crear métodos de solución de problemas específicos a través de la computadora

HORAS	CONTENIDO	DESCRIPCIÓN DEL CONTENIDO	ESTRATEGIAS DIDÁCTICAS (actividades de aprendizaje)
20	<p>VARIABLES Y TIPOS DE DATOS PRIMITIVOS</p> <p>Expresiones y operadores</p> <p>Subrutinas</p> <p>Tipos de datos no primitivos</p>	<p>VARIABLES GLOBALES Y LOCALES.</p> <p>Tipos de datos (numéricos, carácter, cadena de caracteres y lógicos).</p> <p>Alcance de una variable.</p> <p>Apunadores.</p> <p>Asignación .</p> <p>Operadores aritméticos, relacionales y lógicos .</p> <p>Prioridad de operadores y evaluación de expresiones</p> <p>Sentencias de ciclo</p> <p>Sentencias de selección</p> <p>Funciones y Procedimientos.</p> <p>VARIABLES CON SUBÍNDICE (ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES), REGISTROS Y ARCHIVOS</p>	<p>Utilizar diagramas de flujo o pseudocódigo.</p> <p>Selección de un lenguaje estructurado para convertir los diagramas de flujo y pseudocódigo a código de programación.</p> <p>Realización de programas para resolución de problemas de asignaturas con carácter científico</p> <p>Realizar ejercicios en pizarrón y papel</p> <p>Captura de código en editores y ejecución de programas</p>

### **UNIDAD 3 REDES LOCALES Y SERVICIOS DE RED**

b) Propósitos de la unidad

Que los alumnos conozcan las redes de información y su importancia

Que los alumnos realicen comunicaciones remotas y sea capaz de obtener información a través de la red

Que los alumnos transfieran archivos a través de la red.

HORAS	CONTENIDO	DESCRIPCIÓN DEL CONTENIDO	ESTRATEGIAS DIDÁCTICAS (actividades de aprendizaje)
15	<p>Tipos de redes</p> <p>Estándares de redes</p> <p>Hardware de redes</p> <p>Software de redes</p> <p>Canales de</p>	<p>Redes LAN y WAN</p> <p>Topología de estrella, anillo y bus Ethernet y token ring</p> <p>Servidor, terminales, tipos de tarjeta para red.</p> <p>Novell, Microsoft, UNIX</p> <p>Líneas telefónicas , cable coaxial,</p>	<p>Investigar y discutir la clasificación de las redes.</p> <p>Investigar en qué tipo de red está trabajando, discutir por que.</p> <p>Investigar qué tipos de red están instaladas en la UNAM</p> <p>Investigar qué tipo de red tienen en las sucursales bancarias.</p> <p>Discutir de manera grupal por qué se instala un tipo u otro de red.</p> <p>Definir con el profesor por que se</p>

	comunicación	cable de fibra óptica, microonda y satélites. Servicios de transmisión , línea privada, línea conmutada y portadoras comunes especializadas.	necesita un administrador de la red y cuál es su función. Tramitar claves y discutir el por qué de la asignación de éstas. Mandar y recibir mensajes a través del correo electrónico Inscribirse a una lista de discusión
	Servicios de red	Internet: correo electrónico, listas de discusión , Red UNAM , Gopher, transferencia de archivos (imágenes y texto)  Visualizadores de páginas electrónica World Wide Web	. Búsqueda bibliográfica a través de red UNAM Enviar y traer archivos a través de los protocolos Realizar consultas de World Wide Web a través de netscape u otro visualizador

#### **Unidad 4 LA COMPUTADORA Y EL ANÁLISIS ESTADÍSTICO**

##### b) Propósitos de la unidad

Que el alumno procese los datos obtenidos de sus investigaciones y prácticas escolares de forma más rápida.

Que el alumno a través de los diferentes tipos de gráficos, interprete mejor su información.

Que el alumno presente con mayor calidad los resultados de sus trabajos académicos.

<b>HORAS</b>	<b>CONTENIDO</b>	<b>DESCRIPCIÓN DEL CONTENIDO</b>	<b>ESTRATEGIAS DIDÁCTICAS (actividades de aprendizaje)</b>
15	Conceptos básicos de estadística	Aquí el alumno revisará los conceptos de: dato, variable, rango, intervalo y frecuencia,	El profesor discutirá estos conceptos con sus alumnos, considerando que estos últimos cursan la materia de estadística.
	Captura de datos	Conceptos de columna, renglón (fila), dirección absoluta, relativa y mixta, celdas, tipo de datos.	El alumno diseñará y aplicará un cuestionario para obtención de datos y su tratamiento. Definir variables y constantes.
	Análisis de datos	Operadores aritméticos y lógicos, prioridad de los operadores, fórmulas y funciones (matemáticas y estadísticas) ,  Modificación de los datos (insertar, mover, y copiar datos)	Analizarán la información aplicando funciones de la hoja de cálculo (medidas de dispersión y tendencia central)  A través de fórmulas y operadores diseñará prueba de T y $X^2$
	Graficación de datos	Tipos de gráficas (barra, línea, pastel)	

			Graficará los datos, analizará e interpretará la información, formulará conclusiones. El alumno realizará análisis estadístico de investigaciones y trabajos experimentales de otras asignaturas utilizando la computadora.
--	--	--	--

## **BIBLIOGRAFÍA**

1. Freeman, A.. Diccionario de Computación . Hill. México, Mc Graw 1993
2. Pearson, R.Quickstart Excel versión 5 para Windows ., México, Prentice Hall 1993.
3. Flast, L. Lotus 1 2 3.. México, Mc. Graw Hill 1990.
4. Microsoft Corporation . Excel 5 para Windows., Guía del usuario.. México, 1994
5. Borland International Inc. Quattro Pro for windows., User's Guide . USA, 1993
6. Sanders, D. Informática, presente y futuro... 1a edición, México, McGraw Hill 1992
7. Long, L. Introducción a la Informática y al procesamiento de la Información.. México, Prentice Hall 1994
8. Levine, G.. Introducción a la computación y a la programación estructurada. México, Mc Graw Hill 1993
9. Norton P. Toda la PC ., México, Prentice Hall 1993
10. Tucker, A. Lenguajes de programación., México, Mc Graw Hill 1986
11. Ninestein, E . Matemáticas Básicas para computación., México, Trillas 1994
12. Lozano, R. Diagramación y programación., México. Mc Graw Hill 1986
13. Hoffman, P. Iternet Manual de Bolsillo, , México, Mc Graw Hill 1995
14. Peter, K. Iternet fácil , , México, Prentice Hall 1994
15. Levine, B. Iternet para inexpertos , , México, Noriega Editores 1995
16. Minatel J. World Wide Web con Netscape visual, , México. Prentice Hall 1995
17. Schneider, W. Introducció n a la programación y solución de problemas con Pascal, México. Limusa 1986.
18. Person, P. 1-2-3 En el mundo de la estadística., México, Macrobit 1990.

# CORELDRAW

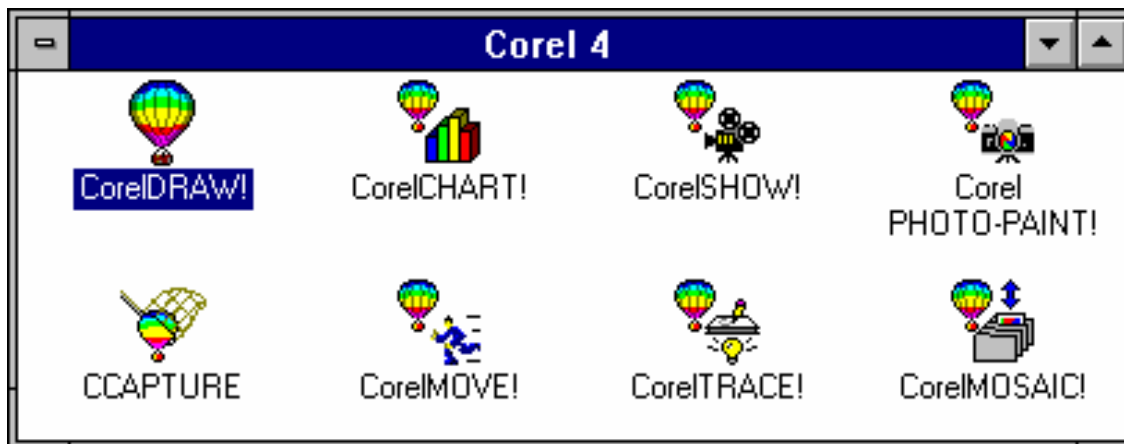
(Diseño Grafico)

CorelDRAW realiza la creatividad al permitir editar cualquier línea, forma o carácter con facilidad y precisión; ajusta el texto según una curva; la vectorización automática de gráficos; crear separaciones de color personalizadas y presentaciones animadas; producir documentos para publicaciones en autoedición, y realizar muchas otras cosas.

Es posible combinar las características de CorelDRAW para conseguir muchos efectos especiales diferentes, como situar una línea de texto o un objeto en perspectiva; doblar, contornear, girar o extrusionar una línea de texto o un objeto; mezclar dos líneas de texto o dos objetos; crear imágenes especulares, mascarar y simulaciones 3-D. CorelDRAW ofrece éstas y otras posibilidades de trabajo a velocidades superiores a las de otros programas gráficos.

CorelDRAW incluye aplicaciones o productos principales, estos productos o aplicaciones son:

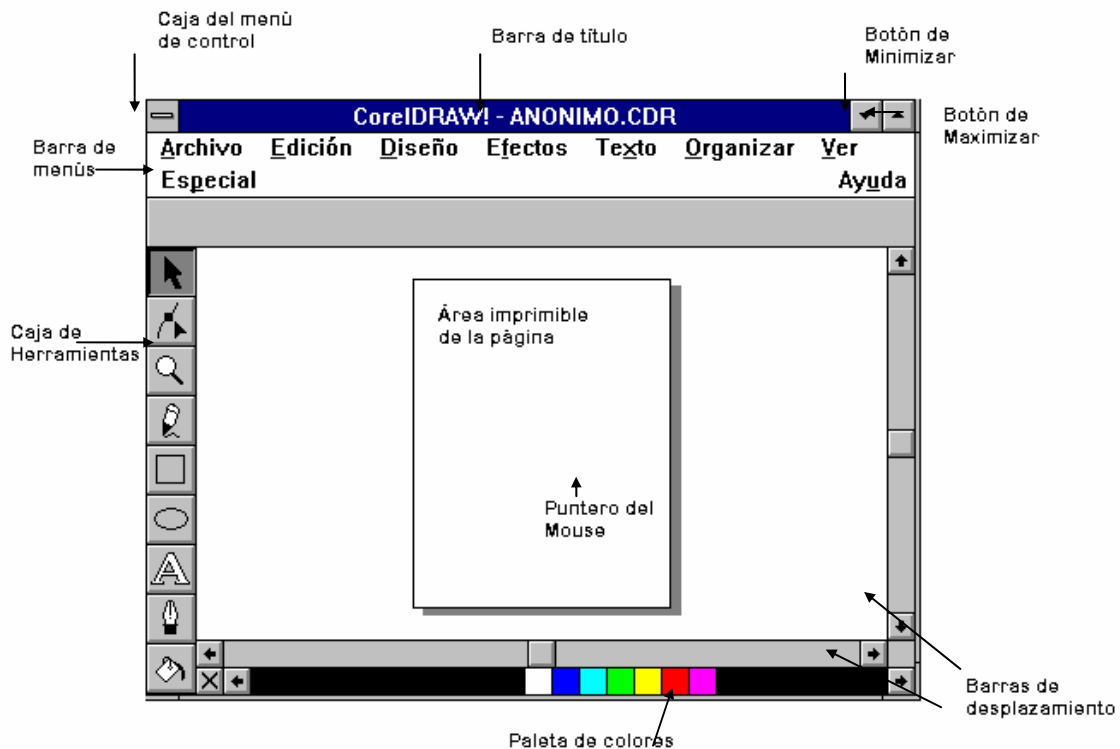
<b>CorelDRAW</b>	Dibujo e ilustraciones con manejo de texto.
<b>CorelPHOTO-PAINT</b>	Pintura y retoque de fotografías con realce de imágenes.
<b>CorelCHART</b>	Representación gráfica de datos introducidos o importados.
<b>CorelSHOW</b>	Creación de presentaciones con otros objetos de Corel.
<b>CorelMOVE</b>	Creación de presentaciones animadas.
<b>CorelTRACE</b>	Conversión de imágenes de mapa de bits gráficos vectoriales.
<b>CorelMOSAIC</b>	Almacenamiento, organización, compresión y acceso a imágenes gráficas.
<b>CCAPTURE</b>	Captura de imágenes de la pantalla.



## Ventana de Aplicaciones de CorelDRAW

**CorelDRAW** es adecuado para documentos de una o dos páginas, de gran contenido visual como prospectos, carteles y pequeños folletos

La pantalla de trabajo de CorelDRAW se compone de gran cantidad de elementos informativos de control. Alguno de estos elementos pueden ser activados o desactivados individualmente por el usuario para obtener una visión sobre la pantalla y una comodidad de trabajo óptimos; otros elementos permanecen siempre en pantalla. fig.C-1










(fig.C-1)

### Pantalla de CorelDRW

#### Los elementos de la Ventana son:

- Borde de las Ventanas.
- Barra de título.
- Botón de minimizar.
- Botón de maximizar.
- Caja del menú de control.
- Barra de menús.

-  La barra Banda.
-  Área imprimible de la página.
-  Barras de desplazamiento.
-  Reglas.
-  Caja de herramientas.
-  Paleta de colores.
-  Línea de estado.










### **Los MENÚS de CorelDRAW**

Cuando se abre o despliega un menú algunas órdenes aparecen en negritas, mientras que otras aparecen más claras o en gris. Es posible seleccionar cualquier orden que aparezca en negritas, pero las órdenes que aparecen en gris no están disponibles en ese momento.

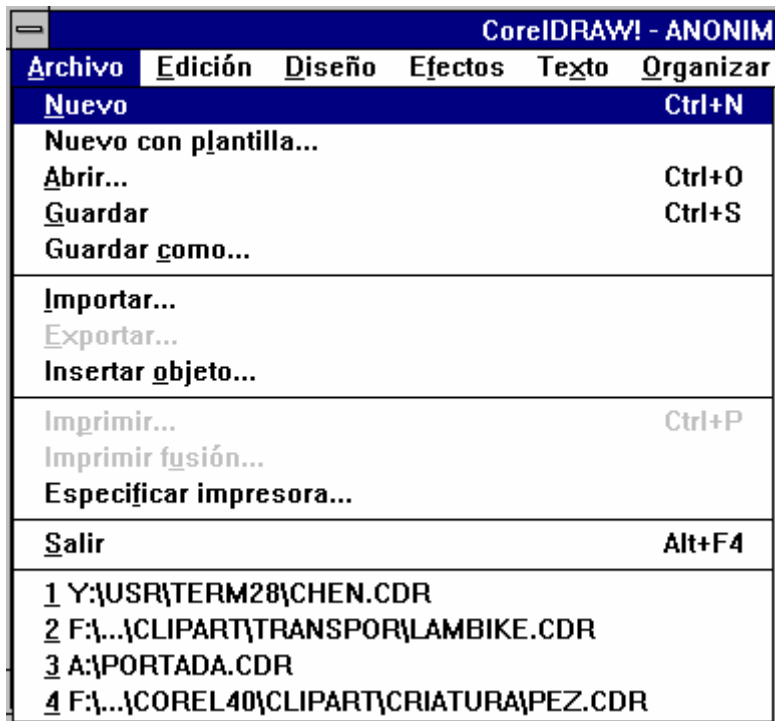
Según los objetos con los que se trabaje y las acciones que se realicen sobre ellos, las órdenes estarán o no disponibles para ser seleccionadas.

El uso de los menús se irán aprendiendo a medida que se trabaje con las funciones concretas de Corel.

Algunos de los menús que se encuentran son:

-  Menú Archivo. (File)
-  Menú Edición. (Edit)
-  Menú Ver. (View)
-  Menú Diseño. (Layout)
-  Menú Organizar. (Arrange)
-  Menú Efectos. (Effects)
-  Menú Texto. (Text)
-  Menú Especial. (Special)
-  Menú Ayuda. (Help)

Cada uno de los menús contienen opciones correspondientes, por ejemplo se muestra a continuación algunos menús con sus opciones.



**Menú Archivo:** Algunas de sus opciones son:

**Nuevo:** Abre un documento nuevo.

**Nuevo con Plantilla:** Abre un documento con una plantilla

**Abrir:** Abre un documento existente.

**Guardar:** Guarda el documento.

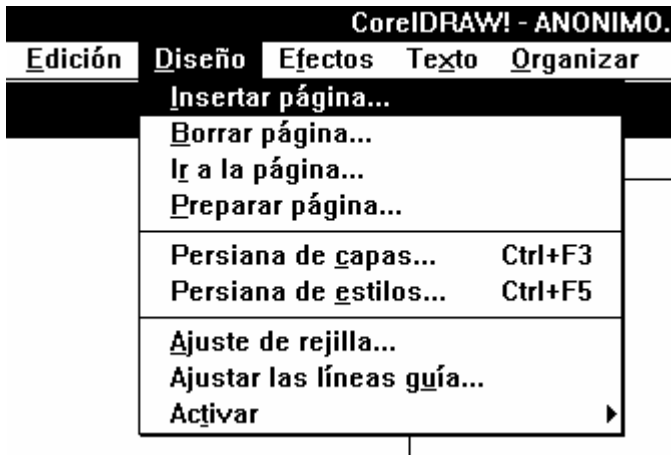
**Importa:** Importa objetos de un archivo a otro.

**Exportar:** Exporta objetos o documentos a otro archivo.

**Insertar objeto:** Inserta objetos de cualquier tipo que se encuentra en el menú.

Etc.





### Menú Diseño

**Insertar página:** Inserta páginas.

**Borrar página:** Borra páginas

**Ir a la página:** Lo lleva a una página seleccionada.

**Preparar página:** Prepara página antes de iniciar el trabajo.

**Persiana de capas:** Permite varias opciones para preparar la capa.

**Persiana de estilo:** Muestra opciones de estilos.

**Ajuste de rejilla:** Ajusta la frecuencia de la rejilla.

**Ajustar las líneas guía:** Permite ajustar el tipo de línea y posición de la regla.

**Activar:** Activa la rejilla, líneas en objetos.

Estas son las opciones de algunos menús de los cuales en cada uno de ellos muestra unas ventanas que son las de dialogo son las que permiten trabajar seleccionando los requerimientos

### **La Caja de Herramientas**

Una de las características que hace tan sencillo el trabajo con CoreIDRAW es la sencillez de la pantalla. El número de herramientas de la caja de herramientas es aparentemente pequeño. Varias herramientas tienen más de una función y se despliegan en submenús laterales anidados cuando se seleccionan. Este método de organización reduce el desorden de la pantalla y mantiene juntas las funciones relacionadas.

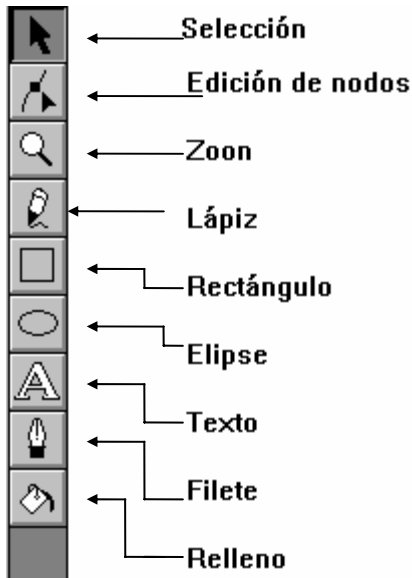
Las herramientas de CoreIDRAW realizan tres tipos diferentes de funciones. Una permite dibujar, otra editar los objetos ya dibujados y el tercer grupo permite alterar el aspecto de la pantalla para trabajar más eficazmente.

### **Herramientas de Dibujo**

CoreIDRAW permite crear o trabajar con nueve tipos diferentes de objetos. No obstante, dado que se utilizan las mismas herramientas y técnicas para algunos objetos, en realidad sólo hay cinco clases diferentes de objetos.

Estas clases, y los tipos de objetos que pueden diseñarse con cada una de ellas, son:

- Líneas, curvas y polígonos.
- Rectángulos y cuadrados.
- Elipses y círculos.
- Texto.
- Imágenes de mapas de bits (basada en pixeles) importadas desde un escáner o programa de pintura.



Los artistas y diseñadores gráficos profesionales saben que las formas geométricas básicas son los elementos de construcción con los que pueden construirse imágenes más elaboradas. Después de “construir” un objeto mediante una de las herramientas de dibujo, se pueden utilizar una o varias herramientas de edición de la caja de herramientas de CorelDRAW para cambiar su forma, reorganizarlo, colorearlo y contornearlo.

Todo lo que necesita para crear ocho de los nueve tipos de objetos en CorelDRAW son las cuatro herramientas creativas, la herramienta Lápiz, la herramienta Rectángulo, la herramienta Elipse, y la herramienta Texto. Con el último tipo de objeto, una imagen de mapas de bits, se trabajará una vez importado.

**La herramienta Lápiz.** Es la herramienta de dibujo más básica que contiene la caja de herramientas de CorelDRAW . Esta sola herramienta permite crear líneas, curvas, objetos curvos y polígonos. La herramienta Lápiz puede cambiarse de un sencillo instrumento de dibujo a mano alzada a un instrumento multimodo más sofisticado para trazar líneas y curvas Bézier o uno de los cuatro modos Líneas de cota. El modo Mano alzada es la opción por defecto y se utiliza para los trabajos menos precisos. El modo Bézier se selecciona desde el menú lateral. Pulsando sobre la herramienta Lápiz y manteniendo pulsado el botón del ratón se visualizan seis iconos, uno por cada modo de dibujo. El modo Bézier se utiliza para curvas suaves y precisas, que permanecen así incluso cuando se aumentan o distorsionan.

Los cuatro modos de líneas se utilizan de cotas se utilizan para trazar líneas que muestran las dimensiones entre dos objetos o lugares o para identificar un objeto, como es usual en los dibujos técnicos.

**La herramienta Rectángulo.** Esta herramienta permite dibujar rectángulos y cuadrados. Sin embargo, para redondear las esquinas de un rectángulo es necesario utilizar la herramienta Edición nodos, una de las herramientas de edición de la caja de herramientas de CorelDRAW.

Para seleccionar la herramienta Rectángulo basta con pulsar la tecla F6 o sobre su icono y , a continuación, desplazar el puntero del ratón al interior del espacio blanco de la página. El icono rectángulo aparecerá resaltado, el puntero del ratón se convierte en una cruz y en las líneas de estado aparecerá el mensaje “Rectángulo”.

**La herramienta Elipse.** Permite crear elipses y círculos perfectos . Para seleccionar la herramienta Elipse basta con pulsar la tecla F7 o sobre su icono y, a continuación, desplazar el puntero del ratón al interior del espacio blanco de la página. El icono Elipse aparecerá resaltado, el puntero del ratón de convertirá en una cruz y en las líneas de estado aparecerá el mensaje “Elipse”.

**La herramienta Texto.** Contiene un menú lateral con dos iconos para elegir entre texto artístico y texto de párrafo. Un texto artístico es una simple cadena de caracteres como la que podría aparecer en un cartel. Un texto de párrafo es más largo y tiene más atributos de espaciado; el texto de un catálogo es un ejemplo de texto de párrafo. Cualquiera de los modos de la herramienta Texto permite el acceso a muchas fuentes de Corel system, así como a miles de otras fuentes comerciales. Junto a los caracteres alfabéticos, CorelDRAW incluye una extensa biblioteca de símbolos a los que también se tiene acceso con las herramientas Texto.

## **Herramientas de edición**

Una vez que se han creado objetos en una página con las herramientas de dibujo o texto, para desplazar, ordenar, cambiar y manipular los objetos se utiliza un grupo diferente de herramientas.

Entre las herramientas de edición se encuentran la herramienta Selección, la herramienta Edición de nodos, la herramienta Filete y la herramienta Relleno.

**La herramienta Selección.** En realidad son dos herramientas en una, según el modo de trabajo. En el modo selección, pueden seleccionar objetos para desplazarlos, reordenarlos, agruparlos o combinarlos. En el modo transformación pueden usarse la herramienta Selección para girar, inclinar, estirar, reflejar, desplazar o redimensionar el objeto seleccionado. No obstante, esta herramienta no permite modificar la forma básica de un objeto.

**La herramienta Edición de nodo.** Permite modificar la forma de un objeto. Con esta herramienta es posible suavizar o distorsionar cualquier figura, redondear los vértices de un rectángulo, convertir un círculo en una cuña o un arco, modificar una curva o ajustar el espaciado entre caracteres individuales de una cadena de texto. La herramienta Edición de nodos puede seleccionarse, pulsando sobre su icono o mediante la tecla F10.

**La herramienta Filete.** Al igual que la herramienta Selección, la herramienta Filete tiene más de un modo de funcionamiento. La herramienta Filete y su menú lateral asociado se utiliza para elegir un color de filete estándar o personalizado, o bien para crear una “pluma” de filete personalizado para el objeto seleccionado

**La herramienta Relleno.** Esta herramienta y su menú lateral asociado se utilizan para seleccionar un color de relleno estándar o personalizado para texto u objeto seleccionado.

## **Herramientas para la personalización de la pantalla de CorelDRAW**

Un tercer grupo de herramientas permite personalizar la interfaz de CorelDRAW para conseguir que trabaje en la forma deseada. Sólo una de estas herramientas, la herramienta Zoom, es visible en la caja de herramientas de CorelDRAW. La herramienta Zoom y su menú lateral asociado permiten controlar qué parte de la imagen es visible en cada momento. Esta herramienta se utiliza cuando se desea trabajar con más detalle sobre un área pequeña o cuando se necesita ampliar o reducir la imagen.

Cuando se ejecuta CorelDRAW o se abre un archivo nuevo, la opción por defecto es visualizar la página completa en la pantalla. Esta vista presenta limitaciones obvias si se necesita editar las imágenes o trabajar con pequeños detalles. La herramienta Zoom permite personalizar el área de visión de la pantalla en la forma que se desee. A medida que esta herramienta vaya resultando más familiar, se verá que es adecuada y facilita el trabajo con dibujos más grandes.

La herramienta Zoom se parece a una pequeña lupa y es en realidad una herramienta de ajuste de la visualización porque permite aumentar o disminuir el área de visión de varias maneras. Dado que la herramienta Zoom proporciona un control completo sobre el contenido del área de visualización, permite mejorar todos los objetos dibujados y aumentar la utilidad de todas las herramientas de CorelDRAW.

En realidad, la herramienta Zoom es un conjunto de seis herramientas independientes. Cuando se selecciona esta herramienta aparece un menú lateral que proporciona seis herramientas para ajustar el área de visión, como se muestra a continuación.

**Acercar:** Esta herramienta permite aumentar cualquier área del dibujo que se seleccione trazando un rectángulo con la herramienta Acercar alrededor del área deseada.

**Alejar:** Esta herramienta reduce la imagen actual en un factor de dos o, si la pantalla actual muestra una vista ampliada, regresa a la vista anterior.

**Zoom tamaño real:** permite ver el dibujo en el tamaño real con que se imprimirá.

**Zoom sobre la selección:** Esta herramienta amplía los objetos seleccionados del dibujo de modo que ellos y sólo ellos rellenen el área de dibujo.

**Zoom sobre todo.** Permite ver el dibujo actual completo, todo lo que se haya colocado en la página en la ventana actual de CorelDRAW.

**Zoom sobre página.** Permite volver a la vista por defecto, la página completa del gráfico.

## **Visualización de un objeto seleccionado**

A menudo sólo se desea ver parte de un objeto, es posible enmarcar dicha parte con la herramienta Acercar y visualizarla. Con CorelDRAW, además, utilizando la herramienta Zoom sobre la selección puede seleccionarse parte de un objeto y ampliar sólo la parte seleccionada. Esto es más sencillo que enmarcar y es útil para objetos de forma rara.

Veamos como funciona esta herramienta con un dibujo que se tenga en pantalla siguiendo estas instrucciones ( la imagen de la pantalla debería estar reducida para mostrar alguno o todos los bordes de la página ):

1. Pulse sobre el objeto de la pantalla que desee seleccionar. Aparecerán ocho cuadros de selección alrededor del objeto.
2. Pulse sobre la herramienta Zoom para abrir su menú lateral y después sobre Zoom sobre la selección. El objeto de la pantalla se ampliará hasta rellenar la ventana de visualización.
3. Borre la pantalla seleccionada Nuevo (New) en el menú Archivo (File).

## **Ajuste de un gráfico en una ventana**

Cuando las imágenes dibujadas se extienden a todo lo largo de la página, quedan ya encajadas dentro de la ventana de visualización . En estos casos, la herramienta Zoom sobre todo no sirve para mucho. Sin embargo, cuando existan algunos espacios en blanco puede utilizarse la herramienta Zoom sobre todo para visualizar todo lo que se haya dibujado y nada más. Esta opción es especialmente útil para pequeños diseños tales como logotipos.

Para seleccionar la herramienta Zoom sobre todo, primeramente hay que seleccionar la herramienta Zoom y después pulsar sobre la herramienta Zoom. La pantalla se redibujará y se rellenará con la vista completa de la imagen.

### **Visualización de una página entera**

Cuando se carga CorelDRAW o se abre un dibujo, la opción por defecto es la visualización de la página completa. Es muy sencillo regresar a esta visualización desde cualquier otra que se haya establecido; sólo hay que seleccionar la herramienta Zoom y pulsar sobre la herramienta Zoom sobre página o pulsar las teclas MAYUS-F4.

Aunque ya se han examinado todas las herramientas de selección de la visualización, todavía existe otro método, denominado *panorámica*, que puede utilizarse para controlar el área de visualización.

## **DIBUJO DE LÍNEAS Y CURVAS**

La herramienta Lápiz es la más versátil de todas las incluidas en la caja de herramientas de CorelDRAW. Utilizando esta herramienta en dos de sus cuatro modos pueden crearse líneas rectas y curvas y, a partir de estos sencillos elementos, construir una variedad casi infinita de polígonos y formas irregulares. El tercer modo, Líneas de cota, permite colocar líneas verticales, horizontales y angulares que muestren las dimensiones de objetos otras líneas de una página. El cuarto modo permite colocar notas en un dibujo.

### **Modo mano alzada frente a modo Bézier**

La herramienta Lápiz tiene dos modos de dibujo: Mano alzada, en el que las curvas reflejan los movimientos de la mano con el ratón, y el modo Bézier, en el que las curvas se colocan con precisión entre dos o más puntos identificados previamente. En ambos modos, el dibujo de líneas rectas es muy similar, pero el dibujo de curva es muy diferente.

### **Dibujo de Línea Recta**

En la terminología de CorelDRAW la palabra *línea* hace referencia a cualquier línea recta, mientras que la palabra *curva* se refiere a líneas curvas, líneas irregulares y objetos cerrados creados con estos tipos de líneas. El dibujo de una línea recta requiere manejar el ratón de manera diferente que cuando se dibuja una línea curva o irregular. Para dibujar una línea recta en CorelDRAW se siguen los siguientes pasos.

1. Sitúe el puntero del ratón sobre el icono de la herramienta Lápiz y dé una pulsación. Al desplazar el puntero fuera del icono se convierte en una cruz, y el icono de la herramienta Mano alzada se oscurece.
2. Sitúe el puntero en cruz donde desee iniciar la línea. Puede ser cualquier punto del área imprimible de la página.
3. Pulse y *suelte inmediatamente* (pulsar sobre) el botón izquierdo del ratón y después desplace la cruz del puntero hasta el punto donde desee que termine la línea. Como se muestra a continuación.

4. Cuando haya establecido la longitud y dirección deseadas, pulse el botón del ratón para completar la línea. Como muestra la siguiente ilustración, aparece un nodo cuadrado en cada extremo de la línea que indican que se ha completado y puede seleccionarse para seguir trabajando con ella.
5. Pulse la tecla SUPR para borrar la línea de la pantalla.

### **Borrado de fragmentos en una línea**

Con CorelDRAW siempre es posible retroceder y acotar una línea mientras no se haya dado la segunda pulsación para completarla.

- Con la herramienta Mano alzada aún seleccionada, escoja un punto de partida para otra línea.
- Desplace el puntero hacia abajo y hacia la derecha hasta que el indicador  $dx$  muestre aproximadamente 5,00 pulgadas y el indicador  $dy$  - 1,00 pulgadas.
- Sin pulsar el botón del ratón por segunda vez, acorte la línea hasta que en el indicador  $dx$  se lean 4,00 pulgadas.
- Pulse el ratón una segunda vez para fijar la línea aproximadamente  $dx$  igual a 4,00 pulgadas.
- Antes de continuar, pulse SUPR para borrar la línea.

### **Dibujo de líneas con múltiples segmentos**

Con CorelDRAW puede dibujar fácilmente una secuencia de líneas rectas de manera que cada una de ellas comience donde finaliza la anterior. Este método se utiliza tanto para dibujar figuras abiertas como para construir polígonos.

*En el siguiente ejercicio creará una serie de picos y valles.*

1. Seleccione la herramienta Mano alzada mediante un atajo de teclas, pulse la tecla F5. Después pulse el ratón para establecer el punto inicial. Extienda la línea hacia arriba y hacia la derecha.
2. Cuando llegue al punto final deseado fije la línea mediante una *doble pulsación* en lugar de una pulsación simple
3. Sin pulsar el ratón nuevamente, desplácelo hacia abajo y hacia la derecha. La línea final flexible seguirá automáticamente a la cruz del puntero.
4. Dé otra doble pulsación para fijar el segundo segmento.
5. Continúe zigzagueando de esta manera hasta que haya creado varios picos y valles similares.
6. Cuando alcance el último valle, pulse una vez el ratón en lugar de dos para finalizar la línea multisegmento.
7. Pulse la tecla SUPR para borrar la pantalla antes de continuar.

**Sugerencia:** Si después de la última pulsación del ratón se ha cometido algún error de dibujo, el último segmento dibujado puede borrarse de dos formas distintas. Pulsando las teclas ALT-RETROCESO para borrar segmentos de línea individuales, o seleccionando la orden Deshacer Anexar curva. En caso de pulsar la tecla SUPR cuando se dibuja una línea multisegmento se borrarán todos los segmentos dibujados hasta ese momento.

## **Dibujo de un Polígono**

Un *polígono* es una figura bidimensional cerrada limitada por líneas rectas. Con CorelDRAW pueden crearse polígonos dibujando líneas de múltiples segmentos y haciendo coincidir el punto inicial con el punto final. En el siguiente ejercicio se creará un polígono .

1. Dibuja la primera línea, dando una doble pulsación en el punto final de la línea para poder seguir dibujando sin interrupción.
2. Dibuje cuatro líneas más de la misma manera. Finalice el ultimo segmento de línea pulsando una vez más sobre el punto de partida del primer segmento.

¿ Se “conectó” el último segmento con el principio del primero, o quedó una pequeña distancia entre ellos ? No importa si se continúa viendo una pequeña separación. Si las líneas se unieron formando un polígono no cerrado, se debería rellenar de un color negro sólido si están vigentes las implantaciones por defecto. Por ahora, borre la pantalla y comience el siguiente ejercicio.

## **Líneas Rectas en modo Bézier**

El modo de dibujo Bézier consiste en identificar los puntos extremos o nodos y colocar líneas o curvas entre ellos. Por tanto, dibujar líneas rectas en modo Bézier es muy similar a dibujarlas en modo Mano alzada. Inténtelo a continuación y compruébelo por si mismo.

### **Para usar el modo Bézier:**

1. Seleccione el modo Bézier pulsando sobre la herramienta Lápiz y mantenga pulsando el botón del ratón hasta que aparezca un menú lateral como el siguiente:
2. Pulse sobre la herramienta Bézier, el segundo icono de la izquierda de los seis que hay en el menú lateral. La línea de estado indicará ahora “*Dibujando en modo Bézier..*” Dibuje ahora un solo segmento de línea al igual que hizo antes en el modo Mano alzada.

## **Dibujo de un polígono en modo Bezier**



Al contrario que el modo Mano alzada, en modo Bézier es posible seguir añadiendo segmentos adicionales a una línea después de haber establecido un modo final con una sola pulsación del ratón. A continuación se construirá un polígono.

1. Pulse el ratón para establecer dos nodos más, el primero en el lado inferior derecho y el segundo en el inferior izquierdo. Aparecerán líneas que unen estos nodos.
2. Sitúe el puntero del ratón sobre el nodo de partida y pulse el ratón una vez más. El resultado es un polígono de cuatro lados.

Como puede verse, hay muchas similitudes en cuanto a la mecánica entre los modos Mano alzada y Bézier para el dibujo de líneas rectas. Los resultados son prácticamente los mismos, pero la pantalla tiene un aspecto muy diferente durante la creación de las líneas.

**Consejo:** Al igual que en modo Mano alzada, en modo Bézier puede desplazarse el puntero del ratón en cualquier dirección, incluso hacia atrás, sobre el camino ya trazado para “borrar” el objeto antes de definir un nodo. Después de establecer un nodo se puede usar la característica Deshacer (ALT-RETROCESO).

### **Dibujo de curvas en modo Mano alzada**

La herramienta Lápiz tiene un doble propósito en CorelDRAW, puede utilizarse para dibujar líneas irregulares o curvas, así como líneas rectas.

Para dibujar una curva simple siga los siguientes pasos:

1. Seleccione la herramienta Mano alzada si no está ya seleccionada.
2. Sitúe la cruz del puntero en la posición de la página donde desee que empiece la curva y después *pulse y mantenga pulsado* el botón del ratón. En la línea de estado aparecerán las coordenadas del principio y fin.
3. Mantenga pulsando el botón del ratón y *arrastre éste* a lo largo de la trayectoria que desee para la curva.
4. Una vez finalizado la curva, libere el botón del ratón. Mientras CorelDRAW calcula exactamente dónde debe situar la curva, ésta desaparecerá momentáneamente. Seguidamente, la curva reaparecerá con varios pequeños nodos de forma cuadrada.

### **Borrado de fragmentos de una curva**

Si se comete algún error al dibujar una curva, se puede volver atrás y borrar lo que se ha dibujado, siempre y cuando no se haya liberado el botón del ratón. Para borrar el fragmento de curva no deseado se utiliza la tecla MAYUS.

1. Pulse y mantenga pulsando el botón del ratón en la posición en la que desee empezar a dibujar otra curva.
2. Desplace el ratón a su antojo. No libere aún el botón del ratón.

3. Mientras mantiene pulsando el botón del ratón, pulse y mantenga pulsada la tecla MAYUS y retroceda sobre el fragmento de curva que desee borrar.
4. Cuando haya borrado el fragmento de curva, libere la tecla MAYUS y continúe el dibujo desplazando el ratón en la dirección deseada.
5. Para finalizar la curva, libere el botón del ratón. Pulse la tecla SUPR para borrar la curva.

## LA FUNCIÓN LÍNEAS DE COTA

Las líneas de cota se utilizan para visualizar medidas, tamaños de objetos o distancias entre ellos, como por ejemplo las que se utilizan en dibujos técnicos.

CorelDRAW dispone de tres herramientas para líneas de cota que son :

Vertical

Horizontal

Angular

### Ajuste de la escala del dibujo

La escala del dibujo se establece en la parte superior de la caja de diálogo Ajuste de rejilla y escala, a la que se accede desde el menú Diseño.

En ella se iguala un número de unidades sobre la página con su equivalente en “*el mundo real*”. Por ejemplo, para medir la escala de un aeroplano, se establece la equivalencia entre las pulgadas de la página y los pies del aeroplano, o el número de milímetros de una reproducción fotográfica con los metros del objeto.

### Uso de líneas de cota

Dado que las tres herramientas de líneas de cota operan de un modo similar, sólo se practicará con dos de ellas. Para dibujar líneas de cota son necesarios tres pasos que son:

1. Pulse sobre el punto del dibujo donde desee empezar a medir.
2. Arrastre la línea de cota hasta la posición en la que desee situar el punto final y pulse el botón del ratón para fijar.
3. Desplace el puntero, trasladando la línea de cota y el rectángulo de rótulo, hasta donde desee extender la línea de cota y colocar el rótulo y luego pulse el botón del ratón para fijar la línea y el rótulo.

## RECTÁNGULOS, CUADRADOS, CÍRCULOS Y ELIPSES

Los rectángulos y las elipses son formas básicas que subyacen en muchas formas complejas de la naturaleza o creadas por el hombre.

En esta sección se utilizará la herramienta Rectángulo para crear rectángulos y cuadrados y la herramienta Elipse para crear elipses y círculos.

## **Dibujo de un Rectángulo**

Utilizando la herramienta Rectángulo de la caja de herramientas de CorelDRAW se puede dibujar un rectángulo a partir de uno de sus cuatro vértices, así como desde el centro hacia afuera. Este grado de libertad y control para la colocación de los rectángulos ahorrará tiempo y esfuerzo a la hora de diseñar las ilustraciones.

### **Dibujo de un rectángulo desde cualquier vértice**

Un rectángulo se puede comenzar a dibujar a partir de cualquiera de sus cuatro vértices. A medida que se dibuja, el vértice que representa el punto de partida permanece fijo mientras que el resto de las líneas se expande o contrae según se desplaza diagonalmente el puntero. Esta flexibilidad en la elección del punto inicial permite ubicar un rectángulo con precisión y rapidez en un dibujo. El siguiente ejercicio permite familiarizarse con el modo en que reacciona CorelDRAW cuando utilizan diferentes vértices como punto de partida.

1. Pulse sobre el menú Ver y asegúrese de que hay marcas de verificación junto a los órdenes Regla y Dibujo de líneas . Si no hay, pulse sobre ambas órdenes para insertarlas.
2. Elija preferencias en el menú Especial y pulse sobre la lengüeta Ver. Si no hay una X en la casilla de verificación Mostrar línea de estado, pulse sobre ella para que aparezca. Pulse después sobre Colocar arriba para desplazar la línea de estado desde su posición por defecto al final de la pantalla a su posición alternativa por encima de la regla horizontal.
3. En el menú Ver, elija Caja de herramienta y a continuación, pulse sobre flotante para desbloquear la caja de herramientas y poder desplazarla por la pantalla. De nuevo, esto le proporciona la posibilidad de colocar la caja de herramientas en distintas posiciones.
4. Seleccione la herramienta Rectángulo situando el puntero del ratón sobre el icono Rectángulo y dando una pulsación; a continuación, desplace el puntero fuera de la caja de herramientas y hacia la derecha. El puntero del ratón cambia a una cruz, el icono Rectángulo queda seleccionado y, en la línea de estado, aparece “Rectángulo”.
5. Sitúe el puntero en cualquier parte del área imprimible de página, pulse y mantenga pulsando el botón izquierdo del ratón y arrastre éste hacia abajo y hacia la derecha siguiendo una trayectoria diagonal. A medida que desplaza el puntero cambiará la lectura de los indicadores Anchura y Altura de la línea de estado.
6. Practique con diferentes anchuras y alturas hasta conseguir la forma deseada. Variando la dirección de desplazamiento del ratón se puede modificar fácilmente la forma del rectángulo.
7. Libere el botón cuando el rectángulo tenga el tamaño y forma deseada. Con esta acción se fija el rectángulo en una posición, y en cada uno de sus cuatro vértices aparecerá un nodo. La línea de estado mostrará los mensajes “Rectángulo” y “Relleno” seguido de un

cuadrado negro. El mensaje “Relleno: Negro” indica que el color interior del rectángulo es el color por defecto, el negro. Si en el menú Ver no selecciono Dibujo de líneas, el rectángulo será negro.

8. Pulse y mantenga pulsado el botón del ratón sobre un nuevo punto de partida y desplace después el puntero siguiendo una trayectoria en diagonal hacia abajo y hacia la izquierda. Libere el botón del ratón cuando el rectángulo tenga las dimensiones deseadas. **Practique haciendo varios rectángulos que empiecen en diferentes vértices.**
9. Mientras permanece seleccionando el último rectángulo exhibirá nodos en los vértices, seleccione Borrar en el menú Edición o pulse la tecla SUPR para eliminarlo.
10. Borre la pantalla por completo seleccionando Nuevo en el menú Archivo y pulsando sobre No como respuesta a la pregunta “¿Guardar los cambios?”. Para borrar la página también se puede pulsar sobre Seleccionar todo en el menú Edición y, a continuación, pulsar la tecla SUPR.

### **Dibujo de un rectángulo a partir del centro**

CorelDRAW permite dibujar un rectángulo partiendo desde el centro. Este método se utiliza para situar formas rectangulares dentro de un gráfico con mayor precisión sin tener que prestar demasiada atención a las reglas o al espaciado de la rejilla. Los indicadores de anchura y altura muestran las dimensiones exactas del rectángulo conforme se dibuja. Dibuje ahora un rectángulo empleando el centro como punto de partida.

1. Con la herramienta Rectángulo seleccionada, Pulse y mantenga pulsada la tecla MAYS y el botón del ratón sobre el punto de inicio que desee. Siga pulsando la tecla MAYS y el botón del ratón mientras desplaza éste. Como con cualquier rectángulo, puede dibujar en cualquier dirección.
2. Para completar el rectángulo, libere primero el botón del ratón y después la tecla MAYS.
3. Pulse la tecla SUPR para borrar el rectángulo de la pantalla.

### **Dibujo de un Cuadrado**

En CorelDRAW se utiliza la misma herramienta tanto para dibujar rectángulos como cuadrados perfectos. El método es similar, con la excepción del uso de la tecla CTRL para restringir un rectángulo a un cuadrado.

### **Dibujo de un cuadrado desde cualquier vértice**

Para dibujar un cuadrado hay que seguir estos pasos. Como en un rectángulo, se puede utilizar cualquier vértice como punto inicial.

1. Con la herramienta Rectángulo seleccionada, sitúe el puntero donde desee iniciar el cuadrado.

2. Pulse y mantenga pulsados el botón del ratón y la tecla CTRL y, a continuación, dibuje diagonalmente en cualquier dirección. Observe que en la línea de estado los indicadores muestran que la anchura y la altura de la figura son iguales.
3. Para completar el cuadrado, libere en primer lugar el botón del ratón y, a continuación, la tecla CTRL. Si libera en primer lugar la tecla CTRL, puede ocurrir que dibuje un rectángulo en lugar de un cuadrado.
4. Pulse la tecla SUPR para borrar el cuadrado de la pantalla.

También se puede dibujar un cuadrado utilizando el centro como punto de partida al igual que se hizo con un rectángulo. Para ello se han de pulsar las teclas CTRL y MAYS, además del botón del ratón

## USO DE LA REJILLA

Si el trabajo incluye aplicaciones orientadas al diseño, tales como ilustraciones técnicas, representaciones arquitectónicas o diseños gráficos, algunas veces será necesario alinear formas geométricas horizontal o verticalmente según incrementos fijos.

Una manera de alinear los objetos mientras se dibujan consiste en aprovechar las ventajas de las órdenes Ajuste de rejilla y escala y Activar en el menú Diseño. El proceso de alineación de nuevos objetos con la rejilla consta de cuatro pasos:

- Ajustar el espaciado de la rejilla.
- Visualizar la rejilla.
- Visualizar las reglas.
- Habilitar la función Activar rejilla.

## Creación de un dibujo mediante Rectángulos y Cuadrados

El siguiente ejercicio hace uso de todos los conocimientos que se han adquirido hasta ahora para crear una taza de té que incluye elementos de dibujo rectangulares, cuadrados y a mano alzada.

Durante el proceso también se aprenderá a ajustar la suavidad relativa de las líneas curvas que se dibujen con la herramienta Mano alzada. Con el ajuste está relacionada la característica denominada Precisión a Mano alzada, que controla cuán estrechamente se adapta CorelDRAW a los movimientos del puntero del ratón cuando se dibujan curvas.

Como preparativos para este ejercicio hay que seleccionar la orden **Ajuste de rejilla y escala** en el menú **Diseño** y ajustar los valores Horizontal y Vertical de Frecuencia de la rejilla a 8,00 por pulgada. Tanto mostrar rejilla como Activar rejilla debe estar aún seleccionadas, y las reglas sobre la pantalla. Se pueden revisar los pasos de la sección anterior si es necesario.

Cuando esté en disposición de crear el dibujo, proceda a través de los siguientes pasos:

1. Dibuje el rectángulo que representa el cuerpo de la taza de té. Debe ser más alto que ancho.

2. Coloque el puntero en el lado derecho de este rectángulo y añada un asa rectangular al cuerpo de la taza. El asa debe tocar el borde de la taza, pero sin soloparse con él; los ajustes de rejilla seleccionados evitarán el solapamiento.
3. Dibuje un rectángulo más pequeño dentro del que acaba de crear para hacer la abertura del asa.
4. Seleccione la herramienta Mano alzada para añadir una línea recta en la base de la taza. (Debe seleccionar el modo Mano alzada, no el Bézier. Si necesita cambiar el modo de dibujo, pulse y mantenga pulsada la herramienta Lápiz hasta que aparezca el menú lateral y después pulse sobre el icono situado más a la izquierda.). Recuerde mantener pulsada la tecla CTRL mientras dibuja para asegurar que la línea queda perfectamente horizontal.
5. Trace las líneas diagonales para los extremos de la parte superior del platillo. Compruebe los indicadores de la línea de estado mientras dibuja. El ángulo de la línea diagonal debe ser de 45° grados, mientras que el ángulo para la línea diagonal de la derecha debe ser 135° grados. Una vez más pulse y mantenga pulsada de nuevo la tecla CTRL mientras dibuja la línea para asegurar que sigue incrementando de 15° grados. Además asegúrese de hacer ambas líneas de la misma longitud. Cada línea se unirá con la base del platillo formando una línea de múltiples segmentos.
6. Añada ahora otra línea para formar la base del platillo, Recuerde utilizar la tecla CTRL para restringir la línea. La base del platillo se unirá a los otros segmentos de línea para formar un único objeto, un polígono.
7. Con la herramienta Mano alzada aún seleccionada, pulse y mantenga pulsado el botón del ratón y dibuje una curva que represente la cuerda de la bolsita de té. ¿ Tiene la cuerda un aspecto demasiado irregular ? Si es así, ajuste la Precisión a Mano alzada .
8. Antes de ajustar el valor de Precisión a Mano alzada, seleccione Deshacer en el menú Edición para borrar la cuerda de la bolsita de té que acaba de dibujar.
9. Seleccione la lengüeta Curva de la orden Preferencias en el menú especial para visualizar la caja de diálogo. El valor por defecto en la caja numérica correspondiente a Precisión a mano alzada es 5, pero para hacer curvas más suaves hay que ajustar a un valor mayor.
10. Utilizando la flecha de desplazamiento, ajuste el nivel de sensibilidad de la opción Precisión a mano alzada a 10 pixels. Este es el mayor número posible y hace que CorelDRAW suavice las curvas al mismo tiempo que se dibujan. Por el contrario, los valores más pequeños hacen que la herramienta Lápiz siga el más mínimo movimiento del ratón.
11. Seleccione Aceptar (OK) para salir de la caja de diálogo y guardar las implantaciones.
12. Dibuje por segunda vez la cuerda de la bolsita de té. La curva será ahora algo más suave.
13. A continuación, añada la bolsita a la cuerda. Seleccione de nuevo la herramienta Rectángulo, sitúe el puntero aproximadamente a  $\frac{3}{4}$  de pulgada por debajo del extremo inferior de la cuerda, pulse y mantenga pulsadas simultáneamente las teclas CTRL y MAYS y dibuje un cuadro desde el centro hacia afuera.
14. Para añadir la etiqueta central de la bolsita, dibuje un cuadro de 10 dentro del primer cuadrado. Para este cuadrado más pequeño seleccione uno de los vértices como punto de partida.

15. Como toque final al dibujo, añade algo de vapor seleccionando la herramienta Mano alzada y dibujando algunas curvas. Puesto que estableció la Precisión a mano alzada en un número alto de pixels, el “vapor” creado será más convincente.
16. Para finalizar, guarde el dibujo. Seleccione la orden Guardar como en el menú Archivo y, después de seleccionar el directorio que desee, teclee **taza**. Al pulsar sobre el botón de orden Aceptar (OK), CorelDRAW añade automáticamente la extensión **.CDR**.
17. Elija Nuevo en el menú Archivo para borrar la pantalla y prepararla para un nuevo dibujo.

## Dibujo de una Elipse

La herramienta elipse de CorelDRAW permite crear elipses y círculos perfectos. En los ejercicios de esta sección se crearán elipses y círculos de varias formas y tamaños.

CorelDRAW permite iniciar una elipse a partir de cualquier punto de su perímetro o a partir del centro hacia afuera mediante el empleo de la tecla MAYS. Este segundo método permite situar con precisión elipses en un gráfico.

## Dibujo de un Círculo

En CorelDRAW se utiliza la misma herramienta para producir elipses y círculos perfectos, al igual que se utiliza la misma herramienta para crear rectángulos y cuadrados. La técnica es similar; usar la tecla CTRL para restringir una elipse a un círculo. Del mismo modo, se puede seleccionar el perímetro o el centro como punto de inicio del círculo.

## Dibujo con Círculos y Elipses

En el siguiente ejercicio se integran todos los conocimientos que se han adquirido hasta este momento. Se creará un dibujo (una casa y su entorno) que incluirá elementos de dibujo elípticos, circulares, rectangulares, cuadrados y de mano alzada. La rejilla puede ser de ayuda para algunos elementos geométricos del dibujo, otros elementos se pueden dibujar a mano alzada. Dado que el dibujo será más ancho que alto, se aprenderá también cómo ajustar el formato de página desde *vertical (formato por defecto)* a horizontal.

Los primeros pasos proporcionarán el hábito de preparación y anticipación de las necesidades del dibujo antes de empezar a dibujar, de modo que se pueda dibujar rápidamente y sin introducción.

Siga los siguientes pasos:

1. Para preparar la parte geométrica del dibujo, seleccione la orden **Ajuste de rejilla** en el menú **Diseño** y ajuste los valores *Horizontal* y *Vertical* de Frecuencia de la rejilla a 4,00 por pulgada.
2. Cambie el ajuste de página para que sea más ancha que larga. Para hacer esto seleccione la orden **Preparar página** en el menú **Diseño**. Cuando aparezca la caja de diálogo **Preparar página**, pulse el botón de opción Horizontal para establecer la orientación en horizontal. Para *salir* seleccione el botón de orden **Aceptar** (OK).
3. Seleccione la lengüeta **Curvas** de la orden **Preferencias** del menú **Especial** y ajuste los valores de **Precisión** a mano alzada y **Unión** automática en 10 pixels, si aún no lo están.

Para salir de ambas cajas de diálogo pulse dos veces sobre el botón de orden **Aceptar** (OK).

4. Desplace el puntero del ratón al icono que aparece en la intersección de las reglas vertical y horizontal. Pulse y mantenga pulsando el botón del ratón y arrastre la cruz de las reglas a la esquina superior izquierda de la página.
5. Seleccione la herramienta **Rectángulo** y sitúe el puntero en la marca 3 ½ pulgadas de la regla horizontal y en la marca de 3 pulgadas de la regla vertical. Dibuje un rectángulo que se extienda desde el punto hasta la marca de 9 pulgadas en la regla horizontal y hasta la marca 5 ½ pulgadas en la regla vertical.
6. El diseño del tejado de la casa precisa tres pasos que involucran la restricción de líneas y la unión automática de líneas para formar un polígono. Seleccione la herramienta **Mano alzada** y sitúe el puntero en la esquina superior izquierda de la “*casa*”. Extienda una línea hacia arriba y hacia la derecha con un ángulo de 45 grados. Recuerde pulsar la tecla CTRL mientras dibuja para restringir esta línea automáticamente al ángulo correcto. Finalice la línea en la marca de 6 pulgadas de la regla horizontal y en la marca de 2 ½ pulgadas de la regla vertical.
7. Extienda una línea hacia arriba y hacia la izquierda con un ángulo de 135 grados. Recuerde pulsar la tecla CTRL mientras dibuja para restringir esta línea automáticamente al ángulo correcto. Cuando alcance la marca de 8 pulgadas en la regla horizontal y la marca de 2 ½ pulgadas en la regla vertical, finalice la línea con una doble pulsación de modo que pueda continuar con otro segmento de línea.
8. Mientras continúa pulsando la tecla CTRL, extienda una línea horizontal de vuelta a la primera línea diagonal y dé una pulsación para finalizarla. Los dos segmentos de línea deben conectarse y formar un polígono que constituya el “tejado” de la casa.
9. Para dibujar los siguientes objetos podrá necesitar incrementos de rejilla más pequeños.
10. Trace dos ventanas cuadradas para la casa.
11. Trace una puerta rectangular para la casa aproximadamente entre las dos ventanas que trazo con anterioridad. Utilice las reglas como ayuda para guiar los movimientos.
12. Termina de crear el dibujo libremente. Solo recuerda utilizar las herramientas que aprendiste.



## INSERCIÓN DE TEXTO

Las avanzadas funciones de manejo de texto que ofrece CorelDRAW permiten convertir el texto en una obra de arte. Es posible girar, inclinar, perfilar y editar un carácter o *cadena* de texto al igual que cualquier otro objeto. Estas proezas pueden realizarse con la extensa biblioteca de fuentes proporcionada con CorelDRAW o con las suministradas por cualquier otro fabricante. Las fuentes incluidas en Corel tienen un aspecto similar a cualquier fuente tipográfica y pueden imprimirse con cualquier impresora compatibles con Windows. Dado que las fuentes de Corel están en el formato TrueType estándar de Windows.

La herramienta Texto, la última de las cuatro herramientas básicas de CorelDRAW, está representada mediante la letra “A”. Esta herramienta se utiliza para insertar texto en los dibujos, de la misma forma que las herramientas Rectángulo y Elipse se utilizan para insertar objetos geométricos. El proceso de insertar texto en un dibujo implica hasta ocho pasos :

1. Selección de la herramienta Texto.
2. Elección de texto artístico o texto de párrafo.
3. Selección del punto de inserción
4. Introducción del texto.
5. Elección del tamaño del texto.
6. Definición del tipo de alineación.
7. Selección de la fuente y el estilo.
8. Ajuste del espaciado entre caracteres, palabras y líneas de texto.

### **Selección de la herramienta Texto**

Para poder introducir texto debe seleccionar la herramienta Texto.

En el siguiente ejercicio se ajustará en primer lugar el formato de la página y después se activará la herramienta Texto.

1. Si la página está en formato Vertical en lugar de Horizontal, seleccione la orden Preparar página en el menú Diseño y elija el formato Horizontal. Si realizó el dibujo del ejercicio anterior, el formato de página aún será el horizontal se debe a que CorelDRAW siempre “recuerda” el formato de página que se utilizó en el último dibujo a la hora de crear uno nuevo.
2. Por esta misma razón, abra el menú Ver y elija la orden Dibujo de líneas para desactivar esta función.

3. Seleccione la herramienta Texto situando el puntero sobre ella. A continuación, pulse y mantenga pulsado momentáneamente el botón del ratón. Los iconos representan los dos modos de uso de la herramienta Texto: para introducir texto artístico y para introducir texto de párrafo. La herramienta Texto artístico, a la izquierda, es la opción por defecto y permite la introducción de texto en modo artístico. La herramienta Texto de párrafo, situado a la derecha permite introducir texto en modo de párrafo.
4. Seleccione el modo de texto artístico pulsando sobre el icono de la izquierda. El puntero del ratón cambiará a una cruz.

### **Texto artístico y texto de párrafo**

El modo *texto artístico* está concebido para cadenas cortas de texto, tales como títulos, pies de ilustración y notas de hasta 8,000 caracteres de longitud. El modo *texto de párrafo* está pensado para bloques de texto más largos, por ejemplo **Folletos**. Un texto de párrafo es una suma de “**viñetas**” donde cada viñeta es un archivo de hasta 850 párrafos, dada uno de ellos de un máximo de 8.000 caracteres.

Para introducir texto artístico basta con pulsar sobre la herramienta Texto artístico en el punto de la página en el que se desea empezar el texto. Para introducir un texto de párrafo se arrastra una *viñeta*, la caja que contendrá el texto, desde el punto en el que se desea empezar el texto hasta donde se desea que termine. Las viñetas pueden encadenarse de modo que el archivo de texto de párrafo continúe de una viñeta a otra.

El texto de párrafo proporciona muchas de las características de un procesador de texto. El texto retorna automáticamente al final de cada línea; el texto puede estar justificado (alineado a la derecha, Izquierda), así como alineado al centro; se puede intercambiar texto con el Portapapeles, por el procesamiento de *cortar* y *pegar*; además del espaciado entre caracteres, palabras y líneas, es posible ajustar el espaciado entre párrafos y crear hasta ocho columnas.

#### **Selección del punto de inserción**

El *punto de inserción* es el lugar de la página donde se desea comenzar una nueva cadena de texto. Con CorelDRAW se puede introducir texto directamente en la página o teclearlo en una caja de diálogo especial, en la que también se selecciona sus atributos. Para seleccionar un punto de inserción y preparar el trabajo a realizar en los siguientes ejercicios siga los siguientes pasos.

1. Abra la caja de diálogo *Ajuste de rejilla y escala* desde el menú *Diseño* y asegúrese de que están habilitadas las funciones *Activar rejilla* y *Mostrar rejilla*; defina como valores Horizontal y Vertical de Frecuencia de la rejilla. Si en la pantalla no aparecen las reglas, seleccione orden *Reglas* en el menú *Ver*.
2. Sitúe el puntero del ratón en el centro de la parte superior del área de página, a 5 ½ pulgadas en horizontal y a ½ pulgadas en vertical (el cero debe estar en la parte superior de la página), pulse una vez el ratón y teclee **CorelDRAW**. Seleccione la orden *Editar texto*, o utilice el atajo de teclado con la combinación de teclas CTRL-MAYS-T. Aparecerá la caja de diálogo *Editar texto*.

El texto puede introducirse directamente en el área imprimible de la página de la ventana de CorelDRAW o en el área de introducción de texto de la caja de diálogo **Editar texto**.

### **Introducción de texto en la caja de diálogo Editar texto.**

Cuando se abre por primera vez la caja de diálogo Editar texto el área de introducción de texto es seleccionado automáticamente como puede verse por el punto de inserción . Si no está en el área de introducción de texto, pulse las teclas CTRL-MAYS-T para regresar a ella. Para introducir texto basta con empezar a teclear.

En el siguiente ejercicio introducirá texto de la siguiente manera:

1. Teclee **CorelDRAW** en la posición del punto de inserción si no está ya en la pantalla..
2. Pulse la tecla INTRO para comenzar una nueva línea, teclee **a su nombre**, y pulse otra vez la tecla INTRO.
3. En la tercera línea, teclee **alcance**.

### **Utilización del teclado para desplazarse**

En CorelDRAW el modo de utilizar el teclado para desplazarse en el área de introducción de texto puede ser diferente del modo utilizado en un procesador de texto. Siempre que tengan varias líneas en el área de introducción de texto, pueden emplearse las siguientes órdenes de teclado.

- La tecla **INTRO** para comenzar una nueva línea en el área de introducción de texto y empezar a escribir texto en ella. La ventana aceptará tantas líneas de texto como se generen, siempre y cuando no se supere el límite de 8,000 caracteres.
- La tecla **Flecha Abajo** para desplazar el punto de inserción una línea hacia abajo.
- La tecla **AV PAG** para desplazar el punto de inserción al extremo derecho de la última línea de texto.
- La tecla **Flecha Arriba** para desplazar el punto de inserción una línea hacia arriba.
- La tecla **RE PAG** para desplazar el punto de inserción a la primera línea de texto.
- La tecla **INICIO** para desplazar el punto de inserción al principio de la línea actual.
- La tecla **FIN** para desplazar el punto de inserción al final de la línea actual.
- La tecla **Flecha Derecha** para desplazar el punto de inserción un carácter hacia la derecha cada vez que se pulse.
- La tecla **Flecha Izquierda** para desplazar el punto de inserción un carácter hacia la izquierda cada vez que se pulse.
- La tecla **Retroceso** para borrar el carácter inmediatamente anterior al punto de inserción.

- La tecla **SUPR** para borrar el carácter inmediatamente posterior al punto de inserción.

### **Utilización del ratón**

También pueden realizarse algunas operaciones de introducción de texto utilizando el ratón:

- Localizar una línea de texto no visible en el área de introducción de texto utilizando la barra de desplazamiento de la derecha.
- Insertar texto en un punto dado pulsando con el ratón sobre ese punto.
- Seleccionar uno o más caracteres de una cadena de texto colocando el punto de inserción en el primer carácter que se quiera seleccionar y, a continuación, arrastrando el ratón sobre los caracteres deseados.
- Borrar una serie de caracteres seleccionados de este modo pulsando la tecla SUPR.

### **Alineación del Texto**

En la persiana **Texto** hay cinco opciones disponibles, tanto para texto artístico como de párrafo: **Izquierda, Centro, Derecha, Justificar y Ninguna.**

La **alineación izquierda** : Es la opción por defecto. Cuando se elige esta opción, el texto se alinea en la página como si el punto de inserción estuviera en el margen **izquierdo**.

La **alineación al centro** : Cuando se selecciona la alineación **Centro**, el punto de inserción inicial es el punto medio de cualquier cadena de texto que se escriba.

La **alineación derecha** : Cuando se selecciona alineación **derecha**, el texto se alinea sobre la página como si el punto inicial de inserción fuera el margen derecho.

**Texto no alineado** : Cuando se selecciona sin alineación, el texto aparece en la página exactamente igual que se introduce. Esta opción es útil cuando se desea añadir un espaciado especial al principio de una línea en una cadena de texto.

En este capítulo el principal objetivo es empezar a conocer los principios básicos de CorelDRAW ya que cuenta con varios módulos de los cuales nos enfocamos principalmente al módulo de CorelDRAW, en el cual se te enseña los principios del dibujo básico (dibujos sencillos).

Como veras CorelDRAW es una guía de enseñanza gráfico que no se requiere de una experiencia gráfica o conocimientos de las computadoras.

# VISUAL BASIC

## *INTRODUCCIÓN.*

Bienvenido a Microsoft® Visual Basic®: la manera más rápida y fácil de crear aplicaciones potentes para los sistemas operativos Microsoft® Windows®. El sistema de programación Visual Basic permite crear aplicaciones robustas y útiles que hagan un uso completo de la interfaz gráfica de usuario (GUI).

Visual Basic le ayuda a ser más productivo proporcionando herramientas apropiadas para los diferentes aspectos del desarrollo de GUI. Puede crear la interfaz gráfica de usuario de su aplicación colocando objetos de manera gráfica. Usted define las propiedades de estos objetos para concretar su apariencia y comportamiento. Después hace que esta interfaz se relacione con el usuario escribiendo el código que responde a los eventos que ocurren en la interfaz.

Usando Visual Basic, puede crear aplicaciones potentes, totalmente funcionales.

- Las funciones de acceso a datos le permiten crear aplicaciones de base de datos y preprocesadoras para los formatos de base de datos más conocidos.
- OLE le permite usar la funcionalidad proporcionada por otras aplicaciones, como el procesador de textos Microsoft® Word para Windows®, la hoja de cálculo Microsoft® Excel y el sistema de planificación de proyectos financieros Microsoft® Project.
- Su aplicación terminada es un archivo .EXE real que en tiempo de ejecución utiliza una biblioteca de vínculos dinámicos (DLL) que se puede distribuir libremente.



## **SU PRIMERA APLICACIÓN EN VISUAL BASIC.**

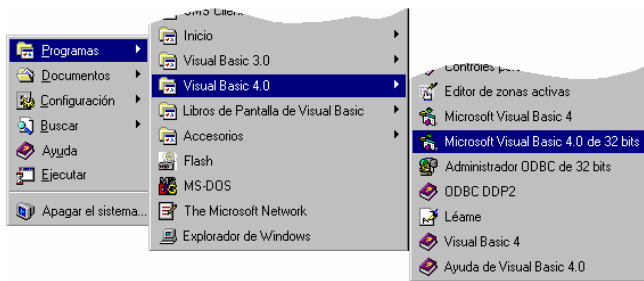
Sólo se necesitan unos minutos para construir su primera aplicación con Visual Basic. Puede crear la interfaz “dibujando” controles, tales como cuadros de texto y botones de comando, en un formulario. Después definir las propiedades del formulario y de los controles, como el título, el color y el tamaño. Finalmente, escribir el código para dar vida a la aplicación. Los pasos básicos que tienen lugar en la creación de su primera aplicación le enseñarán los principios que usará con cualquier otra aplicación que desarrolle.

## **PRIMEROS PASOS CON VISUAL BASIC.**

Cuando ejecuta el programa de instalación de Visual Basic, **Instalar** le permite colocar los elementos de programa en un grupo de programas existente o crear un nuevo grupo de programas y nuevos elementos de programa para Visual Basic en Windows. Entonces está preparado para iniciar Visual Basic desde Windows.

### **Para iniciar Visual Basic desde Windows**

1. Haga clic en Inicio en la barra de tareas.
2. Seleccione Programas y después Visual Basic 4.0.

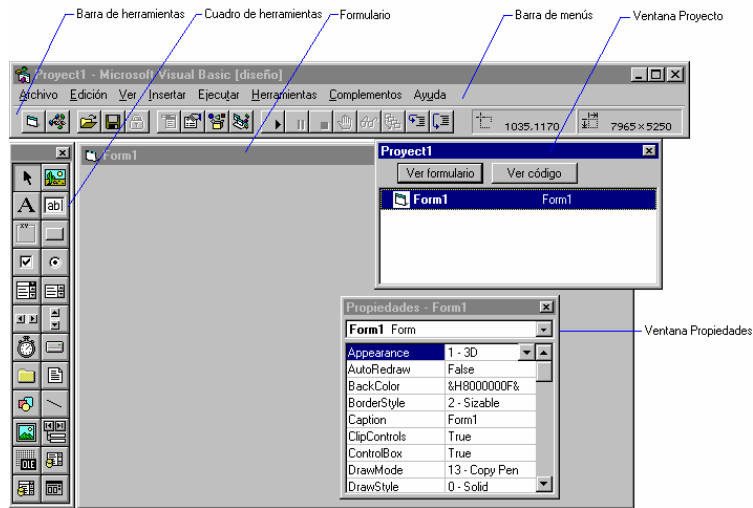


– O bien –

3. Haga clic en Inicio en la barra de tareas.
4. Seleccione Programas.
5. Use el Explorador de Windows para encontrar el archivo ejecutable de Visual Basic (VB32.EXE).
6. Haga doble clic en el icono de Visual Basic.

También puede crear un acceso directo a Visual Basic y hacer doble clic en dicho acceso.

La primera vez que inicie Visual Basic, verá la interfaz del entorno de programación, como se muestra en la figura










**Figura Entorno de programación de Visual Basic**











## **ELEMENTOS DE LA INTERFAZ.**

La interfaz de Visual Basic consta de los siguientes elementos.

### **Barra de herramientas**

Proporciona un acceso rápido a los comandos usados normalmente en el entorno de programación. Haga clic en un botón de la barra de herramientas para llevar a cabo la acción representada por ese botón.

<b>Botón de la barra de herramientas</b>	<b>Acción</b>	<b>Equivalente en Menú</b>
	Crea un formulario nuevo	Comando <b>Formulario</b> en el menú <b>Insertar</b>
	Crea un módulo nuevo	Comando <b>Módulo</b> en el menú <b>Insertar</b>
	Abre un proyecto existente	Comando <b>Abrir Proyecto</b> en el menú <b>Archivo</b>
	Guarda el proyecto actual	Comando <b>Guardar Proyecto</b> del menú <b>Archivo</b>
	Bloquea y desbloquea los controles del formulario activo	Comando <b>Bloquear Controles</b> en el menú <b>Edición</b>
	Presenta el Editor de menús	Comando <b>Editor de menús</b> en el menú <b>Herramientas</b>
	Presenta la ventana Propiedades	Comando <b>Propiedades</b> en el menú <b>Ver</b>

	Presenta el Examinador de objetos	Comando <b>Examinador de objetos</b> en el menú <b>Ver</b>
	Presenta la ventana Proyecto	Comando <b>Proyecto</b> en el menú <b>Ver</b>
	Inicia una aplicación en modo diseño	Comando <b>Iniciar</b> en el menú <b>Ejecutar</b>
	Usado para depuración, detiene la ejecución de un programa mientras se está ejecutando	Comando <b>Interrumpir</b> en el menú <b>Ejecutar</b>
	Detiene la ejecución de una aplicación y vuelve al modo diseño	Comando <b>Terminar</b> en el menú <b>Ejecutar</b>
	Alternar puntos de ruptura en la línea actual	Comando <b>Alternar puntos de ruptura</b> en el menú <b>Ejecutar</b>
	Presenta el valor de la selección actual en la ventana Código	Comando <b>Inspección instantánea</b> en el menú <b>Herramientas</b>
	Presenta la estructura de las llamadas activas	Comando <b>Llamadas</b> en el menú <b>Herramientas</b>
	Hace el seguimiento del código línea a línea pasando por los procedimientos	Comando <b>Paso a paso por instrucciones</b> en el menú <b>Ejecutar</b>
	Ejecuta una línea de código, sentencia o procedimiento, en la ventana Código	Comando <b>Paso a paso por procedimientos</b> en el menú <b>Ejecutar</b>

### **Caja de herramientas**

Proporciona un conjunto de herramientas que puede usar durante el diseño para colocar controles en un formulario.

### **Barra de menú**

Presenta los comandos que se usan para construir su aplicación.

### **Formulario**

Es una ventana que se personaliza, como la interfaz de su aplicación. Agregue controles, gráficos e imágenes a un formulario para crear la apariencia que desea.

### **Ventana Proyecto**

Enumera los formularios y módulos de su proyecto actual. Un *proyecto* es una colección de archivos que se usan para generar una aplicación.



## **Ventana Propiedades**

Enumera los valores de las propiedades del formulario o control seleccionado. Una *propiedad* es una característica de un objeto, como su tamaño, título o color.

## **Examinador de objetos**

Disponible en el menú **Ver**, el Examinador de objetos enumera los objetos que puede usar en su proyecto y le proporciona una manera rápida de desplazarse a través de su código. Puede usar el Examinador de objetos para explorar objetos en Visual Basic y en otras aplicaciones, ver qué métodos y propiedades están disponibles para esos objetos y pegar procedimientos de código en su aplicación.

**Nota** También puede agregar funciones a la interfaz de Visual Basic usando un programa llamado *Complemento*. Los complementos, disponibles de Microsoft y de terceros, pueden proporcionar funciones como control de código origen, que le permiten afrontar proyectos de desarrollo en grupo.

## **Menús de contexto**

El entorno de desarrollo de Visual Basic proporciona menús de contexto que contienen accesos directos a acciones realizadas frecuentemente. Para abrir un menú de contexto, haga clic con el botón primario del *Mouse* en el objeto que está usando. La lista específica de accesos directos disponibles en los menús de contexto depende del componente del entorno en el que hace clic con el botón derecho del *Mouse*. Por ejemplo, el menú de contexto presentado cuando hace clic con el botón derecho en la caja de herramientas le permite presentar el cuadro de diálogo Controles personalizados, o hacer que la Caja de Herramientas esté siempre visible, o no.

## ***PASOS PARA LA CREACIÓN DE UNA APLICACIÓN CON VISUAL BASIC.***

Hay tres pasos importantes en la creación de una aplicación Visual Basic para Windows:

1. Crear la interfaz.
2. Definir las propiedades.
3. Escribir el código.


Para ver como se hace esto, continúe hasta el final de esta sección para crear una aplicación simple que consiste en un cuadro de texto y un botón de comando. Cuando haga clic en el botón de comando, en el cuadro de texto aparecerá el mensaje “¡Hola, mundo!”.

## CREACIÓN DE LA INTERFAZ.

Los formularios son la base de la creación de la interfaz de una aplicación. Puede usar formularios para agregar ventanas y cuadros de diálogo a su aplicación. También los puede usar como contenedores de los elementos que no son parte visible de la interfaz de la aplicación. Por ejemplo, puede tener un formulario en su aplicación que sirva como contenedor de los gráficos que piensa presentar en otros formularios.

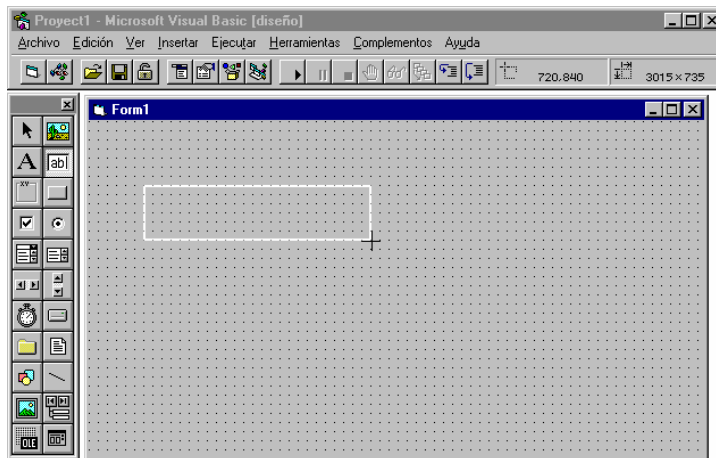
El primer paso en la generación de un aplicación Visual Basic es la creación de los formularios que serán la base de la interfaz de su aplicación. Después coloque en los formularios los objetos que componen la interfaz que ha creado. Para esta primera aplicación, usará dos controles de la caja de herramientas.

### BotónControl

-  TextBox
-  CommandButton

### Para colocar un control usando la caja de herramientas

1. Haga clic en la herramienta del control que ha decidido usar, en este caso, el cuadro de texto.
2. Mueva el puntero dentro del formulario. El puntero toma forma de cruz, como se muestra en la figura.



**Figura Colocar un cuadro de texto usando la caja de herramientas**

3. Sitúe la cruz donde quiera tener la esquina superior izquierda del control.
4. Arrastre la cruz hasta que el control tenga el tamaño que quiera. (*Arrastrar* quiere decir mantener presionado el botón primario del *Mouse* mientras mueve un objeto con el *Mouse*).
5. Suelte el botón del *Mouse*.  
El control aparece en el formulario.

Otra manera simple de agregar un control a un formulario es hacer doble clic en dicho control en la caja de herramientas. Así se crea un control con su tamaño predeterminado situado en el centro del formulario.

### **Cambiar el tamaño, mover y bloquear controles.**

Note que en las esquinas del control aparecen pequeños cuadros rectangulares llamados *controladores de tamaño*; en el paso siguiente usará estos controladores de tamaño para modificar el tamaño del control. También puede usar el *Mouse*, el teclado y comandos de menús para mover controles, bloquear y desbloquear la posición de los controles y ajustar sus posiciones.

#### **Para modificar el tamaño de un control**

1. Seleccione el control al que quiere modificar su tamaño haciendo clic en él con el *Mouse*.

En el control aparecen los controladores de tamaño.

2. Posicione el *Mouse* sobre un controlador de tamaño y arrástrelo hasta que el control tenga el tamaño que decida.

Los controladores de las esquinas modifican el tamaño de los controles horizontal y verticalmente, mientras que los controladores de los lados modifican el tamaño en una sola dirección.

3. Suelte el botón del *Mouse*.

– O bien –

Use MAYÚSCULAS con las teclas flecha para modificar el tamaño del control seleccionado.

#### **Para mover un control**

- Use el *Mouse* para arrastrar el control a otra posición del formulario.  
– O bien –

Use la ventana **Propiedades** para modificar las propiedades Top y Left.

Cuando se selecciona un control, puede usar CTRL con las teclas de flechas para mover el control una unidad de cuadrícula cada vez. Si no está usando la cuadrícula, el control se mueve un píxel cada vez.

#### **Para bloquear la posición de todos los controles**

- En el menú **Edición**, elija **Bloquear controles**.  
– O bien –

Haga clic en el botón “Bloquear controles” de la barra de herramientas.

Así se bloquean todos los controles del formulario en su posición actual para evitar que los mueva inadvertidamente una vez que los tiene en la posición deseada. Sólo se bloquean los controles del formulario seleccionado; los controles de otros formularios no se ven afectados. Este es un comando

alternante, así que puede usar el mismo comando para desbloquear la posición de los controles.

### Para ajustar la posición de controles bloqueados

- Puede “arrastrar” el control enfocado manteniendo presionada la tecla CTRL y presionando la tecla flecha apropiada.
  - O bien –
- Puede modificar las propiedades Top y Left del control en la ventana Propiedades.

Ya tiene la interfaz de la aplicación “¡Hola, mundo!”, como se muestra en la figura.

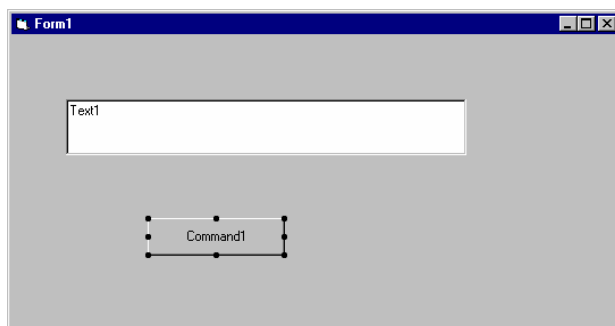


Figura Interfaz de la aplicación “¡Hola, mundo!”

### DEFINICIÓN DE PROPIEDADES.

El siguiente paso es definir las propiedades de los objetos que ha creado. La ventana Propiedades (ver figura) proporciona una manera fácil de definir las propiedades de todos los objetos de un formulario. Para abrir la ventana Propiedades, elija el comando **Propiedades** en el menú **Ver**, haga clic en el botón “Propiedades” en la barra de herramientas o use el menú de contexto del control.

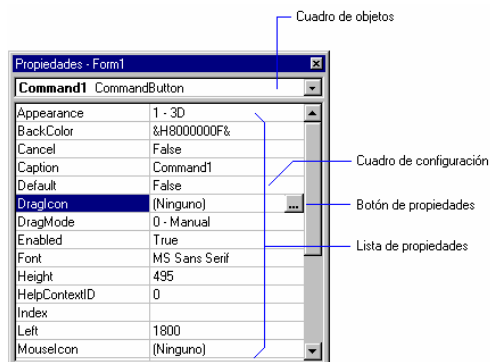


Figura Ventana Propiedades

La ventana Propiedades consiste en los siguientes elementos:

- Cuadro de objetos — Presenta el nombre del objeto para el cual se definen las propiedades. Para presentar la lista de objetos del formulario actual haga clic en la flecha situada a la derecha del cuadro de objetos.
- Lista de propiedades —La columna izquierda presenta todas las propiedades del objeto seleccionado. Puede editar y ver los valores en la columna derecha.

### Para definir propiedades en la ventana Propiedades

1. En el menú **Ver**, elija **Propiedades** o haga clic en el botón “Propiedades” de la barra de herramientas.

La ventana Propiedades presenta los valores del formulario o control seleccionado.

2. En la lista de propiedades, seleccione el nombre de la propiedad.
3. En la columna derecha, escriba o seleccione el nuevo valor de la propiedad.

Las propiedades enumeradas tienen una lista predefinida de valores. Puede presentar la lista haciendo clic en la flecha hacia abajo en la derecha del cuadro Valores o puede recorrer la lista haciendo doble clic en el elemento de la lista.

En la aplicación ejemplo “¡Hola, mundo!”, Necesitará modificar los valores de tres propiedades. Use los valores predefinidos para todas las demás propiedades.

<b>Objeto</b>	<b>Propiedad</b>	<b>Valor</b>
Form	Caption	¡Hola, mundo!
TextBox	Text	(Vacío)
CommandButton	Caption	Aceptar

### Definición de la propiedad Icono

Todos los formularios de Visual Basic tienen un icono genérico, predeterminado, que aparece cuando minimiza un formulario. Sin embargo, probablemente cambiará este icono por otro que ilustre el uso del formulario o de su aplicación. Para asignar un icono a un formulario, defina la propiedad Icono de ese formulario. Puede usar iconos de 32 x 32 píxeles que son estándar en Windows y que se usan también en Windows 95 y Windows NT, así como los iconos de 16 x 16 píxeles usados en Windows 95.

### **ESCRITURA DE CÓDIGO.**

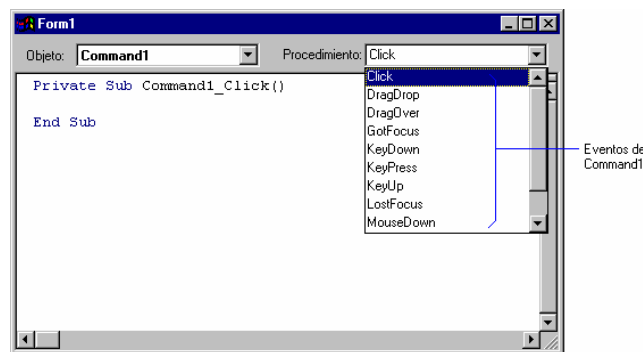
La ventana Código es donde escribe el código Visual Basic de su aplicación. El código consiste en sentencias del lenguaje, constantes y declaraciones.

Usando la ventana Código, puede ver y editar rápidamente cualquier código de su aplicación.

### Para abrir la ventana Código

- Haga doble clic en el formulario o control para el que se desea escribir código.
    - O bien –
- En la ventana Proyecto, seleccione el nombre de un formulario o módulo y elija el botón “Ver código”.

La figura muestra la ventana Código que aparece cuando hace doble clic en el control Commandbutton y los eventos para ese comando.



**Figura Ventana Código**

Puede elegir presentar todos los procedimientos en la misma ventana Código o presentar un solo procedimiento a la vez.

### Para presentar todos los procedimientos en la misma ventana Código

1. En el menú **Herramientas**, seleccione el cuadro de diálogo **Opciones**.
2. En la sección Editor del cuadro de diálogo **Opciones**, seleccione la casilla de verificación a la izquierda de “Ver todos los módulos”. La casilla de verificación a la izquierda de “Separador de procedimientos” agrega o quita una línea separadora entre los procedimientos.

### Para presentar un procedimiento a la vez en la ventana Código

1. En el menú **Herramientas**, seleccione el cuadro de diálogo **Opciones**.
2. En la sección Editor del cuadro de diálogo **Opciones**, borre la casilla de verificación “Ver módulos completos”.

La ventana Código incluye los siguientes elementos:

- Cuadro de objetos, presenta el nombre del objeto seleccionado. Haga clic en la flecha a la derecha del cuadro de lista para presentar la lista de todos los objetos del formulario.
- Cuadro de lista de procedimientos, lista los procedimientos de un objeto. El cuadro presenta el nombre del procedimiento seleccionado, en este caso,

Clic. Elija la flecha a la derecha del cuadro para presentar todos los procedimientos del objeto.

### Creación de procedimientos de evento

El código de una aplicación Visual Basic se divide en pequeños bloques llamados *procedimientos*. Un *procedimiento de evento*, como los que creará aquí, contiene el código que se ejecuta cuando ocurre un evento (como cuando un usuario hace clic en un botón). Un procedimiento de evento de control combina el nombre del control (especificado en la propiedad Name), un subrayado y el nombre del evento. Por ejemplo, si quiere que un botón de comando llamado Command1 invoque un procedimiento de evento cuando se hace clic en él, use el procedimiento Comando1\_Click.

### Para crear un procedimiento de evento

1. En el cuadro de lista "Objeto", seleccione el nombre de un objeto del formulario activo. (El formulario *activo* es el formulario que tiene actualmente tiene el enfoque).

Para este ejemplo, elija el botón de comando "Command1".

2. En el cuadro de lista "Procedimiento", seleccione el nombre de un evento del objeto seleccionado.

Aquí, el procedimiento Clic ya está seleccionado porque es el procedimiento predefinido para un botón de comando. Note que en la ventana Código se presenta una *plantilla* para el procedimiento de evento.

3. Escriba el siguiente código entre las sentencias **Sub** y **End Sub**:

```
Text1.Text = "¡Hola, mundo!"
```

El procedimiento de evento debería parecerse a esto:

```
Private Sub Comando1_Click ()
```

```
    Text1.Text = "¡Hola, mundo!"
```

```
End Sub
```

Notará que aquí el código simplemente cambia la propiedad Text del control llamado Text1 para que se lea "¡Hola, mundo!". La sintaxis de este ejemplo tiene la forma *objeto.propiedad*, donde Text1 es el objeto y Text es la propiedad. Puede usar esta sintaxis para modificar los valores de las propiedades de cualquier formulario o control en respuesta a eventos que se producen mientras se ejecuta su aplicación.

## **EJECUCIÓN DE LA APLICACIÓN.**

Para ejecutar la aplicación, elija **Iniciar** en el menú **Ejecutar**, haga clic en el botón “Iniciar” de la barra de herramientas o presione F5. Haga clic en el botón de comando que ha creado en el formulario y verá “¡Hola, mundo!” Presentado en el cuadro de texto.

## **TRABAJAR CON APLICACIONES EJEMPLO.**

Visual Basic le proporciona una riqueza de herramientas más allá de las que ha usado en su primera aplicación, así que pronto usará otras muchas funciones para administrar y personalizar sus aplicaciones. Revisar las aplicaciones muestra puede ser un excelente modo de aprender más sobre Visual Basic. La sencilla aplicación siguiente ilustra lo fácil que es agregar funciones especiales a su aplicación.

## **ANIMACIÓN SIMPLE.**

La aplicación Mariposa demuestra cómo se puede usar un *control Image* para crear una simple mariposa animada. La aplicación consiste en tres controles Image. El primer control Image está cargado con un mapa de bits mariposa “abierta” y el segundo con un mapa de bits “cerrada” (vea la figura). Cuando el tercer control Image se carga primeramente con el mapa de bits “abiertas” y después con el mapa de bits “cerradas”, la mariposa parece que abre y cierra las alas mientras se mueve a través de la pantalla.



**Figura Mapa de bits mariposa(estados abierta y cerrada)**

El *control Timer* proporciona un evento Timer (cronómetro) que puede usar para mover, conmutar y cargar mapas de bits a intervalos específicos. El temporizador se mide en milisegundos, así que este ejemplo usará un valor de intervalo de 500 para hacer que las alas se abran y se cierren cada 0,5 segundos.

Para hacer que la mariposa “vuele” por la pantalla, use el método **Move**. (Un *método* es una función Visual Basic que actúa sobre un objeto particular). En este caso, el método **Move** hace que el control Image se mueva a través de la pantalla. La sintaxis para especificar un método (*objeto.método*) es similar a la sintaxis de definir propiedades (*objeto.propiedad*). El método **Move** le permite especificar cuánto se mueve por la pantalla el mapa de bits horizontal y verticalmente.



## Más información

Busque Ayuda acerca de *Move*.

## Creación del proyecto

Empiece a crear esta aplicación eligiendo **Proyecto Nuevo** en el menú **Archivo**. Visual Basic crea un proyecto nuevo y presenta un formulario nuevo. Para dibujar la interfaz, use controles Image, un control Timer y un control CommandButton.

## Botón Control



Image

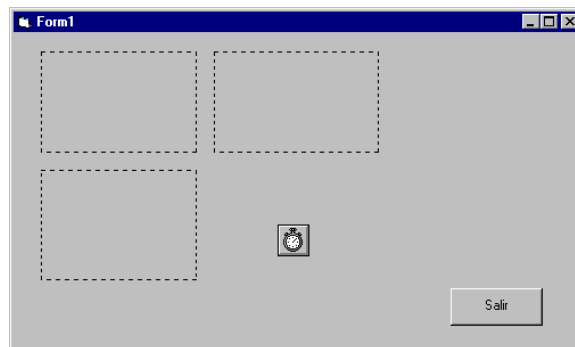


Timer



CommandButton

Use la caja de herramientas para dibujar en el formulario los controles Image, un control Timer y un botón de comando (o control CommandButton), como se muestra en la figura.



**Figura** Formulario de la aplicación muestra Mariposa

**Nota** Puede colocar el control Timer en cualquier lugar del formulario, no será visible cuando se ejecute el programa.

## Definición de propiedades

En la ventana Propiedades, defina las propiedades de los objetos de acuerdo con la siguiente tabla. Los mapas de bits de la mariposa se pueden encontrar en el directorio SAMPLES\FIRSTAPP. Use los valores predeterminados para las demás propiedades.

Objeto	Propiedad	Valor
Form	Caption	
WindowState		
BackColor		
ScaleMode	Mariposa	
2-Maximized		
(White)		

3-Pixel		
Timer	Interval	500
Image1	Name	
Picture		
Visible	AlasAbiertas	
BFLY1.BMP		
<b>False</b>		
Image2	Name	
Picture		
Visible	AlasCerradas	
BFLY2.BMP		
<b>False</b>		
Image3	Name	
Picture		
Visible	Principal	
BFLY1.BMP		
<b>True</b>		
CommandButton	Caption	&Salir

La definición de la propiedad WindowState del formulario como 2-Maximized y de la propiedad BackColor como White borra el fondo sobre el que se presenta el mapa de bits. La propiedad Visible determina si el control puede verse en ejecución. Como dos de los controles Image (AlasAbiertas y Alas Cerradas) son contenedores de mapas de bits, puede hacerlos invisibles en ejecución definiendo su propiedad Visible como **False**. La propiedad Visible del tercer control Image (Principal) es **True**.

### Escritura del código de eventos

Todo el código de la aplicación está contenido en los procedimientos de evento Form\_Load, Command1\_Click y Timer1\_Timer. Haga doble clic en el formulario o control correspondiente y escriba el código para cada procedimiento de evento.

Agregue este código al procedimiento de evento Form\_Load:

```
Private Sub Form_Load ()
    Command1.Move 10, 10
End Sub
```

Agregue este código al procedimiento de evento Comando1\_Click:

```
Private Sub Command1_Click ()
    Unload Me
```

```
End
Termina la aplicación.
```

```
End Sub
```

Agregue este código al procedimiento de evento Timer1\_Timer:

```
Private Sub Timer1_Timer ()
Static PickBmp As Integer
    Main.Move Principal.Left + 20, Principal.Top - 5
    If PickBmp Then
        Main.Picture = AlasAbiertas.Picture      ' Mariposa abierta.
    Else
        Main.Picture = AlasCerradas.Picture      ' Mariposa cerrada.
    End If
    PickBmp = Not PickBmp                        ' Alterna el valor.
End Sub
```

La variable local PickBmp determina qué mapa de bits está cargado en Principal.Picture. La declaración de PickBmp como **Static** mantiene su valor (**True** o **False**) entre los eventos del temporizador. PickBmp es una variable entera y el valor predefinido para una variable entera es 0, que es igual que **False**.

Cada 0,5 segundos, cuando se genera el evento, el procedimiento usa el operador **Not** para alternar el valor de PickBmp entre **True** y **False**; esto a su vez controla si el control Image principal presenta el control Image AlasAbiertas o el AlasCerradas.

Finalmente, el método **Move** provoca que el control Image principal se mueva hacia arriba y hacia la derecha de la pantalla especificando nuevos valores para las propiedades Top y Left del control.

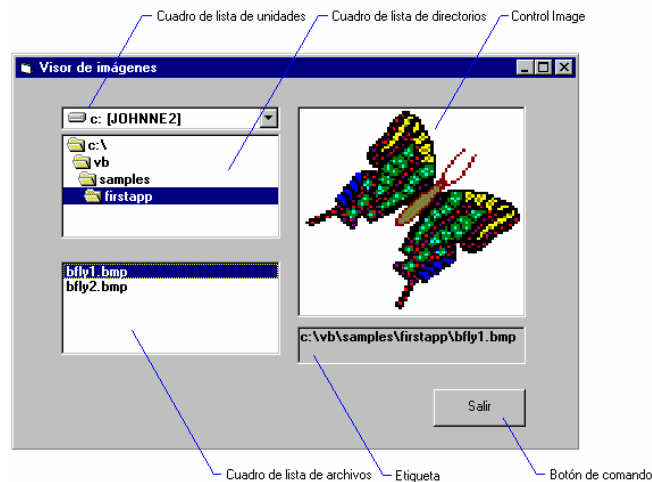
**Nota** Para controlar cómo la mariposa se mueve por la pantalla puede experimentar con valores diferentes de las propiedades Top y Left.

**Guardar el proyecto**

Para finalizar el trabajo en su aplicación elija **Guardar proyecto** en el menú **Archivo**. Visual Basic le pedirá que guarde el formulario y el proyecto separadamente. Un posible nombre para el proyecto es “Aplicación de la mariposa voladora”. Tanto Windows 95 como Windows NT le permiten usar nombres de hasta 255 caracteres de longitud y los nombres de archivo pueden contener espacios. Con Microsoft Windows 3.1 está limitado a nombres de archivo de ocho caracteres con una extensión de tres caracteres.

## UN VISOR DE IMÁGENES.

Sólo con unas líneas de código puede crear una útil aplicación que le permita ver mapas de bits, iconos y metarchivos Windows. La aplicación consiste en seis controles: un cuadro de lista de unidades, un cuadro de lista de directorios, un control Image, un cuadro de lista de archivos, una etiqueta y un botón de comando. El formulario del Visor de imágenes se muestra en la figura.



**Figura** Formulario de la aplicación muestra el Visor de imágenes






Cuando ejecuta la aplicación, puede presentar una imagen haciendo doble clic en el nombre del archivo del mapa de bits, icono o metarchivo del cuadro de lista de archivos.

Los cuadros de lista de unidades, directorios y archivos funcionan conjuntamente, del mismo modo que funcionan en otros cuadros de diálogo basados en Windows. Seleccione la unidad y después use la barra de desplazamiento para seleccionar el directorio que desee. Cuando selecciona el directorio, en el cuadro de lista de archivos se presenta la lista de los archivos de ese directorio.

## Creación del proyecto y definición de propiedades

Para crear esta aplicación, comience eligiendo **Proyecto nuevo** en el menú **Archivo**; después coloque los controles en el formulario de acuerdo con la figura Usará los controles de la siguiente tabla.

### BotónControl

	DriveListBox
	DirectoryListBox
	FileListBox
	Image
	Label
	CommandButton

Después de diseñar el formulario, necesitará definir las siguientes propiedades. (Use los valores predeterminados para las demás propiedades y objetos).

Objeto	Propiedad	Valor
Formulario	Caption	Visor de imágenes
Cuadro de lista de archivos	Pattern	*.BMP; *.WMF; *.ICO
Imagen	BorderStyle	
Name		
Stretch	1-Fixed Single	
Open		
<b>True</b>		
Etiqueta	BorderStyle	
Caption	1-Fixed Single	
(Vacío)		
Botón de comando	Caption	&Salir

La propiedad Pattern especifica los tipos de archivos presentados en el cuadro de lista de archivos. Como quiere ver mapas de bits, iconos y metarchivos, puede especificar sus extensiones de nombre de archivo para prevenir que se presenten otros tipos de archivos.

En la aplicación Visor de imágenes, creará procedimientos de evento para cinco controles diferentes.

### Evento Carga de formulario

El evento Form\_Load define la unidad y la ruta como la unidad y directorio donde está cargada la aplicación. Necesitará agregar el siguiente código al procedimiento de evento Form\_Load:

```
Private Sub Form_Load ()
    Drive1.Drive = App.Path
    Dir1.Path = App.Path
```

End Sub

### **Evento Click en botón de comando**

El evento Click del botón de comando termina la aplicación. Necesitará agregar el siguiente código al procedimiento de evento Comand1\_Click:

```
Private Sub Comand1_Click ()  
    Unload Me  
    End                ' Termina la aplicación.  
End Sub
```

### **Eventos Cambio en las listas de unidades y de directorios**

Para hacer que los cuadros de lista de unidades, de directorio y de archivo funcionen conjuntamente, necesitará agregar el siguiente código a los procedimientos de evento Drive1\_Change y Dir1\_Change:

```
Private Sub Drive1_Change ()  
    Dir1.Path = Drive1.Drive                ' Cambia ruta directorio.  
End Sub
```

```
Private Sub Dir1_Change ()  
    file1.Path = Dir1.Path                ' Cambia ruta archivos.  
End Sub
```

### **Evento Doble clic en el cuadro de lista de archivos**

El procedimiento de evento Archivo1\_DbIcIck del cuadro de lista de archivos presenta la ruta completa del archivo seleccionado en el control Label y la imagen misma en el control Image. Necesitará el siguiente código:

```
Private Sub file1_DbIcIck ()  
    If Right(file1.Path, 1) <> "\" Then  
        Label1.Caption = file1.Path & "\" & file1.FileName
```

```

Else
    ' Si es el directorio raíz
    Label1.Caption = file1.Path & file1.FileName

End If

Form1.Open.Picture = LoadPicture(Etiqueta1.Caption)

End Sub

```

Observe cómo se construye la ruta completa:

- File1.Path devuelve la unidad y ruta del directorio, "\" agrega un separador barra contraria y File1.FileName devuelve el nombre del archivo.
- La función **Right** comprueba si la ruta es el directorio raíz (\); si no, el nombre de ruta se asigna al título de la etiqueta. Si la ruta es el directorio raíz la barra contraria se omite.
- La función **LoadPicture** carga el archivo de mapa de bits especificado por el nombre del título de la etiqueta.

### Más información

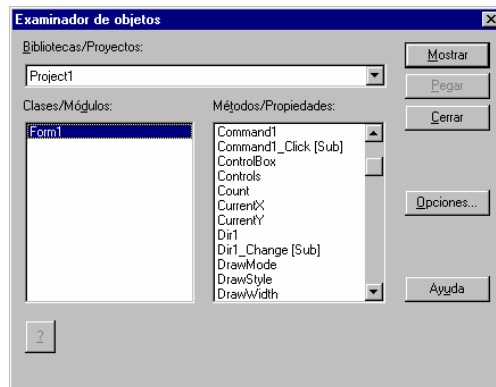
Busque Ayuda sobre *Right* y *LoadPicture*.

### Trabajar con múltiples procedimientos

Cuando crea múltiples procedimientos para una aplicación como el Visor de imágenes, querrá moverse entre ellos a menudo según los escribe. Para moverse entre los procedimientos en la ventana Código, use el Examinador de objetos, como se muestra en la figura.

### Para moverse entre los procedimientos de una aplicación

1. En el menú **Ver**, elija **Examinador de objetos** o presione F2.



**Figura Examinador de objetos**

2. Seleccione el proyecto que contiene el procedimiento en el cuadro Bibliotecas/Proyectos.
3. Seleccione el nombre del objeto y el procedimiento en el cuadro de diálogo y después elija el botón “Mostrar”.

En la ventana Código se presenta el procedimiento seleccionado.

### **Para recorrer los procedimientos de un módulo**

- Presione CTRL+FLECHA ARRIBA para presentar el procedimiento anterior o CTRL+FLECHA ABAJO para presentar el procedimiento siguiente.

### **Más información**

Para obtener la lista de teclas que puede usar en la ventana Código, busque Ayuda acerca de *ventana Código*.

## ***DISTRIBUCIÓN DE SUS APLICACIONES.***

Antes de verificar una aplicación, es buena práctica guardar su trabajo. Si su programa no funciona como se preveía, compruebe que ha escrito bien las líneas de código. Mientras que Visual Basic detecta los errores de sintaxis según se escriben, algunos errores, como escribir código en un procedimiento de evento equivocado, sólo aparecerán cuando ejecute el programa.

Una vez que terminado un proyecto, puede guardar la aplicación como archivo ejecutable (.EXE): En el menú **Archivo**, elija el comando **Crear archivo EXE**. Una vez que ha guardado su aplicación como archivo ejecutable, está listo para distribuirla.

**Nota** Visual Basic se suministra con SETUP1.VBP, un programa de instalación ejemplo que puede usar como plantilla para los programas de instalación que escriba para cualquier aplicación Visual Basic que quiera distribuir. El código origen de SETUP1.VBP se encuentra en el subdirectorio \SETUPKIT\SETUP1 del directorio raíz de Visual Basic. Para obtener más información, busque Ayuda acerca del *Kit de instalación*.



## UNIDAD 2. CREACIÓN Y USO DE LOS CONTROLES.

El primer paso en la creación de una aplicación de Visual Basic es diseñar la interfaz: los formularios, los controles y otros objetos que el usuario verá y usará. Después se definen las propiedades y se escribe el código Visual Basic para activar la interfaz.

Los controles se usan para obtener datos del usuario y presentar la salida. Algunos de los controles incluyen cuadros de texto, botones de comando y cuadros de lista. Otros controles le permiten acceder a otras aplicaciones y procesar datos como si la aplicación remota formase parte de su código. Cada control tiene su propio juego de propiedades, métodos y eventos.



### CONTROLES DE VISUAL BASIC.
















La Caja de herramientas de Visual Basic contiene las herramientas para dibujar controles en los formularios. Cada herramienta de la Caja de herramientas (Figura) representa un control.








**Figura La caja de herramientas de Visual Basic**

La siguiente tabla presenta un resumen de los controles de Visual Basic que se encuentran en la Caja de herramientas. Algunos de estos controles pertenecen a versiones anteriores de Visual Basic.

Botón	Nombre de clase	Descripción
	Pointer	El Pointer (puntero) proporciona una manera de mover y cambiar el tamaño de formularios y controles. (Tenga en cuenta que este no es un control).
	PictureBox	El control PictureBox (cuadro de imagen) presenta mapas de bits, iconos o metarchivos de Windows. También puede presentar texto o actuar como contenedor visual de otros controles.

	Label	Un control Label (etiqueta) presenta texto con el que el usuario no puede interactuar ni modificar.
	TextBox	El control TextBox (cuadro de texto) proporciona un área para escribir o presentar texto.
	Frame	El control Frame (marco) proporciona un contenedor visual y funcional para controles.
	CommandButton	El CommandButton (botón de comando) ejecuta una acción o comando cuando el usuario lo selecciona.
	CheckBox	El control CheckBox (casilla de verificación) presenta una opción Verdadero/Falso o Sí/No. Es posible activar cualquier número de casillas de verificación en un formulario al mismo tiempo.
	OptionButton	El control OptionButton (botón de opción), se incluye como parte de un grupo de opciones con otros botones de opción, que presentan múltiples opciones, de las cuales el usuario puede elegir sólo una.
	ComboBox	El control ComboBox (cuadro combinado) es una combinación de un cuadro de texto con un cuadro de lista. Permite al usuario escribir su selección o elegir un elemento de una lista desplegable.
	ListBox	El control ListBox (cuadro de lista) presenta una lista de elementos entre los que el usuario puede elegir.
	HScrollBar	
	VScrollBar	La HScrollBar (barra de desplazamiento horizontal) y VScrollBar (barra de desplazamiento vertical) permiten al usuario seleccionar un valor dentro de un rango de valores. (Se usan como controles separados y no son iguales que las barras de desplazamiento que están incorporadas en muchos controles).
	Timer	El Timer (cronómetro) ejecuta eventos cronometrados en intervalos de tiempo especificados.
	DriveListBox	El DriveListBox (cuadro de lista de unidades) presenta y permite al usuario seleccionar unidades de disco válidas.
	DirListBox	El DirListBox (cuadro de lista de directorios) presenta y permite al usuario seleccionar directorios y rutas.
	FileListBox	El (FileListBox) cuadro de lista de archivos presenta y permite al usuario seleccionar de una lista de archivos.
	Shape	El control Shape (forma) agrega un rectángulo, cuadrado, elipse o círculo a un formulario.
	Line	El control Line (línea) agrega un segmento de línea recta a un formulario.
	Image	El control Image (imagen) presenta mapas de bits, iconos o metarchivos de Windows. Cuando se hace clic en él, actúa como un botón de comando.

	Data	El control Data (datos) le permite conectarse a una base de datos existente y presentar información de la misma en sus formularios.
	OLE	El control contenedor OLE incrusta datos en una aplicación de Visual Basic.
	DBCombo	El control DBCombo (cuadro combinado enlazado a datos) proporciona la mayoría de las funciones del control cuadro combinado estándar, más capacidades ampliadas de acceso a datos.
	DBList	El control DBList (cuadro de lista enlazado a datos) proporciona la mayoría de las funciones del control cuadro de lista estándar, más capacidades ampliadas de acceso a datos.
	Menu	El control Menu (menú) crea menús en las aplicaciones de Visual Basic. Se trabaja con controles Menu a través del Editor de menús, al que es posible acceder seleccionando Editor de menús en el menú <b>Herramientas</b> o haciendo clic en el botón Editor de menús de la barra de herramientas.

También es posible utilizar la Ayuda en pantalla (en línea) para los métodos, propiedades y eventos soportados por cada control.

## **CATEGORÍA DE CONTROLES.**

En Visual Basic hay tres grandes categorías de controles:

*Controles estándar*, como los controles botón de comando y marco. Estos controles están contenidos en el archivo .EXE de Visual Basic. Los controles estándar siempre se incluyen en la Caja de herramientas, a diferencia de los controles personalizados y los objetos insertables, que se incorporan y quitan de la Caja de herramientas según sea necesario .

*Controles personalizados*, que existen como archivos separados con las extensiones .VBX u .OCX. Incluyen controles especializados, como el cuadro de diálogo común y controles personalizados de terceras partes, así como versiones ampliadas de controles estándar, tales como los controles cuadro de lista enlazado a datos y el cuadro combinado enlazado a datos.

Los controles personalizados con extensión .OCX en el nombre de archivo se benefician de la nueva tecnología OLE. Los controles personalizados con extensión .VBX en el nombre de archivo usan una tecnología anterior y se encuentran en aplicaciones escritas con versiones anteriores de Visual Basic.

**Nota** Es posible seguir usando controles .VBX con la versión de 16 bits de Visual Basic, pero no será posible utilizar estos controles con la versión de 32 bits de Visual Basic.

Es posible usar controles .OCX con las versiones de 16 bits y de 32 bits de Visual Basic, sin embargo, tendrá que usar controles .OCX de 16 bits con la versión de 16 bits de Visual Basic y controles .OCX de 32 bits con la versión de 32 bits de Visual Basic.

Cuando Visual Basic abre un proyecto que contiene un control .VBX, el comportamiento predeterminado es reemplazar el control .VBX por un control .OCX, pero sólo si hay una versión .OCX del control disponible.

*Objetos insertables*, tales como un objeto hoja de cálculo de Microsoft Excel que contenga la lista de los empleados de su empresa, o un objeto calendario de Microsoft Project que contenga información de planificación de los plazos de un proyecto. Como estos objetos se pueden añadir a la Caja de herramientas, se consideran como controles personalizados. Algunos de estos objetos aceptan automatización OLE, lo que permite programar objetos de otras aplicaciones desde una aplicación de Visual Basic.

### **Para ver la lista de los controles personalizados incluidos en su proyecto**

En el menú **Herramientas**, elija **Controles personalizados**. Las casillas activadas en el cuadro de “Controles disponibles” indican los controles .VBX, los controles .OCX y los objetos insertables incluidos en su proyecto.

### **Para agregar un control personalizado**

1. En el menú **Herramientas**, elija **Controles personalizados**.
2. Seleccione la casilla de verificación que esta junto al nombre del control .VBX, control .OCX u objeto insertable y elija “Aceptar”. Una vez que un control personalizado se encuentra en la Caja de herramientas, es posible agregarlo a un formulario de la misma manera que un control estándar.

### **Para quitar un control personalizado**

1. Quite todas las instancias del control de los formularios de su proyecto. Borre todas las referencias al control en el código del proyecto. Si deja referencias a un control eliminado dentro del código, cuando compile la aplicación se presentará un mensaje de error.
2. En el menú **Herramientas**, elija **Controles personalizados**.
3. Desactive la casilla de verificación que se encuentra junto al nombre del control .VBX, control .OCX u objeto insertable y elija “Aceptar”. Si en su proyecto quedan instancias del control se presentará un mensaje de error.

## CONVENCIONES PARA NOMBRAR OBJETOS.

Cuando se crea un *objeto* (formulario o control), Visual Basic le asigna un valor predeterminado a su propiedad Name. Por ejemplo, a todos los botones de comando se le establece su propiedad Name inicialmente en Comando $n$ , donde  $n$  es 1, 2, 3 y así sucesivamente. Visual Basic le asigna el nombre Comando1 al primer botón de comando que se dibuja en un formulario, Comando2 al segundo Comando3 al tercero, etc.

Esos nombres predeterminados pueden mantenerse. Sin embargo, cuando existen varios controles del mismo tipo, tiene más sentido asignarle a la propiedad Name un contenido que sea más descriptivo. El uso de una convención de nombres puede ser útil, porque tal vez resulte difícil distinguir el botón Comando1 de MiFormulario del botón Comando1 de TuFormulario. Esto es especialmente válido cuando una aplicación se compone de varios módulos de formularios, estándar y de clase.

Por ejemplo, es posible utilizar un prefijo para describir la clase, seguido de un nombre descriptivo del control. El uso de esta convención de nombres hace que el código se comprenda más por sí solo y se puedan agrupar objetos similares de forma alfabética en el cuadro de lista "Objetos".

En esta guía se usan las siguientes convenciones de nombre para objetos de Visual Basic.

**Tabla Convenciones de nombres de objetos en Visual Basic**

Objeto	Prefijo	Ejemplo
Form	frm	frmAbrirArc
Check box	chk	chkSóloLectura
Combo box	cbo	cboInglés
Data-bound combo box	dbc	dbcInglés
Command button	cmd	cmdCancelar
Data	dat	datBiblio
Directory list box	dir	dirOrigen
Drive list box	drv	drvDestino
File list box	fil	filOrigen
Frame	fra	fraldioma
Grid	grd	grdPrecios
Data-bound grid	dbg	dbgPrecios
Horizontal scroll bar	hsb	hsbVolumen
Image	img	imgIcono
Label	lbl	lblMensajeAyuda
Line	lin	linVertical
List box	lst	lstCódigos
Data-bound list box	dbl	dblCódigos

Menu	mnu	mnuAbrirArchivo
OLE container	ole	oleObjeto1
Option button	opt	optFrancés
Picture box	pic	picEspacioDisco
Shape (circle, square, oval, rectangle, rounded rectangle, y rounded square)	shp	shpCírculo
Text box	txt	txtTextEnt
Timer	tmr	tmrAlarma
Vertical scroll bar	vsb	vsbTasa

Los nombres que se asignen a formularios y controles:

- Deben empezar con una letra.
- Sólo pueden contener letras, números y subrayados (\_); no se permiten caracteres de puntuación ni espacios.
- No pueden tener más de 40 caracteres.

### **USO DEL VALOR DE UN CONTROL.**

Todos los controles tienen una propiedad que permite almacenar o recuperar valores refiriéndose solamente al control, sin usar el nombre de la propiedad. Esto se conoce como el *valor* del control y suele ser la propiedad más importante o la más habitualmente usada de ese tipo de control. La tabla indica la propiedad que se considera el valor para cada tipo de control.

**Tabla Controles y las propiedades que son sus valores**

<b>Control</b>	<b>Propiedad</b>
Check box	Value
Combo box	Text
Command button	Value
Common dialog	Action
Data	Caption
Data-bound combo box	Text
Data-bound grid	Text
Data-bound list box	Text
Directory list box	Path
Drive list box	Drive
File list box	FileName
Frame	Caption
Grid	Text
Horizontal scroll bar	Value
Image	Picture
Label	Caption

Line	Visible
List box	Text
Menu	Enabled
Option button	Value
Picture box	Picture
Shape	Shape
Text box	Text
Timer	Enabled
Vertical scroll bar	Value

Siempre que sea necesario hacer referencia a una propiedad de un control, que además sea el valor de ese control, puede hacerse sin especificar el nombre de la propiedad en el código. Por ejemplo, esta línea establece la propiedad Text de un control TextBox:

```
Texto1 = "Este texto es asignado a la propiedad Text de Texto1"
```

En el siguiente ejemplo, a la propiedad Caption de Etiqueta1 se le asigna el valor de la propiedad FileName de Archivo1 cuando el usuario haga clic en el cuadro de lista de archivos:

```
Sub Archivo1_Click ()
    Etiqueta1 = Archivo1
End Sub
```

**Nota** Como el uso del valor de un control hace que el código sea menos comprensible al leerlo, los ejemplos de esta guía no lo usan y en su lugar se refieren explícitamente a las propiedades de todos los controles. Puede intentar escribir su código de las dos maneras y decidir usar el valor de los controles en su código, si esa forma no le provoca dificultades para leerlo.

### ***HACIENDO CLIC EN LOS BOTONES PARA EJECUTAR ACCIONES.***

La manera más fácil de permitir que un usuario interactúe con una aplicación es proporcionarle un botón en el que hacer clic. Es posible usar el control botón de comando proporcionado por Visual Basic o crear su propio "botón" usando un control imagen que contenga un gráfico, como puede ser un icono.

## **USO DE LOS BOTONES DE COMANDO.**

La mayoría de las aplicaciones de Visual Basic tienen botones de comando para permitir que los usuarios sólo tengan que hacer clic en ellos para ejecutar una acción. Cuando el usuario elige el botón, no sólo lleva a cabo la acción apropiada, sino que también parece que ha sido presionado y soltado. Cada vez que el usuario hace clic en un botón se invoca el procedimiento de evento Click. Al colocar código en el procedimiento de evento Click, se ejecuta cualquier acción que se desee.

Hay muchas maneras de seleccionar un botón de comando durante la ejecución:

- Use el *mouse* para hacer clic en el botón.
- Mueva el enfoque al botón usando la tecla TAB y entonces seleccione el botón presionando la BARRA ESPACIADORA o ENTRAR.
- Presione la tecla de acceso rápido (ALT+ la tecla definida) del botón de comando.
- En el código, asigne el valor **True** (Verdadero) a la propiedad Value del botón de comando.
- Si el botón de comando es el *botón de comando* predeterminado del formulario, presionando ENTRAR se selecciona dicho botón, incluso si cambia el enfoque a un control diferente. En tiempo de diseño, se especifica un botón de comando predeterminado estableciendo la propiedad Default de ese control como **True**.
- Si el botón de comando es el *botón Cancelar* predeterminado del formulario, presionando ESC se selecciona dicho botón, incluso si cambia el enfoque a un control diferente. En tiempo de diseño, se especifica un botón Cancelar predeterminado estableciendo la propiedad Cancel de ese control como **True**.

Todas estas acciones hacen que Visual Basic invoque el procedimiento de evento Click.

### **Aplicación Botones de prueba**

La propiedad Caption se usa para presentar en el botón el texto que explica al usuario lo que hace el botón. En la figura , el ejemplo Botones de prueba de la aplicación ejemplo Controls contiene un botón de comando con su propiedad Caption definida como "Change Signal". (Una versión de este ejemplo se puede ver en BUTTON.FRM, en la aplicación ejemplo CONTROLS.VBP).

Fíjese que la tecla de acceso de este botón es 'S'. Si inserta el signo & en el texto de la propiedad Caption, el carácter que le siga a este signo será la tecla de acceso del botón.





### **Figura Botón de comando con título**

Cuando un usuario hace clic en el botón de comando, se ejecuta el código del procedimiento de evento Click de dicho botón de comando. En el ejemplo, se presenta un icono diferente cada vez que se hace clic en el botón.

## ***UN CONTROL IMAGE PARA INTERACTUAR CON EL USUARIO.***

Un control Image también reconoce el evento Click, por lo que puede utilizarse dondequiera que se haya usado un botón de comando. Agrupar varios controles Image horizontalmente en la parte alta de la pantalla, normalmente dentro de un cuadro, le permite crear una barra de herramientas en su aplicación.

Por ejemplo, el ejemplo Botones de prueba también le muestra un control Image que los usuarios pueden seleccionar de la misma forma que seleccionan un botón de comando. Cuando se presenta el formulario por primera vez, el control presenta uno de los iconos de tráfico de la biblioteca de iconos incluida con Visual Basic. Cada vez que se hace clic en el control se presenta un nuevo icono.

La figura muestra el control Image con uno de los iconos de tráfico (TRFFC10A.ICO).



### **Figura Control Image con un icono de tráfico**

Para crear un borde alrededor del control imagen, defina la propiedad BorderStyle como 1-Fixed Single.

**Nota** A diferencia de los botones de comando, los controles Imagen no dan el efecto de ser presionados cuando se les hace clic. Esto quiere decir que si no cambia el mapa de bits en el evento MouseDown, no hay indicación visual al usuario de que el “botón” está siendo presionado.

## ***PRESENTACIÓN E INTRODUCCIÓN DE TEXTO.***

Cuando quiera que su aplicación muestre texto en un formulario, use etiquetas y cuadros de texto. Las etiquetas contienen texto que sólo se puede leer, mientras que los cuadros de texto contienen texto que puede ser editado.

## USO DE ETIQUETAS PARA PRESENTAR TEXTO.

Un control etiqueta presenta texto que el usuario no puede modificar directamente. También es posible usar etiquetas para identificar controles, como cuadros de texto y barras de desplazamiento, que no disponen de la propiedad Caption.

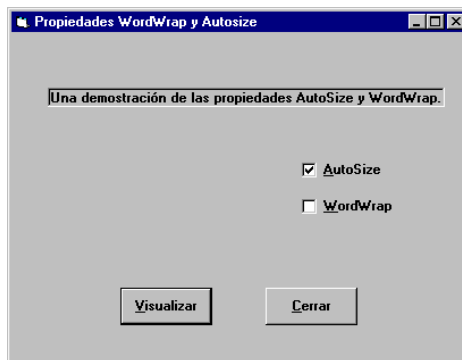
Está predeterminado que el valor de Caption sea la única parte visible del control etiqueta. Sin embargo, si se define la propiedad BorderStyle como 1 (lo que es posible durante el diseño) la etiqueta aparece con un borde, dándole una apariencia similar a la de un cuadro de texto.

**Nota** Se trabaja con formularios y controles, se definen las propiedades y se escribe el código para los eventos en *tiempo de diseño*, el cual es cualquier momento en el que se esté desarrollando una aplicación en el entorno Visual Basic. El *tiempo de ejecución* es el que corresponde a los momentos en que se está ejecutando la aplicación e interactuando con ella.

### Modificar el tamaño de una etiqueta para ajustar su contenido

Las etiquetas de una sola línea pueden definirse en tiempo de diseño en la ventana Propiedades. Pero ¿qué pasa cuando se quiere introducir un título más largo o un título que puede cambiar en el tiempo de ejecución?. Las etiquetas tienen dos propiedades que le ayudan a modificar los controles para ajustarse a títulos más largos o más cortos: AutoSize y WordWrap.

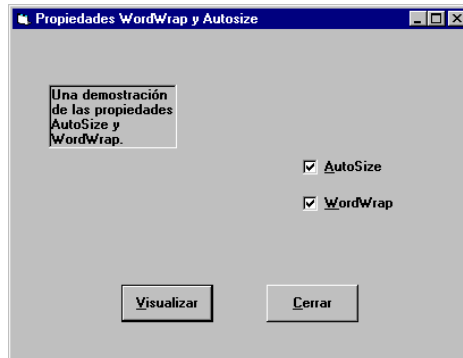
La propiedad AutoSize determina si un control debería modificar automáticamente su tamaño para ajustarse a su contenido. Si se establece como **True**, la etiqueta crece horizontalmente para ajustarse a su contenido, como se muestra en la figura.



**Figura** Ejemplo de AutoSize

La propiedad WordWrap provoca que el control crezca verticalmente para ajustarse a su contenido, manteniendo su anchura, como se muestra en la

figura Para ver un ejemplo de este caso, vea WORDWRAP.FRM en la aplicación ejemplo CONTROLS.VBP.



**Figura Ejemplo de Ajuste de líneas**

**Nota** Si ejecuta el ejemplo AutoSize de CONTROLS.VBP, podrá ver que para que el ejemplo WordWrap funcione, las dos casillas de verificación del formulario deben estar seleccionadas antes de hacer clic en el botón Display. Para que la propiedad WordWrap de la etiqueta tenga efecto, AutoSize debe estar establecida como **True**. La anchura de la etiqueta sólo se incrementa si el tamaño de una palabra excede la anchura actual del control.

### **Uso de etiquetas para crear teclas de acceso**

Es posible crear una tecla de acceso para cualquier control que tenga una propiedad Caption anteponiendo un signo & a la letra que decida usar como tecla de acceso. Para asignar teclas de acceso a controles que no tengan la propiedad Caption, use una etiqueta con el control. Como las etiquetas no pueden recibir el enfoque, éste pasa al control siguiente dentro del orden de tabulación. Use esta técnica para asignar teclas de acceso a cuadros de texto, cuadros de imagen, cuadros combinados, cuadros de lista, cuadros de lista de unidades, cuadros de lista de directorios, cuadrículas e imágenes.

### **Para asignar una tecla de acceso a un control con una etiqueta**

1. Dibuje primero la etiqueta y después el control.

– O b i e n –

Dibuje los controles en cualquier orden y establezca la propiedad TabIndex de la etiqueta con el número inmediatamente inferior al del control.

2. Use el signo & en la propiedad Caption de la etiqueta para definir la tecla de acceso de la etiqueta.

**Nota** Es posible que le interese incluir signos & en un control etiqueta en vez de usarlos para crear teclas de acceso. Esto podría ocurrir cuando quiera vincular un control etiqueta con datos de un conjunto de registros que incluya signos &. Para presentar signos & en un control etiqueta establezca la propiedad UseMnemonic como **False**.

## **TRABAJAR CON CUADROS DE TEXTO.**

Los cuadros de texto son controles versátiles que se pueden usar para obtener entradas del usuario o presentar texto. Los cuadros de texto no se deben utilizar para presentar texto que no se desee que pueda ser modificado por el usuario, a menos que haya establecido la propiedad Locked como **True**, para impedir que los usuarios modifiquen el texto.

### **Cuadros de texto de múltiples líneas y saltos de línea**

La apariencia y el comportamiento de un cuadro de texto están influenciadas por dos propiedades, MultiLine y ScrollBars, que se definen sólo en tiempo de diseño.

**Nota** La propiedad ScrollBars no debe confundirse con los controles barra de desplazamiento, que no están incorporados a los cuadros de texto y tienen su propio juego de propiedades.

Establecer MultiLine como **True** habilita a un cuadro de texto para que acepte múltiples líneas de texto en tiempo de ejecución. Un cuadro de texto de múltiples líneas administra automáticamente los saltos de línea, mientras no exista una barra de desplazamiento horizontal. La propiedad ScrollBars presenta el valor predeterminado 0-None. El salto de línea automático ahorra al usuario el trabajo de insertar saltos de línea al final de las líneas. Cuando una línea de texto es más larga de lo que puede presentar una línea del cuadro, el texto continúa en la línea siguiente.

Los saltos de línea no pueden introducirse en la ventana Propiedades en tiempo de diseño. Dentro de un procedimiento, es posible crear un salto de línea insertando el carácter salto de línea (NL), un retorno de carro seguido de una línea nueva (caracteres ANSI 13 y 10). Por ejemplo, el siguiente procedimiento de evento coloca dos líneas de texto en un cuadro de texto de múltiples líneas (Texto1) cuando se abre el formulario:

```
Sub Form_Load ()
```

```
    NL = Chr(13) + Chr(10)           ' Definir carácter nueva línea.
```

```
    Texto1.Text = "Aquí hay dos líneas " & NL & "en un cuadro de texto"
```

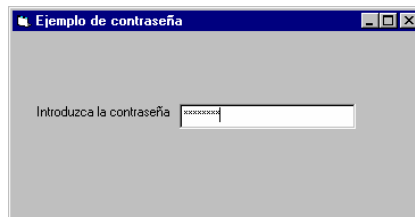
```
End Sub
```

### **Creación de un cuadro de texto de contraseña**

Un cuadro de contraseña es un cuadro de texto que permite al usuario escribir en el mismo su contraseña mientras se muestran sólo caracteres marcadores de posición, como pueden ser asteriscos. Visual Basic dispone de dos

propiedades de cuadro de texto, PasswordChar y MaxLength, que facilitan la creación un cuadro de texto de contraseña.

PasswordChar especifica el carácter que se presentará en el cuadro de texto. Por ejemplo, si quiere presentar asteriscos en el cuadro de contraseña, establezca la propiedad PasswordChar como \* en la ventana Propiedades. De esa forma, se presentará un asterisco independientemente del carácter que el usuario escriba, como se muestra en la figura .



**Figura Ejemplo de contraseña**

Con MaxLength es posible determinar cuántos caracteres pueden escribirse en el cuadro de texto. Cuando se excede MaxLength, se emite un sonido y el cuadro de texto no aceptará más caracteres.

### **Cancelación de pulsaciones en un cuadro de texto**

Es posible usar el evento KeyPress para restringir o transformar caracteres al ser escritos. El evento KeyPress usa un argumento, *keyascii*. Este argumento es un entero que representa el equivalente numérico (ASCII) del carácter introducido en el cuadro de texto.

El siguiente ejemplo muestra cómo cancelar pulsaciones de teclas al ser introducidas. Si el carácter escrito no se encuentra dentro de un rango especificado, el procedimiento lo cancela estableciendo KeyAscii como 0. El cuadro de texto de este ejemplo se llama txtEnterNums y el procedimiento impide que el cuadro de texto reciba caracteres que no sean números. Compara KeyAscii directamente con el valor numérico (**Asc**) de varios caracteres.

```
Private Sub txtEnterNums_KeyPress (KeyAscii As Integer)
```

```
    If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then  
        KeyAscii = 0 ' Cancela el  
carácter.  
        Beep ' Emitir sonido  
de error.  
    End If  
End Sub
```

### **Creación de un cuadro de texto de sólo lectura**

Es posible usar la propiedad Locked para impedir que los usuarios editen el contenido de un cuadro de texto. Establezca la propiedad Locked como **True** para permitir a los usuarios desplazarse y resaltar texto sin que puedan realizar modificaciones. Con la propiedad Locked establecida como **True** el comando **Copiar** funcionará en el cuadro de texto, pero los comandos **Cortar** y **Pegar** no. La propiedad Locked sólo afecta a la *interacción del usuario* en tiempo de ejecución. Todavía puede cambiar el contenido del cuadro de texto *por programa* en tiempo de ejecución, modificando la propiedad Text del cuadro de texto.

### Imprimir comillas en una cadena

En una cadena de texto algunas veces aparecen comillas (" ").

Ella dijo, "Te mereces un agasajo!"

Como las cadenas asignadas a una variable o propiedad están encerradas entre comillas (" "), es necesario insertar un juego adicional de comillas por cada juego a presentar en una cadena. Visual Basic interpreta dos comillas seguidas como una comilla incrustada.

Por ejemplo, para crear la cadena anterior, use el siguiente código:

```
Texto1.Text = "Ella dijo, ""Te mereces un agasajo!"" "
```

Para conseguir el mismo efecto, es posible usar el código ASCII de las comillas (34):

```
Texto1.Text = "Ella dijo, " & Chr(34) + "Te mereces un agasajo!" & Chr(34)
```

### **CONTROLES QUE PRESENTAN OPCIONES AL USUARIO.**

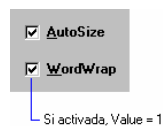
En la mayoría de las aplicaciones, querrá proporcionar opciones a los usuarios. Estas pueden presentarse como botones de opción o casillas de verificación que pueden activarse, como listas en las que se eligen elementos o como barras de desplazamiento que permiten seleccionar un valor de una escala. La siguiente tabla resume estos controles.

<b>Para proporcionar esta función</b>	<b>Use este control</b>
Un conjunto de opciones de las cuales el usuario puede elegir una o más de una.	Check boxes (casillas de verificación)
Un conjunto de opciones de las cuales el usuario sólo puede elegir una.	Option buttons (use frames (marcos) si se necesitan grupos adicionales)
Una lista desplegable de opciones entre las que el usuario puede elegir.	List box (cuadro de lista)

Un cuadro de lista que contenga un campo para editar texto. Puede elegir de la lista o escribir la opción deseada en el campo de edición.	Combo box (cuadro combinado)
Un rango de opciones correspondientes a una escala numérica.	Barra de desplazamiento horizontal o barra de desplazamiento vertical

## SELECCIÓN DE OPCIONES INDIVIDUALES CON CASILLAS DE VERIFICACIÓN.

Una casilla de verificación indica si una condición particular está activada o desactivada. En una aplicación, haga uso de casillas de verificación para proporcionarle una opción Verdadero/Falso o Sí/No al usuario. Como cada casilla de verificación funciona independientemente de las demás, un usuario puede seleccionar cualquier número de casillas de verificación al mismo tiempo. Así, en la figura, se pueden seleccionar AutoSize y WordWrap.

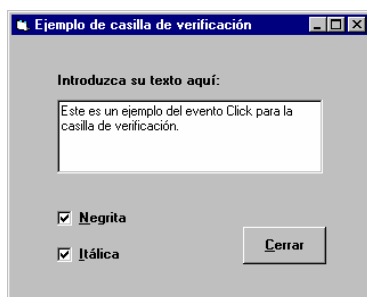


**Figura Casillas de verificación**

### La aplicación Check Box

El ejemplo Check box usa una casilla de verificación para determinar si el texto se ha de presentar con una fuente normal o en cursiva. Un ejemplo de un caso así puede verse en CHECK.FRM, en la aplicación ejemplo CONTROLS.VBP.

La aplicación tiene un cuadro de texto, una etiqueta, un botón de comando y dos casillas de verificación, como se muestra en la figura.



**Figura Ejemplo de casilla de verificación**

La siguiente tabla relaciona las propiedades de los objetos de la aplicación.

Objeto	Propiedad	Valor
Form	Name	
Caption	frmCheck	

Ejemplo Check Box		
Cuadro de texto	Name	
Text	txtDisplay	
(Vacío)		
Primera casilla de verificación	Name	
Caption	chkBold	
&Negrita		
Segunda casilla de verificación	Name	
Caption	chkItalic	
&Cursiva		
Botón de comando	Name	
Caption	cmdClose	
&Cerrar		

Cuando se activa Negrita o Cursiva, el valor de la propiedad Value de la casilla de verificación es 1, cuando se desactiva, la propiedad Value es 0.

### Eventos de la aplicación Check Box

El evento Click de la casilla de verificación ocurre en el momento en que se hace clic en la casilla. El evento Click de la casilla de verificación comprueba si ésta ha sido seleccionada (esto es, si Value = 1 o vbChecked). Si es así, el texto se convierte en negrita o cursiva según se establezcan las propiedades Bold o Italic del objeto Font.

```
Private Sub chkNegrita_Click ()
    Dim objF As Font
    Set objF = txtDisplay.Font If chkBold.Value = vbChecked Then
        ' Si está seleccionada
        objF.Bold = True
    Else
        ' Si no está
        seleccionada objF.Bold = False
    End If
End Sub
```

```
Private Sub chkCursiva_Click ()
    Dim objF As Font
    Set objF = txtDisplay.Font
    If chkItalic.Value = vbChecked Then ' Si está seleccionada
        objF.Italic = True
    Else
        ' Si no está
        seleccionada objF.Italic = False
    End If
End Sub
```

```
Private Sub cmdSalir_Click()
```



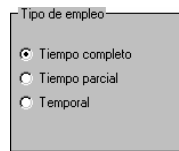
Unload Me

End Sub

## **AGRUPAR OPCIONES CON OPTION BUTTONS.**

Los botones de opción (Option buttons) presentan un conjunto de opciones al usuario. Sin embargo, a diferencia de las casillas de verificación, los botones de opción deben funcionar siempre como parte de un grupo; la selección de un botón de opción desactiva inmediatamente las de los demás botones de opción del grupo. Cuando se establece un grupo de botones de opción se le está diciendo al usuario, "Aquí hay un conjunto de opciones de las que sólo puede elegir una".

Por ejemplo, en el grupo de botones de opción mostrado en la figura, el usuario puede seleccionar uno de los tres botones de opción.

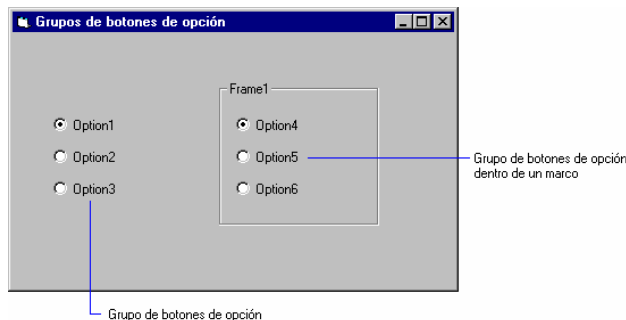


**Figura Selección de un botón de opción**

### **Creación de grupos de botones de opción**

Todos los botones de opción situados directamente en un formulario, no dentro de un marco ni dentro de un cuadro de imagen, constituyen un grupo. Si quiere crear grupos de botones de opción adicionales, tiene que colocar algunos de ellos dentro de marcos o cuadros de imagen.

Todos los botones de opción situados dentro de un marco dado constituyen un grupo separado, así como todos los botones de opción dentro de un cuadro de imagen. Cuando cree un grupo separado de esta manera, dibuje siempre primero el marco o el cuadro de imagen y después dibuje los botones de opción dentro de él. La figura muestra dos grupos de botones de opción.



**Figura Grupos de botones de opción**

Un marco proporciona una agrupación visual y funcional de controles. Los marcos se usan frecuentemente para agrupar botones de opción o casillas de verificación. Si dibuja botones de opción dentro de un marco, el usuario sólo podrá seleccionar un botón de opción del grupo.

### **Para agrupar controles en un marco**

1. Seleccione la herramienta **Frame** de la Caja de herramientas y dibuje el marco.
2. Seleccione cada control en la Caja de herramientas, después dibuje el control dentro del marco.

Si dibuja primero el marco y después los controles que van dentro de él, entonces será posible mover juntos el cuadro y los controles. Si intenta mover controles existentes dentro de un marco, los controles no se moverán junto con el marco.

**Nota** Si tiene controles ya existentes que quiere agrupar en un marco, puede seleccionar todos los controles y luego cortarlos y pegarlos dentro de un control marco o cuadro de imagen.

### **Contenedores de controles**

Mientras que los controles son objetos independientes, existe cierta *relación de interdependencia* entre formularios y controles. La figura demuestra cómo los botones de opción pueden estar contenidos dentro de un formulario o de un control marco.

Para comprender el concepto de contenedor, necesita comprender que todos los controles son formularios secundarios del formulario donde han sido dibujados. De hecho, la mayoría de los controles soportan la propiedad Parent de sólo lectura, que devuelve el formulario donde está ubicado un control. El ser un formulario secundario afecta la colocación del control en el formulario primario. Las propiedades Left y Top de un control son relativas al formulario primario y los controles no pueden moverse fuera de los límites de dicho formulario. Cuando se mueve el formulario, se mueve también a los controles.

Cuando un control está incorporado en un control marco o en un control PictureBox, las propiedades Left y Top del control son relativas a su contenedor. Esto significa que cuando se mueve el contenedor, la posición del control relativa a las propiedades Left y Top del contenedor no cambia, porque el control se mueve junto con el contenedor.

### **Seleccionar o deshabilitar botones de opción**

Cuando el usuario selecciona un botón de opción, el botón aparece sombreado. Un botón de opción puede seleccionarse:

Haciendo clic en él con el *mouse* en tiempo de ejecución.

Situándose con el tabulador en el grupo de botones de opción y después usando las teclas de flecha para seleccionar un botón de opción dentro del grupo.

Asignando a su propiedad Value el valor **True** con el código:

```
Opción1.Value = True
```

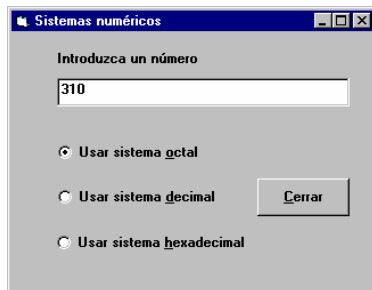
Usando una tecla de método abreviado especificada en el título de una etiqueta.

Para seleccionar un botón predeterminado de un grupo de botones de opción, establezca su propiedad Value como **True** en tiempo de diseño. Dicho botón permanece seleccionado hasta que el usuario seleccione un botón de opción diferente o hasta que cambie el código.

Para deshabilitar un botón de opción, establezca su propiedad Enabled como **False**. Cuando el programa se ejecute aparecerá atenuado, lo que indica que no está disponible.

### Aplicación Sistema de numeración

El formulario mostrado en la figura usa botones de opción para determinar cual de los tres sistemas de numeración (octal, decimal, o hexadecimal) se utilizará. Cuando el usuario selecciona un botón, el número presentado en el cuadro de texto es convertido al nuevo sistema de numeración. Un ejemplo de un caso así puede verse en NUMBER.FRM, en la aplicación ejemplo CONTROLS.VBP.



**Figura Ejemplo de botón de opción**

Cree un formulario con una etiqueta, un cuadro de texto, un botón de comando y tres botones de opción. Las propiedades se definen como se muestra en la siguiente tabla.

Objeto	Propiedad	Valor
Etiqueta	Name	
Caption	lblNúmero	
Introduzca un número:		
Cuadro de texto	Name	
Text	txtNúmero	

(Vacío)		
Botón de comando	Name	
Caption	cmdCerrar	
&Cerrar		
Primer botón de opción	Name	
Caption		
Value	optOctBotón	
Usar sistema &octal		
<b>True</b>		
Segundo botón de opción	Name	
Caption	optDecBotón	
Usar sistema &decimal		
Tercer botón de opción	Name	
Caption	optHexBotón	
Usar sistema &hexadecimal		

### Eventos de la aplicación Sistema de numeración

La aplicación Sistema de numeración responde a los siguientes eventos:

El evento Change del cuadro de texto introduce el valor (usando el sistema de numeración apropiado) y lo almacena en una variable numérica de formulario, CurrentNum.

El evento Click de optOctBotón devuelve CurrentNum en octal.

El evento Click de optDecBotón devuelve CurrentNum en decimal.

El evento Click de optHexBotón devuelve CurrentNum en hexadecimal.

La clave de esta técnica radica en el uso de una variable de formulario, CurrentNum. Esta variable representa el valor del cuadro de texto (txtNúmero) en forma numérica. El evento Change mantiene este valor actualizado, de manera que los procedimientos de evento Click sólo tienen que devolver el número en el sistema de numeración adecuado. CurrentNum tiene que estar declarada en la Sección de declaraciones del código de formularios como:

```
Dim CurrentNum As Variant
```

El valor predeterminado de la variable es 0. Este es un valor aceptable, así que no hay necesidad de código de inicialización.

El procedimiento de evento Change determina qué sistema de numeración (octal, decimal, o hexadecimal) está en vigor y después introduce el número. Este procedimiento examina la condición *BotónOpción.Value* para determinar el sistema de numeración que se ha de usar:

```
Private Sub txtNúmero_Change ()
```

```
If optOctBotón.Value = True Then
    CurrentNum = Val("&O" & LTrim(txtNúmero.Text) & "&")
Elseif optDecBotón.Value = True Then
    CurrentNum = Val(LTrim(txtNúmero.Text) + "&")
Else
    CurrentNum = Val("&H" & LTrim(txtNúmero.Text) & "&")
End If
End Sub
```

La función **Val** se usa para convertir la cadena a número y puede reconocer cadenas octales, decimales y hexadecimales. La función **LTrim** elimina del texto los espacios a la izquierda. El prefijo “&O” hace que una cadena de dígitos se interprete como octal y el prefijo “&H” hace que se interprete como hexadecimal.

Los procedimientos de evento Click de los botones de opción usan las funciones **Oct** o **Hex** para presentar CurrentNum en el sistema de numeración apropiado:

```
Private Sub optOctBotón_Click ()
    txtNúmero.Text = Oct(CurrentNum)
End Sub
```

```
Private Sub optDecBotón_Click ()
    txtNúmero.Text = Format(CurrentNum)
End Sub
```

```
Private Sub optHexBotón_Click ()
    txtNúmero.Text = Hex(CurrentNum)
End Sub
```

## Para más información

Busque Ayuda sobre *Val*, *LTrim*, *Oct*, o *Hex*.

## USO DE CUADRO DE LISTA Y CUADROS COMBINADOS.

Los cuadros de lista y cuadros combinados presentan una lista de opciones al usuario. De forma predeterminada, las opciones se presentan verticalmente en una columna única, aunque también pueden establecerse múltiples columnas. Si el número de elementos excede al que se puede presentar en el cuadro de lista o en el cuadro combinado, en el control aparecerán automáticamente barras de desplazamiento. El usuario podrá entonces desplazarse de arriba a abajo o de izquierda a derecha por la lista. La figura muestra un cuadro de lista de una columna.



**Figura Cuadro de lista de una columna**

Un control cuadro combinado (Combo Box) une las funciones de un cuadro de texto y las de un cuadro de lista. Este control permite que el usuario seleccione entre escribir texto en el cuadro combinado o seleccionar un elemento de la lista.

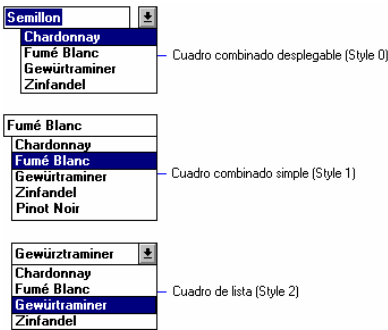
Visual Basic incluye las versiones estándar y la enlazada a datos de los cuadros de lista y de los cuadros combinados. Mientras que ambas versiones de los controles cuadro de lista y cuadro combinado le permiten presentar, editar y actualizar información de los tipos de bases de datos más habituales, los cuadros de lista y combinado enlazados a datos proporcionan funciones más avanzadas de acceso a datos. Estos controles soportan también un juego de propiedades y métodos diferente que los controles estándar cuadro de lista y cuadro combinado.

### Estilos de cuadros combinados

Hay tres tipos de cuadros combinados:

- Cuadro combinado desplegable (Style 0)
- Cuadro combinado simple (Style 1)
- Cuadro de lista desplegable (Style 2)

La figura muestra los tres estilos de cuadros combinados.



**Figura Estilos de cuadros combinados**

### **Cuadro combinado desplegable (Style 0)**

Un cuadro combinado con el valor predeterminado (Style = 0), es un cuadro combinado desplegable. El usuario puede introducir texto directamente (como en un cuadro de texto) o hacer clic en la flecha que se presenta en la parte derecha del cuadro, para abrir una lista de opciones. Al seleccionar una de las opciones, ésta se inserta en el cuadro de texto ubicado en la parte superior del cuadro combinado. El usuario también puede abrir la lista presionando ALT+FLECHA ABAJO cuando el control tiene el enfoque. Cuando quiera permitir que el usuario sólo seleccione de una lista y quiera conservar espacio en la pantalla, es posible usar un cuadro de lista desplegable.

### **Cuadro combinado simple (Style 1)**

Cuando se establece la propiedad Style de un cuadro combinado como 1, se especifica un cuadro combinado simple en el que la lista se presenta siempre. Para presentar todos los elementos en la lista, el cuadro de lista debe definirse lo bastante grande para ello. Cuando hay más elementos de los que se pueden presentar, aparece automáticamente una barra de desplazamiento. El usuario puede todavía introducir texto directamente o seleccionar de la lista. Al igual que con el cuadro combinado desplegable, un cuadro combinado simple también permite que los usuarios introduzcan opciones que no están en la lista.

### **Cuadro de lista desplegable (Style 2)**

Una cuadro de lista desplegable es como un cuadro de lista normal, presenta una lista de elementos entre los cuales el usuario debe elegir. A diferencia de los cuadros de lista, la lista no se presenta hasta que se hace clic en la flecha a la derecha del cuadro. La diferencia clave entre este y el cuadro combinado desplegable está en que el usuario no puede escribir texto en el cuadro, sólo puede seleccionar un elemento de la lista. Use este tipo de lista cuando el espacio sea escaso. El cuadro de diálogo **Agregar inspección** de Visual Basic contiene dos cuadros de lista desplegables. Es posible abrir este cuadro de diálogo seleccionando **Agregar inspección** en el menú **Herramientas**.

### **Cuando usar un cuadro combinado en lugar de un cuadro de lista**

Generalmente, un cuadro combinado es apropiado cuando hay una lista de opciones *sugeridas* y un cuadro de lista es apropiado cuando quiere limitar la entrada a lo que contiene la lista. Un cuadro combinado contiene un campo de edición, así que en este campo se pueden escribir opciones que no están en la lista.

Además, los cuadros combinados ahorran espacio en el formulario. Como la lista no se despliega hasta que el usuario hace clic en la flecha (excepto en Style 1, que está siempre visible), un cuadro combinado puede colocarse fácilmente en un espacio pequeño, donde no sería posible ubicar un cuadro de lista.

La siguiente tabla relaciona eventos de cuadro combinado no aceptados.

<b>Propiedad Style</b>	<b>Evento no aceptado</b>
0 (Combinado desplegable)	DbClick
1 (Combinado simple)	DropDown
2 (Lista desplegable)	Change, DbClick

Una práctica recomendada para eventos de cuadro lista (especialmente cuando la lista aparece como parte de un cuadro de diálogo) es agregar un botón de comando para usarlo con el cuadro de lista. El procedimiento de evento Click de ese botón debe hacer uso de la selección del cuadro de lista, efectuando la acción que sea apropiada para la aplicación.

Hacer doble clic en un elemento de la lista debe tener el mismo efecto que seleccionar el elemento y después hacer clic en el botón de comando. Para hacer esto, haga que el procedimiento DbClick del cuadro de lista llame al procedimiento Click del botón de comando:

```
Private Sub List1_DbClick ()  
    Comando1_Click  
End Sub
```

O bien, establezca el valor de la propiedad Value del botón de comando como **True**, lo que invocará automáticamente el procedimiento de evento:

```
Private Sub List1_DbClick ()  
    Comando1.Value = True  
End Sub
```



Esto proporciona un método abreviado a los usuarios del *mouse*, sin impedir que los usuarios de teclado puedan efectuar la misma acción. Tenga en cuenta que no hay un equivalente de teclado para el evento `DbClick`.

### Agregar elementos a una lista

Para agregar elementos a un cuadro de lista estándar o a un cuadro combinado estándar, use el método **AddItem**, que tiene la siguiente sintaxis:

*cuadro*.**AddItem** *elemento*[, *índice*]

Parámetro	Descripción
<i>cuadro</i>	Nombre del cuadro de lista o combinado.
<i>elemento</i>	Expresión de cadena para agregar a la lista. Si <i>elemento</i> es una constante literal, delimítela entre comillas.
<i>índice</i>	Especifica donde se ha de insertar el nuevo elemento en la lista. Un <i>índice</i> de 0 representa la primera posición. Si se omite <i>índice</i> , el elemento se inserta al final (o en el orden apropiado).
Mientras los elementos de la lista se agregan habitualmente en el procedimiento de evento <code>Form_Load</code> , es posible usar el método <b>AddItem</b> en cualquier momento. Esto permite agregar dinámicamente elementos a lista (en respuesta a acciones de los usuarios).	

### Ejemplo

El siguiente código coloca "Alemania", "India", "Francia" y "USA" en un cuadro de lista llamado `List1`:

```
Private Sub Form_Load ()
    List1.AddItem "Alemania"
    List1.AddItem "India"
    List1.AddItem "Francia"
    List1.AddItem "USA"
End Sub
```

Siempre que el formulario es abierto en tiempo de ejecución, la lista aparecerá como se muestra en la figura.

Alemania
India
Francia
Estados Unidos

**Figura Cuadro de lista “Países”**

### **Agregar un elemento en una posición específica**

Para agregar un elemento a una lista en una posición determinada, especifique un valor índice después del nuevo elemento. Por ejemplo, la siguiente línea de código inserta “Japón” en la primera posición, ajustando la posición de los otros elementos hacia abajo:

```
List1.AddItem "Japón", 0
```

Fíjese que es 0, no 1, lo que especifica la primera posición en una lista (vea la figura).

Japón
Alemania
India
Francia
Estados Unidos

**Figura Agregar un elemento a una lista**

### **Eliminar elementos de una lista**

Es posible usar el método **RemoveItem** para eliminar elementos de un cuadro de lista estándar o de un cuadro combinado. **RemoveItem** tiene un argumento, *índice*, que especifica el elemento a eliminar:

*cuadro.RemoveItem índice*

Los parámetros *cuadro* e *índice* son los mismos que para **AddItem**.

Por ejemplo, para eliminar el primer elemento de una lista, se agrega la siguiente línea de código:

```
List1.RemoveItem 0
```

Para eliminar todos los elementos de una lista en las versiones estándar o enlazada a datos de los cuadros de lista y cuadros combinados, use el método **Clear**:

```
List1.Clear
```

### **Ordenación de una lista**

Es posible especificar que los elementos sean agregados a una lista en orden alfabético estableciendo la propiedad **Sorted** como **True** y omitiendo el índice.

La ordenación no distingue entre mayúsculas y minúsculas, así que, las palabras “fede” y “Fede” se tratan igual.

Cuando la propiedad Sorted está establecida como **True**, el uso del método **AddItem** con el argumento *índice* puede provocar resultados con ordenaciones impredecibles.

### **Obtener el contenido de una lista con la propiedad Text**

Normalmente, la manera más fácil de obtener el valor del elemento seleccionado actualmente es usar la propiedad Text:

Con los cuadros de lista, la propiedad Text siempre se corresponde con un elemento que el usuario selecciona en tiempo de ejecución.

Con los cuadros combinados, la propiedad Text se corresponde con lo que se haya introducido en tiempo de ejecución en la parte correspondiente al cuadro de texto del control. Esto puede ser un elemento seleccionado de la lista o una cadena que el usuario escriba en el cuadro de texto.

Por ejemplo, el siguiente código presenta información sobre Canadá, si un usuario selecciona “Canadá” de un cuadro de lista:

```
Private Sub List1_Click ()  
    If List1.Text = "Canadá" Then  
        Text1.Text = "Canadá tiene 24 millones de habitantes."  
    End If  
End Sub
```

La propiedad Text contiene el elemento seleccionado actualmente en el cuadro de lista List1. El código comprueba si se ha seleccionado “Canadá” y, si es verdad, presenta la población en el cuadro de texto.

### **Acceso a los elementos de una lista con la propiedad List**

En los cuadros de lista o combinados estándar, la propiedad List proporciona acceso a todos los elementos de la lista. Esta propiedad contiene una matriz en la que cada elemento de la matriz es un elemento de la lista. Cada elemento se representa en forma de cadena. Para referirse a un elemento de una lista, use la siguiente sintaxis:

*cuadro*.**List**(*índice*)

El argumento *cuadro* es una referencia al cuadro de lista o al cuadro combinado, e *índice* es la posición del elemento. El elemento superior tiene un índice de 0, el siguiente tiene un índice de 1 y así sucesivamente. Por ejemplo,

la siguiente instrucción presenta en un cuadro de texto el tercer elemento (*índice* = 2) de una lista:

```
Text1.Text = List1.List(2)
```

### **Determinar la posición con la propiedad ListIndex**

Si se quiere conocer la posición de un elemento seleccionado de un cuadro de lista o combinado estándar, se utiliza la propiedad ListIndex. Esta propiedad asigna o devuelve el índice del elemento seleccionado actualmente en el control y sólo está disponible en tiempo de ejecución. La asignación de la propiedad ListIndex de un cuadro de lista o combinado también genera un evento Click en el control.

El valor de esta propiedad es 0 si está seleccionado el primer elemento (el más alto), 1 si está seleccionado el elemento siguiente y así sucesivamente. ListIndex es - 1 si no hay ningún elemento seleccionado o si un usuario introduce su opción en un cuadro combinado (Style 0 ó 1) en vez de seleccionar un elemento existente de la lista.

**Nota** En los cuadros de lista y combinados estándar, la propiedad NewIndex le permite conocer el índice del último elemento agregado a la lista. Esto puede ser útil cuando se inserta un elemento en una lista ordenada. Para obtener detalles sobre NewIndex, busque *Ayuda sobre NewIndex*.

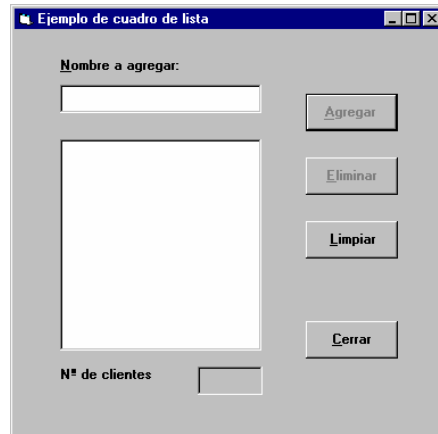
### **Determinar el número de elementos con la propiedad ListCount**

Para determinar el número de elementos de un cuadro de lista o combinado estándar, use la propiedad ListCount. Por ejemplo, la siguiente instrucción usa la propiedad ListCount para determinar el número de elementos de un cuadro de lista:

```
Text1.Text = "Tiene " & List1.ListCount & " elementos en la lista"
```

### **Aplicación List Box**

Este ejemplo le muestra la forma de utilizar los métodos **AddItem**, **RemoveItem** y **Clear** con las propiedades ListIndex y ListCount para agregar y eliminar elementos de una lista en tiempo de ejecución. El ejemplo de la figura permite al usuario escribir el nombre de un cliente en un cuadro de texto y agregarlo a la lista haciendo clic en el botón **Añadir**. Un usuario puede eliminar un elemento de la lista actual, seleccionando el elemento y seleccionando el botón **Borrar** o eligiendo **Limpiar** para eliminar todos los elementos. Un ejemplo de un caso así puede verse en LISTBOX.FRM, en la aplicación ejemplo CONTROLS.VBP.



**Figura Cuadro de lista que usa los métodos AddItem, RemoveItem y Clear**

El número de clientes del cuadro de lista se presenta en una etiqueta que parece un cuadro de texto (BorderStyle se define como 1-Fixed Single). Esta etiqueta se actualiza cada vez que se agrega o se elimina un nombre de cliente. Dado que la propiedad Sorted del cuadro de lista está definida como **True**, los elementos se agregan a la lista manteniendo su orden alfabético.

Cree un formulario con un cuadro de texto, un cuadro de lista, tres etiquetas y cuatro botones de comando. La siguiente tabla relaciona los valores de las propiedades de los objetos de la aplicación.

Objeto	Propiedad	Valor
Cuadro de texto superior	Name	
Text	txtNombre	
(Vacío)		
Etiqueta superior	Name	
Caption	lblNombre	
&Nombre a añadir		
Cuadro de lista	Name	
Sorted	lerCliente	
<b>True</b>		
Etiqueta inferior	Name	
Caption	lblClientes	
Nº de Clientes		
Etiqueta del número de clientes		
(parece un cuadro de texto)	Name	
Caption		
BorderStyle	lblPresenta	
(Vacío)		
1-Fixed Single		
Primer botón de comando	Name	

Caption	cmdAgregar	
&Agregar		
Segundo botón de comando	Name	
Caption	cmdBorrar	
&Borrar		
Tercer botón de comando	Name	
Caption	cmdLimpiar	
L&impiar		
Cuarto botón de comando	Name	
Caption	cmdCerrar	
&Cerrar		

### Eventos en la aplicación Cuadro de lista

Agregue este código al procedimiento de evento cmdAgregar\_Click:

```
Private Sub cmdAgregar_Click ()
    lcrCliente.AddItem txtNombre.Text           ' Agregar a la lista.
    txtNombre.Text = ""                         ' Borra cuadro
    texto.
    txtNombre.SetFocus
    lblPresenta.Caption = lcrCliente.ListCount ' Presenta número.
End Sub
```

Agregue este código al procedimiento de evento cmdBorrar\_Click:

```
Private Sub cmdBorrar_Click ()
    Dim Ind As Integer

    Ind = lcrCliente.ListIndex                 ' Obtener índice.
    ' Asegurarse que hay un elemento seleccionado.
    If Ind >= 0 Then
        ' Borrarlo de la lista.
        lcrCliente.RemoveItem Ind
        ' Presentar número.
    End If
End Sub
```

```
        lblPresenta.Caption = lerCliente.ListCount
    Else
        Beep
    End If
    ' Deshabilita botón si la lista está vacía.
    cmdBorrar.Enabled = (lerCliente.ListIndex <> -1)
End Sub
```

Agregue este código al procedimiento de evento cmdLimpiar\_Click:

```
Private Sub cmdLimpiar_Click ()
    ' Vacía la lista.
    lerCliente.Clear
    ' Deshabilita el botón Borrar.
    cmdBorrar.Enabled = False
    ' Presenta número.
    lblPresenta.Caption = lerCliente.ListCount
End Sub
```

Agregue este código al procedimiento de evento cmdCerrar\_Click:

```
Private Sub cmdCerrar_Click ()
    Unload Me
End Sub
```

Agregue este código al procedimiento de evento lerCliente\_Click:

```
Private Sub lerCliente_Click ()
    cmdBorrar.Enabled = lerCliente.ListIndex <> -1
End Sub
```

Agregue este código al procedimiento de evento txtNombre\_Change:

```
Private Sub txtNombre_Change ()  
  
' Habilita el botón Agregar si hay algún carácter en el nombre.  
  
cmdAgregar.Enabled = (Len(txtNombre.Text) > 0)  
  
End Sub
```

### Creación de cuadros de lista de múltiples columnas y de selección múltiple

La propiedad Columns le permite especificar el número de columnas de un cuadro de lista estándar. Esta propiedad puede asumir los siguientes valores.

Valor	Descripción
0	Cuadro de lista normal de una columna con desplazamiento vertical.
1	Cuadro de lista de una columna con desplazamiento horizontal.
>1	Cuadro de lista de múltiples columnas con desplazamiento horizontal.
Visual Basic se encarga de los saltos de línea requeridos para los elementos de la lista y de agregar una barra de desplazamiento si es necesario. Si la lista cabe en una columna, no se agrega barra de desplazamiento. El salto a la columna siguiente se produce también automáticamente si es necesario. Observe que si un elemento de un cuadro de lista es más ancho que el ancho de una columna, el texto es truncado.	

Es posible permitir que los usuarios seleccionen varios elementos de la lista. La selección múltiple en cuadros de lista estándar se establece definiendo la propiedad MultiSelect, que puede asumir los siguientes valores.

Valor	Tipo de selección	Descripción
0	Ninguno	Cuadro de lista estándar.

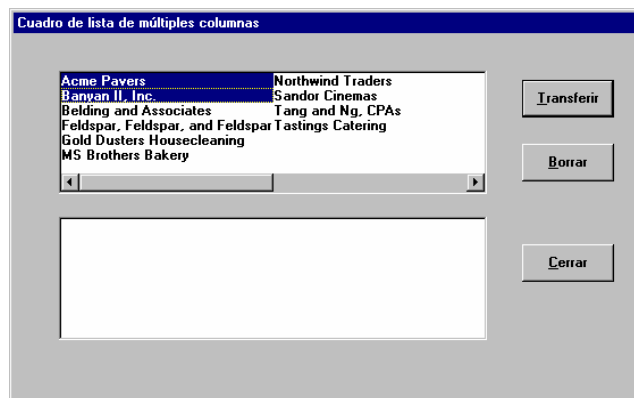


1	Selección múltiple simple	Un clic o la BARRA ESPACIADORA selecciona o desactiva elementos adicionales de la lista.
2	Selección múltiple extendida	MAYÚSCULAS+clic o MAYÚSCULAS+una tecla de flecha extiende la selección para incluir todos los elementos entre las selecciones anterior y actual. CTRL+clic selecciona o desactiva un elemento de la lista.

### Aplicación Cuadro de lista de múltiples columnas

Para crear un cuadro de lista de múltiples columnas y selección múltiple, necesita establecer las propiedades Columns y MultiSelect de un cuadro de lista. En el siguiente ejemplo, estas propiedades se usan para crear un cuadro de lista de ese tipo. Un ejemplo de un caso así puede verse en MULTI.FRM, en la aplicación ejemplo CONTROLS.VBP.

Podrá ver que cuando usted ejecuta la aplicación, el cuadro de lista contiene dos columnas, como se muestra en la figura.



**Figura Cuadro de lista de múltiples columnas**

Si dibuja el cuadro de lista lo bastante grande para contener todos los elementos en una columna, la segunda columna estará vacía; la segunda columna se ocupará y aparecerán barras de desplazamiento horizontal solamente si el cuadro de lista no es lo bastante grande. Cambie el tamaño del cuadro de lista superior y agregar elementos de lista adicionales para ver cómo Visual Basic gestiona automáticamente varias columnas.

El ejemplo usa la propiedad Selected, una matriz booleana que contiene el estado de selección de un cuadro de lista estándar, para determinar qué elementos están seleccionados. Cada elemento de la matriz se corresponde con un elemento de la lista y está establecido como **True** si el elemento está seleccionado o como **False** si no está seleccionado. Una vez de que el usuario selecciona elementos de la lista, cada elemento de la matriz es verificado para saber si está seleccionado (**True**). Si es así, el elemento es agregado a la segunda lista, un cuadro de lista estándar normal, usando el método **AddItem**.

Defina las propiedades para este ejemplo como se indica en la siguiente tabla.

Objeto	Propiedad	Valor
Form	Caption	Cuadro de lista de múltiples columnas
Cuadro de lista superior	Name	
Columns		
MultiSelect	lerSup	
2		
2-Extended		
Cuadro de lista inferior	Name	lerInf
Primer botón de comando	Name	
Caption	cmdTransferencia	
&Transferir		
Segundo botón de comando	Name	
Caption	cmdBorrar	
B&orrar		
Tercer botón de comando	Name	
Caption	cmdCerrar	
&Cerrar		

La propiedad MultiSelect le permite seleccionar un rango de valores en cuadros de lista y cuadros combinados estándar. Si hace clic en el primer elemento de la lista y después presiona MAYÚSCULAS y hace clic en el último elemento del rango (o usa las teclas MAYÚSCULAS+FLECHA ABAJO), se seleccionan todos los elementos del rango.

### Eventos de la aplicación Cuadro de lista de múltiples columnas

Agregue código al procedimiento Form\_Load para inicializar la lista superior, lerSup:

```
Private Sub Form_Load ()
```

```
    lerSup.AddItem "Acme Pavers"
```

```
    lerSup.AddItem "Belding and Associates"
```

```
    lerSup.AddItem "Banyan II, Inc."
```

```
    lerSup.AddItem "Feldspar, Feldspar, and Feldspar"
```

```
    lerSup.AddItem "Gold Dusters Housecleaning"
```

```
    lerSup.AddItem "MS Brothers Bakery"
```

```
    lerSup.AddItem "Northwind Traders"
```

```

lerSup.AddItem "Sandor Cinemas"
lerSup.AddItem "Ian's Catering"
lerSup.AddItem "Teng and Ng, CPAs"
1erSup.Selected(0) = True           ' Selección de dos elementos.
1erSup.Selected(1) = True
End Sub

```

**Nota** Es posible agregar elementos a cuadros de lista sin usar repetidamente el método **AddItem**, escribiendo elementos en la propiedad List de la ventana Propiedades. Después de introducir cada elemento, presione CTRL+ENTRAR para avanzar a la siguiente línea. Esto le permite escribir varios elementos en un cuadro de lista de múltiples columnas.

Agregue el siguiente código al procedimiento de evento 1erSup\_DblClick:

```

Private Sub 1erSup_DblClick ()
    cmdTransfer.Value = True       ' Presionar botón Transferencia.
End Sub

```

Agregue el siguiente código al procedimiento de evento Click del botón de comando Transferencia:

```

Private Sub cmdTransfer_Click ()
    For n = 0 To (lerSup.ListCount - 1)
        If lerSup.Selected(n) = True Then           ' Si seleccionado,
            lerInf.AddItem lerSup.List(n)          ' agregar a la lista.
        End If
    Next
    cmdBorrar.Enabled = True
End Sub

```

Note cómo los valores del índice de la matriz empiezan por 0 y van hasta ListCount - 1.

Agregue el siguiente código al procedimiento de evento Click del botón de comando Borrar:

```
Private Sub cmdBorrar_Click ()  
    lerInf.Clear  
    cmdBorrar.Enabled = False  
End Sub
```

Agregue el siguiente código al procedimiento de evento Click del botón de comando Cerrar:

```
Private Sub cmdCerrar_Click ()  
    Unload Me  
End Sub
```

### ***USO DE LAS BARRAS DE DESPLAZAMIENTO COMO DISPOSITIVOS DE ENTRADA.***

Aunque las barras de desplazamiento están asociadas a menudo con cuadros de texto o ventanas, algunas veces las usará como dispositivos de entrada. Como estos controles pueden indicar la posición actual en una escala, los controles barra de desplazamiento se pueden usar individualmente para controlar la entrada de programa, por ejemplo, para controlar el volumen del sonido o para ver el tiempo transcurrido en un proceso cronometrado.

Estos tipos de controles operan independientemente de otros controles y tienen su propio juego de eventos, propiedades y métodos. Los controles barra de desplazamiento no son, como las barras de desplazamiento intrínsecas de cuadros de texto, cuadros de lista, cuadros combinados o formularios MDI (los cuadros de texto y los formularios MDI tienen la propiedad ScrollBars para agregar o quitar barras de desplazamiento que se adjuntan al control).

Por ejemplo, en la aplicación ejemplo CONTROLS.VBP, la demostración Barra de desplazamiento ilustra el uso de barras de desplazamiento como se muestra en la figura.

Visual Basic le permite usar barras de desplazamiento de la misma manera. Con la Caja de herramientas, es posible crear sus propios controles barras de desplazamiento horizontales y verticales.

## La propiedad Value

La propiedad Value (cuyo valor predeterminado es 0) es un valor entero correspondiente a la posición del cuadro de desplazamiento dentro de la barra de desplazamiento. Cuando la posición del cuadro de desplazamiento se encuentra en el valor mínimo, se mueve a la posición más a la izquierda (para barras de desplazamiento horizontales) o a la posición superior (para barras de desplazamiento verticales). Cuando el cuadro de desplazamiento tiene el valor máximo, el cuadro se mueve a las posiciones más a la derecha o inferior. De igual forma, un valor a medio camino entre el inferior y el superior del rango sitúa el cuadro de desplazamiento en el medio de la barra de desplazamiento.

Además de usar clics del *mouse* para cambiar el valor de la barra de desplazamiento, un usuario también puede arrastrar el cuadro de desplazamiento a cualquier punto a lo largo de la barra. El valor resultante depende de la posición del cuadro de desplazamiento, pero estará siempre dentro del rango entre las propiedades Min y Max establecidas por el usuario.

**Nota** Min puede ser mayor que Max si se quiere que la barra de desplazamiento presente información que cambia de mayor a menor valor.

## Eventos de barra de desplazamiento

Los siguientes eventos le permiten seguir el movimiento de la barra de desplazamiento.

Evento	Descripción
Change	Ocurre después de que el cuadro de desplazamiento se mueve.
Scroll	Ocurre cuando el cuadro de desplazamiento se mueve. No ocurre si las flechas de desplazamiento o la barra de desplazamiento reciben un clic.

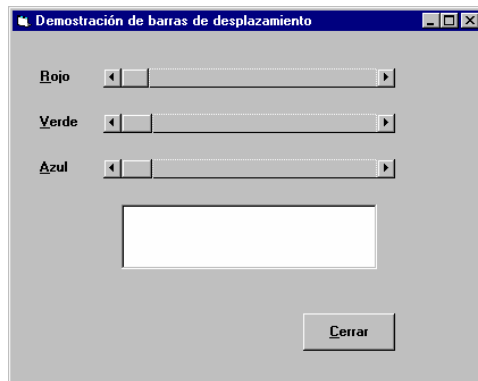
El uso del evento Scroll proporciona acceso al valor de la barra de desplazamiento cuando se está arrastrando. El evento Change ocurre una vez que se suelta el cuadro de desplazamiento o cuando se hace clic en la barra o en las teclas de desplazamiento. Es posible usar los eventos Scroll y Change conjuntamente, como se muestra en la aplicación Barras de desplazamiento, para obtener una actualización inmediata del valor de desplazamiento según el comportamiento normal de la barra de desplazamiento.

## Aplicación Barras de desplazamiento

Los colores proporcionan una interesante demostración de los controles barra de desplazamiento. El ejemplo de barra de desplazamiento mostrado en la figura permite al usuario cambiar interactivamente el color de fondo de un cuadro de texto ajustando tres barras de desplazamiento horizontales. Un

ejemplo de un caso así puede verse en SCROLL.FRM, en la aplicación ejemplo CONTROLS.VBP.

Las barras de desplazamientos se corresponden con los tres argumentos de la función **RGB**: *rojo*, *verde* y *azul*. (La función **RGB** le permite establecer colores en una aplicación Visual Basic). Cuando una barra de desplazamiento se cambia, el color RGB también cambia y se le asigna a la propiedad BackColor del cuadro de texto.



**Figura Uso de las barras de desplazamiento para controlar el color**

Comience seleccionando **Proyecto Nuevo** en el menú **Archivo** y después dibuje tres barras de desplazamiento horizontales, tres etiquetas, un cuadro de texto y un botón de comando, como se muestra en la figura . La siguiente tabla muestra los valores de las propiedades de los objetos de la aplicación.

Objeto	Propiedad	Valor
Form	Caption	Demostración de barras de desplazamiento
Etiqueta superior	Caption	&Rojo
Segunda etiqueta	Caption	&Verde
Tercera etiqueta	Caption	&Azul
Primera barra de desplazamiento	Name	
Max		
Min		
LargeChange		
SmallChange	hsbRojo	
255		
0		
35		
5		
Segunda barra de desplazamiento	Name	
Max		
Min		
LargeChange		

SmallChange	hsbVerde	
255		
0		
35		
5		
Tercera barra de desplazamiento	Name	
Max		
Min		
LargeChange		
SmallChange	hsbAzul	
255		
0		
35		
5		
Cuadro de texto	Name	
Text	txtColor	
(Vacío)		
Botón de comando	Name	
Caption	cmdCerrar	
&Cerrar		

Min y Max especifican el rango de valores posible de la barra de desplazamiento. 0 y 255 se corresponden con los valores válidos de la función **RGB**.

LargeChange y SmallChange especifican el cambio de valor cuando el usuario hace clic en la flecha de desplazamiento, o en el área entre el cuadro de desplazamiento y una flecha de desplazamiento.

El procedimiento Scroll de cada una de las barras de desplazamiento responde llamando a un procedimiento general, DoColor, para recalculer el color. El procedimiento Change también se usa para llamar al procedimiento Scroll. El uso conjunto de los eventos Scroll y Change resulta en barras de desplazamiento que cambian instantáneamente cuando se arrastra el cuadro de desplazamiento. Si no se necesita el valor de las barras de desplazamiento hasta que se haya soltado el *mouse*, entonces es posible utilizar el evento Change solo.

### **Procedimientos de evento Scroll**

Para crear los procedimientos de evento para las barras de desplazamiento, haga doble clic en la barra superior para presentar la ventana Código y después seleccione el evento Scroll en la lista de procedimientos. Agregue una línea que llame al procedimiento DoColor en cada procedimiento de evento:

```
Private Sub hsbRojo_Scroll ()
```

```
        DoColor                ' Llama al procedimiento general
DoColor.
End Sub
```

```
Private Sub hsbVerde_Scroll ()
```

```
        DoColor                ' Llama al procedimiento general
DoColor.
End Sub
```

```
Private Sub hsbAzul_Scroll ()
```

```
        DoColor                ' Llama al procedimiento general
DoColor.
End Sub
```

### **Procedimientos de evento Change**

Después, seleccione el evento Change en la lista de procedimientos. Agregue una línea que llame el procedimiento de evento Scroll:

```
Private Sub hsbRojo_Change ()
```

```
        hsbRojo_Scroll          ' Llama al de evento Scroll.
End Sub
```

```
Private Sub hsbVerde_Change ()
```

```
        hsbVerde_Scroll        ' Llama al de evento Scroll.
End Sub
```

```
Private Sub hsbAzul_Change ()
```

```
        hsbAzul_Scroll         ' Llama al de evento Scroll.
End Sub
```



**Sugerencia** Las tres barras de desplazamiento también pueden ser creadas como una matriz de controles. Por ejemplo, la primera barra de desplazamiento podría ser `hsbColor(0)`, la segunda barra de desplazamiento `hsbColor(1)` y la tercera barra de desplazamiento `hsbColor(2)`. Esto le permitiría tener un procedimiento `Change` para los tres controles.

### Procedimiento general DoColor

El procedimiento `DoColor` usa los valores de las tres barras de desplazamiento, los combina con la función **RGB** y asigna el resultado al color de fondo del cuadro de texto. La función **RGB** toma valores de entrada rojo, verde y azul, cada uno dentro de un rango de 0 a 255.

### Para crear el procedimiento DoColor

1. Seleccione “(General)” en el cuadro de lista “Objetos”.
2. En el menú **Insertar**, elija **Procedimiento**.
3. Bajo “Tipo”, seleccione la opción “Sub”.
4. Escriba `DoColor` en el cuadro “Nombre”.

Visual Basic crea automáticamente un procedimiento para `DoColor` en la ventana Código. Escriba el siguiente código en el procedimiento `DoColor`:

```
Sub DoColor ()  
    txtColor.BackColor = RGB(hsbRojo.Value, hsbVerde.Value, _  
        hsbAzul.Value)  
End Sub
```

### **OBTENER Y ESTABLECER EL ENFOQUE DE UN OBJETO.**

El *enfoque* es la capacidad de recibir entradas del usuario a través del *mouse* o del teclado. Cuando un objeto tiene el enfoque, puede recibir entradas del usuario. Por ejemplo, cuando un cuadro de texto tiene el enfoque, el usuario puede escribir texto en él.

Los eventos `GotFocus` y `LostFocus` ocurren cuando un objeto recibe o pierde el enfoque.

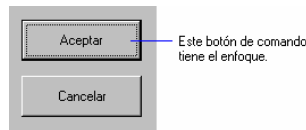
Evento	Descripción
<code>GotFocus</code>	Ocurre cuando un objeto recibe el enfoque.
<code>LostFocus</code>	Ocurre cuando un objeto pierde el enfoque. Un procedimiento de evento <code>LostFocus</code> se usa primariamente para

	verificar y validar actualizaciones, o para invertir o cambiar condiciones que se establecen en el procedimiento GotFocus del objeto.
--	---

Un objeto puede recibir el enfoque de tres maneras:

- Seleccionando un objeto en tiempo de ejecución.
- Usando una tecla de acceso para seleccionar un control en tiempo de ejecución.
- Usando el método **SetFocus** en el código.

Es posible saber cuando algunos objetos tienen el enfoque. Por ejemplo, cuando los botones de comando tienen el enfoque, se muestran con un borde resaltado alrededor del título (vea la figura).



**Figura Un botón de comando mostrando el enfoque**

Un objeto puede recibir enfoque solamente si sus propiedades Enabled y Visible se han establecido como **True**. La propiedad Enabled permite que el objeto responda a eventos generados por el usuario tales como eventos del teclado y del *mouse*. La propiedad Visible determina si un objeto es visible en la pantalla.

**Nota** Los marcos, etiquetas, menús, líneas, formas, imágenes y cronómetros no pueden recibir enfoque. Un formulario puede recibir enfoque sólo si no contiene controles que puedan recibir enfoque.

Para determinar qué control tiene el enfoque en tiempo de ejecución, pueden utilizarse las propiedades ActiveForm o ActiveForm del objeto Screen. Por ejemplo, es posible usar la propiedad ActiveForm y la sentencia **If...TypeOf** para determinar si el control con el enfoque es un cuadro de texto:

```

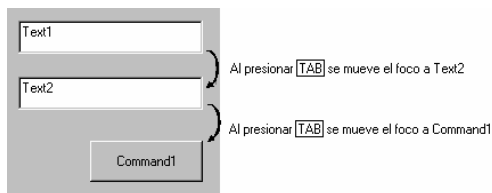
If TypeOf Screen.ActiveControl Is Textbox Then
    :
    :
End If

```

**DEFINICIÓN DEL ORDEN DE TABULACION.**

El orden de tabulación es el orden en el que un usuario pasa de un control a otro presionando la tecla TAB. Normalmente, el orden de tabulación es el mismo orden en el que se han creado los controles.

Por ejemplo, suponga que crea primero dos cuadros de texto, Texto1 y Texto2 y después un botón de comando, Comando1. Cuando se inicia la aplicación, Texto1 tiene el enfoque. Al presionar la tecla TAB, se mueve el enfoque entre controles en el orden en el que fueron creados, como se muestra en la figura.



**Figura Ejemplo de tabulación**

**Para cambiar el orden de tabulación de un control**

Establezca la propiedad TabIndex del control.

El primer control que se dibuja tiene 0 como valor predeterminado en TabIndex, el segundo tiene un valor de 1 en TabIndex y así sucesivamente. Cuando cambia la posición del orden de tabulación de un control, Visual Basic vuelve a numerar automáticamente las posiciones del orden de tabulación de los otros controles para reflejar inserciones y eliminaciones.

Por ejemplo, es posible establecer que el botón Comando1 sea el primero en el orden de tabulación con esta instrucción:

```
Comando1.TabIndex = 0
```

Los valores TabIndex de los otros controles se ajustan automáticamente hacia arriba, como se relaciona en la siguiente tabla.

<b>Control</b>	<b>TabIndex antes de que Comando1 sea el primero del orden de tabulación</b>	<b>TabIndex cuando Comando1 es el primero del orden de tabulación</b>
Texto1	0	1
Texto2	1	2
Comando1	2	0

El valor más alto de TabIndex es siempre uno menos que el número de controles del orden de tabulación (porque la numeración empieza por 0). Incluso si establece la propiedad TabIndex con un número más alto que el número de controles, Visual Basic convierte el valor al número de controles menos 1. Si asigna un número negativo a la propiedad TabIndex, Visual Basic genera un error.

**Nota** Los controles que no pueden recibir enfoque, al igual que los controles deshabilitados e invisibles no se incluyen en el orden de tabulación. Cuando un usuario presiona la tecla TAB, estos controles se saltan.

### ***ELIMINAR UN CONTROL DEL ORDEN DE TABULACIÓN.***

Normalmente, presionar la tecla TAB en tiempo de ejecución selecciona cada control según el orden de tabulación. Es posible eliminar un control del orden de tabulación estableciendo su propiedad TabStop como **False** (0).

Un control cuya propiedad TabStop ha sido establecida como **False** siempre mantiene su posición dentro del orden de tabulación, incluso aunque el control sea saltado cuando pase por los controles con la tecla TAB.

**Nota** Un grupo de botones de opción tiene una única posición dentro del orden de tabulación. El botón seleccionado (esto es, el botón con Value establecido como **True**) tiene su propiedad TabStop automáticamente establecida como **True**, mientras que los otros controles tienen su propiedad TabStop establecida como **False**.

### ***HABILITACIÓN, DESHABILITACIÓN Y CONTROL DE LA VISIBILIDAD EN TIEMPO DE EJECUCIÓN.***

En tiempo de ejecución, es posible cambiar el estado de un control estableciendo propiedades en el código. Para botones de comando hay varias acciones que resultar de utilidad, como se muestra en la siguiente tabla.

<b>Acción</b>	<b>Técnica</b>	<b>Propósito</b>
Deshabilitar	Establecer la propiedad Enabled como <b>False</b> .	Evita que el usuario elija el comando.
Habilitar	Establecer la propiedad Enabled como <b>True</b> .	Vuelve a permitir que el usuario elija el comando.
Hacer invisible	Establecer la propiedad Visible como <b>False</b> .	Deshabilita el control y lo borra del formulario visible.

Cuando deshabilita un botón de comando, está controlando el acceso del usuario a ese comando. Generalmente, se deshabilita un comando porque no es apropiado en ese momento, por ejemplo, cuando se supone que un comando borra un archivo especificado, pero el archivo es de sólo lectura.

La propiedad Enabled es un ejemplo de propiedad booleana: toma el valor **True** o **False**. Visual Basic almacena valores booleanos como enteros. Todos los enteros distintos de cero representan **True** y cero es **False**.

Por ejemplo, es posible verificar si un botón de comando está habilitado comprobando su propiedad Enabled:

```
If Comando1.Enabled = True Then
```

```
    Texto1.Text = "Botón habilitado."
```

```
Else
```

```
    Texto1.Text = "Botón deshabilitado."
```

```
End If
```

Todos los controles excepto los controles marco, forma, línea y etiqueta tienen la propiedad Enabled. Es posible usar un código similar para desactivar un control o para conocer si está habilitado.

## **ADMINISTRACIÓN DE PROYECTOS.**

Para crear una aplicación con Visual Basic se trabaja con proyectos. Un *proyecto* es una colección de archivos que se usan para generar una aplicación.

### **COMPRENSIÓN DE LOS PROYECTOS.**

Cuando desarrolla un aplicación, usa un archivo de proyecto para administrar todos los diferentes archivos que crea. Un proyecto consta de:

- Un archivo para cada formulario (.FRM)

- Un archivo binario para cada formulario que contenga controles con propiedades que tengan valores binarios (.FRX)

- Un archivo para cada módulo de clase (.CLS)

- Un archivo para cada módulo estándar (.BAS)

- Uno o más archivos con controles personalizados (.VBX y .OCX)

- Un único archivo de recursos (.RES)

- Un archivo de proyecto que realiza el seguimiento de todos los componentes (.VBP)

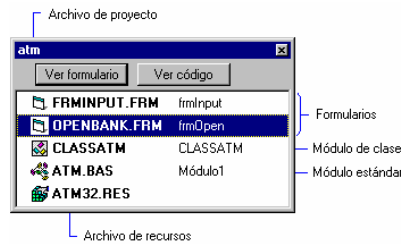
El *archivo de proyecto* es simplemente una lista de todos los archivos y objetos asociados con el proyecto, así como información sobre las opciones de entorno establecidas. Esta información se actualiza cada vez que guarda el proyecto. Todos los archivos y objetos también se pueden compartir con otros proyectos.

Cuando ha completado todos los archivos del proyecto puede convertir el proyecto en un archivo ejecutable (.EXE): en el menú **Archivo**, elija el comando **Crear archivo EXE**.

### **VENTANA PROYECTO.**

Cuando crea, agrega o elimina archivos editables de un proyecto, Visual Basic refleja los cambios en la ventana Proyecto, que contiene una lista actualizada de los archivos del proyecto. Cuando inicia Visual Basic o crea un proyecto nuevo, se cargan automáticamente los componentes de un archivo de proyecto predeterminado.

La ventana Proyecto de la figura muestra los tipos de archivos que puede incluir en un proyecto Visual Basic.



**Figura Ventana Proyecto**

## ***EL ARCHIVO DE PROYECTO.***

Cada vez que guarda un proyecto, Visual Basic actualiza el archivo de proyecto (.VBP). Un archivo de proyecto contiene la misma lista de archivos que aparece en la ventana Proyecto, así como referencias a los controles y objetos insertables que se usan en el proyecto. El formato específico de la información almacenada en el archivo .VBP se describe en el apéndice A, “Formatos de archivo de formulario y proyecto”.

Puede abrir un archivo de proyecto existente haciendo doble clic en su icono o seleccionando el comando **Abrir proyecto** en el menú **Archivo**.

## ***ELEMENTOS DE UN PROYECTO.***

Las siguientes secciones describen los diferentes tipos de archivos y objetos que puede incluir en un proyecto.

### **Módulos de formulario**

Los formularios (extensión de nombre de archivo .FRM) pueden contener descripciones en forma de texto del formulario y sus controles, incluyendo la configuración de sus propiedades. También pueden contener declaraciones a nivel de formulario de constantes, variables y procedimientos externos, procedimientos que gestionan eventos y procedimientos generales

### **Módulos de clase**

### **Módulos estándar**

Los módulos estándar (extensión de nombre de archivo .BAS) pueden contener declaraciones públicas o a nivel de módulo de tipos, constantes, variables, procedimientos externos y procedimientos públicos.

## Archivos de recursos

Los archivos de recursos (extensión de nombre de archivo .RES) contienen mapas de bits, cadenas de texto u otros datos que puede modificar sin volver a editar el código. Por ejemplo, si piensa traducir su aplicación a un idioma extranjero, puede guardar todas las cadenas de texto de la interfaz de usuario y los mapas de bits en un archivo de recursos, y simplemente traducir el archivo de recursos en vez de la aplicación completa. Un proyecto sólo puede contener un archivo de recursos.

## Controles personalizados OLE y objetos insertables

Los controles personalizados OLE (extensión de nombre de archivo .OCX) y los objetos insertables son controles personalizados con una interfaz visible. Cuando instala Visual Basic, los archivos que contienen los controles personalizados OLE incluidos en Visual Basic se copian al subdirectorio \WINDOWS\SYSTEM.

Los *objetos insertables*, como un objeto hoja de cálculo Microsoft Excel, son componentes que puede usar como bloques prefabricados para generar soluciones integradas. Una *solución integrada* puede contener datos en diferentes formatos, como hojas de cálculo, mapas de bits y texto creados por diferentes aplicaciones.

## Controles personalizados VBX

Los controles personalizados VBX (extensión de nombre de archivo .V BX) son parecidos a los controles personalizados OLE, pero sólo pueden usarse con la versión de 16 bits de Visual Basic.

## Controles estándar

Los controles estándar están dentro del archivo .EXE de Visual Basic. Los controles estándar, como Botón de comando o Marco, siempre están incluidos en la Caja de herramientas, al contrario que los controles personalizados OLE, los controles personalizados VBX y los objetos insertables, que se pueden agregar y eliminar de la Caja de herramientas.

Cuatro comandos del menú **Archivo** le permiten crear, abrir y guardar proyectos.

Comando de menú	Descripción
<b>Nuevo proyecto</b>	Cierra el proyecto actual, preguntándole si desea guardar los archivos modificados. Después Visual Basic crea un proyecto nuevo, agregando un formulario nuevo y todos los formularios, módulos, referencias y controles personalizados incluidos en el archivo



	AUTO32LD.VBP. (Consulte la sección <u>"Edición del archivo Autocarga"</u> ).
<b>Abrir proyecto</b>	Cierra el proyecto actual, preguntándole si desea guardar los cambios. Después Visual Basic abre un proyecto existente, incluyendo los formularios, módulos y controles personalizados enumerados en su archivo de proyecto (.VBP).
<b>Guardar proyecto</b>	Actualiza el archivo de proyecto del proyecto actual y todos sus módulos de formulario, estándar y de clase.
<b>Guardar proyecto como</b>	Actualiza el archivo de proyecto del proyecto actual, guardando el archivo de proyecto con un nombre que especifica el propio usuario. Visual Basic también le pide que guarde cualquier formulario o módulo que haya cambiado.

Observe que aunque sólo puede tener un proyecto abierto al mismo tiempo en una instancia de Visual Basic, puede compartir archivos entre proyectos. Un archivo concreto, como un formulario, puede formar parte de varios proyectos.

Cuatro comandos del menú **Archivo** le permiten crear, abrir y guardar proyectos.

<b>Comando de menú</b>	<b>Descripción</b>
<b>Nuevo proyecto</b>	Cierra el proyecto actual, preguntándole si desea guardar los archivos modificados. Después Visual Basic crea un proyecto nuevo, agregando un formulario nuevo y todos los formularios, módulos, referencias y controles personalizados incluidos en el archivo AUTO32LD.VBP.
<b>Abrir proyecto</b>	Cierra el proyecto actual, preguntándole si desea guardar los cambios. Después Visual Basic abre un proyecto existente, incluyendo los formularios, módulos y controles personalizados enumerados en su archivo de proyecto (.VBP).
<b>Guardar proyecto</b>	Actualiza el archivo de proyecto del proyecto actual y todos sus módulos de formulario, estándar y de clase.
<b>Guardar proyecto como</b>	Actualiza el archivo de proyecto del proyecto actual, guardando el archivo de proyecto con un nombre que especifica el propio usuario. Visual Basic también le pide que guarde cualquier formulario o módulo que haya cambiado.

Observe que aunque sólo puede tener un proyecto abierto al mismo tiempo en una instancia de Visual Basic, puede compartir archivos entre proyectos. Un archivo concreto, como un formulario, puede formar parte de varios proyectos. Para obtener más información sobre el uso compartido de archivos, consulte la siguiente sección “Adición, eliminación y almacenamiento de archivos”. Para obtener información sobre el uso compartido de archivos entre diferentes versiones de Visual Basic, consulte el apéndice C, “Compatibilidad con otras versiones”.

## **ADICIÓN, ELIMINACIÓN Y ALMACENAMIENTO DE ARCHIVOS.**

Cuatro comandos del menú **Archivo** le permiten agregar, eliminar y guardar archivos.

<b>Comando de menú</b>	<b>Descripción</b>
<b>Agregar archivo</b>	Agrega a un proyecto un módulo de formulario, módulo estándar, módulo de clase o archivo de recursos existente.
<b>Eliminar archivo</b>	Elimina de un proyecto un módulo de formulario, módulo estándar, módulo de clase o archivo de recursos.
<b>Guardar archivo</b>	Guarda en un proyecto un módulo de formulario, estándar o de clase.
<b>Guardar archivo como</b>	Guarda en un proyecto un módulo de formulario, estándar o de clase con el nombre de archivo especificado.

Si *elimina* un archivo de un proyecto, Visual Basic actualiza esta información en el archivo de proyecto al guardar el proyecto. Sin embargo, si elimina el archivo fuera de Visual Basic, éste no puede actualizar el archivo de proyecto; de esta manera, cuando abre el proyecto, Visual Basic presenta un mensaje de error advirtiéndole de que falta un archivo. Para agregar un archivo nuevo o renombrado a un proyecto, abra el proyecto y, en el menú **Archivo**, elija el comando **Agregar archivo**.

Si *agrega* un archivo a un proyecto simplemente está incluyendo en el proyecto una referencia al archivo existente, no agregando una copia de un archivo. Así, si efectúa cambios en un archivo y lo guarda, los cambios afectarán a cualquier proyecto que contenga el archivo. Para cambiar un archivo sin afectar otros proyectos, elija **Guardar archivo como** en el menú **Archivo** y guarde el archivo con un nombre nuevo.

**Nota** Puede arrastrar y colocar archivos desde el Explorador de Windows, el Administrador de archivos o el Entorno de red a la ventana Proyecto para

agregarlos al proyecto. También puede arrastrar y colocar archivos .OCX en la Caja de herramientas para agregar controles nuevos.

### ***COMBINACIÓN DE TEXTO.***

También puede insertar texto de otros archivos en alguno de sus módulos.

#### **Para insertar un archivo de texto en su código**

1. En la ventana Proyecto, seleccione el formulario o módulo en el que desea insertar código.
2. Elija **Ver código** y mueva el cursor al punto donde desee insertar el código.
3. En el menú **Insertar**, elija **Archivo**.
4. Seleccione el nombre del archivo de texto que quiere insertar y elija "Aceptar".

## **FUNDAMENTOS DE PROGRAMACIÓN.**

Visual Basic es un lenguaje de programación completo que acepta las estructuras de programación que se encuentran en la mayoría de los lenguajes de programación modernos.

### ***ESTRUCTURA DE UNA APLICACIÓN VISUAL BASIC.***

Una aplicación puede contener varios tipos diferentes de archivos:

Los módulos de formulario (.FRM) contienen los elementos visuales de un formulario, incluyendo todos sus controles y el código Basic asociado con él.

Los módulos estándar (.BAS) y de clase (.CLS) contienen código Basic.

Los controles personalizados (.VBX u .OCX) incluyen controles especializados, así como versiones mejoradas de controles estándar.

Un único archivo de recursos (.RES) contiene cadenas y mapas de bits usados en la aplicación.

### ***COMO FUNCIONA UNA APLICACIÓN CONTROLADA POR EVENTOS.***

Un *evento* es una acción reconocida por un formulario o control. Las aplicaciones controladas por eventos ejecutan código Basic en respuesta a un evento. Cada formulario y control de Visual Basic tiene un juego de eventos predefinido. Si ocurre uno de estos eventos, Visual Basic invoca el código del procedimiento de evento asociado.

Aunque los objetos de Visual Basic *reconocen* automáticamente un juego predefinido de eventos, usted determina si *responden* a un evento concreto y cómo. Cuando se desea que un control responda a un evento, se escribe el código llamado *procedimiento de evento* para ese evento.

Muchos objetos reconocen el mismo evento, aunque objetos diferentes pueden ejecutar procedimientos de evento diferentes cuando éste ocurre. Por ejemplo, si el usuario hace clic en un formulario, se ejecuta el procedimiento de evento **Form\_Click**; si el usuario hace clic en un botón de comando llamado "Command1", se ejecuta el procedimiento de evento **Command1\_Click**.

Esto es lo que ocurre en una típica aplicación controlada por eventos:

1. La aplicación se inicia y automáticamente se carga y presenta el formulario inicial.

2. Un formulario o control recibe un evento. El evento puede ser provocado por el usuario (por ejemplo, presionando una tecla), por el sistema (por ejemplo, un evento de temporizador) o indirectamente por el código (por ejemplo, un evento **Load** cuando se carga un formulario).
3. Si existe un procedimiento correspondiente a dicho evento, se ejecuta.
4. La aplicación espera el siguiente evento.

**Nota** Muchos eventos se producen junto con otros eventos. Por ejemplo, cuando se produce el evento **DbClick**, también se producen los eventos **MouseDown**, **MouseUp** y **Click**.

### Control por eventos frente a programación tradicional

En una aplicación tradicional o “procedural”, la propia aplicación y no un evento, controla las partes de código que se ejecutan. La ejecución comienza por la primera línea de código ejecutable y sigue un camino definido a lo largo de la aplicación, llamando a procedimientos cuando es necesario.

En programas controlados por eventos, una acción del usuario o un evento del sistema ejecutan un procedimiento de evento. Así, el orden de ejecución del código depende del evento que se produzca, que depende, a su vez, de lo que haga el usuario. Esta es la esencia de las *interfaces* gráficas de usuario y de la programación controlada por eventos: el usuario manda y el código responde.

Como no puede predecirse lo que va a hacer el usuario, el código debe asumir ciertas hipótesis sobre “el estado del mundo” cuando se ejecuta. Cuando tenga que hacer suposiciones (por ejemplo, que un cuadro de texto contiene texto antes de que se presione un botón de comando), debe estructurar su aplicación de manera que dichas suposiciones sean siempre válidas (por ejemplo, desactivando el botón de comando y activándolo solamente en el evento **Change** del cuadro de texto).

El código puede desencadenar eventos adicionales al ejecutar ciertas operaciones. Por ejemplo, la carga de un formulario provoca el evento de formulario **Load**, y cambiar la propiedad **Text** de un cuadro de texto provoca el evento **Change**.

**Nota** Evite que en un procedimiento de evento se realicen operaciones que provoquen el mismo evento otra vez, esto origina un *evento en cascada*. Por ejemplo, si el código del procedimiento de evento **Change** de un cuadro de texto define la propiedad **Text** de dicho cuadro de texto, se vuelve a llamar al procedimiento de evento **Change**, el código vuelve a definir la propiedad **Text**, que vuelve a provocar el evento **Change**, y así hasta que Visual Basic genera el error No hay suficiente espacio de pila.

### Código que se ejecuta inicialmente

De manera predeterminada, el primer formulario de una aplicación se denomina *formulario inicial*. Cuando se inicia una aplicación, se presenta dicho

formulario (de modo que el primer código que se ejecuta es el del evento **Form\_Initialize** de ese formulario). Si quiere presentar un formulario distinto cuando inicie su aplicación, debe cambiar el formulario inicial.

### Para cambiar el formulario inicial

1. En el menú **Herramientas**, elija **Opciones**.
2. Elija la ficha **Proyecto**.
3. En el cuadro de lista "Formulario inicial", seleccione el formulario que desee como nuevo formulario inicial.
4. Elija "Aceptar".

Algunas veces deseará que su aplicación se inicie sin cargar inicialmente ningún formulario. Por ejemplo, quizás desee ejecutar código que cargue un archivo de datos y después presente uno entre varios formularios, dependiendo del contenido del archivo de datos. Puede hacerlo creando un procedimiento **Sub** llamado "Main" en un módulo estándar. Este procedimiento tiene que ser un procedimiento **Sub**, y no puede estar en un módulo de formulario. Después, en el menú **Herramientas**, elija **Opciones**, seleccione la ficha **Proyecto** y seleccione **Sub Main** en el cuadro "Formulario inicial".

### Terminación de una aplicación

Una aplicación controlada por eventos termina su ejecución cuando todos sus formularios están cerrados y no se ejecuta ningún código. Si hay algún formulario oculto cuando se cierra el último formulario visible, parecerá que la aplicación ha terminado (porque no hay formularios visibles) pero de hecho continuará su ejecución hasta que se cierren todos los formularios ocultos. Esta situación puede producirse por algún acceso a las propiedades o controles de un formulario no cargado, que provoca implícitamente su carga sin mostrarlo.

La mejor manera de evitar este problema al cerrar una aplicación es asegurarse de que todos los formularios están *descargados*. Si tiene más de un formulario, puede usar la colección **Forms** y la instrucción **Unload**. Por ejemplo, suponga que el formulario principal tiene un botón de comando llamado **cmdQuit** que permite que el usuario termine el programa. Si su aplicación sólo tiene un formulario, el procedimiento de evento **Click** podría ser tan sencillo como este:

```
Private Sub cmdClose_Click ()
```

```
    Unload Me
```

```
End Sub
```

Si la aplicación usa múltiples formularios, puede descargarlos escribiendo código en el procedimiento de evento **Unload** del formulario principal. Puede usar la colección `Forms` para encontrar y cerrar todos los formularios.

Puede haber casos en los que necesite terminar su aplicación sin importarle el estado de los formularios u objetos existentes. Con este propósito, Visual Basic proporciona la instrucción **End**.

La instrucción **End** termina una aplicación inmediatamente; después de la instrucción **End** no se ejecuta código alguno y no se producen más eventos. En particular, Visual Basic no ejecutará los procedimientos de evento **QueryUnload**, **Unload** ni **Terminate** de ningún formulario. Se liberarán las referencias a objetos, pero si ha definido sus propias clases, Visual Basic no ejecutará los eventos **Terminate** de los objetos creados a partir de sus propias clases.

Además de la instrucción **End**, la instrucción **Stop** detiene una aplicación. Sin embargo, sólo debe usar la instrucción **Stop** en la fase de depuración, porque no libera las referencias a objetos.

## **VARIABLES.**

A menudo es necesario almacenar valores temporalmente cuando se realizan cálculos en Visual Basic. Por ejemplo, puede que desee calcular varios valores, compararlos y realizar operaciones distintas sobre ellos dependiendo del resultado de la comparación. Si desea comparar dichos valores, pero no almacenarlos en una propiedad, es necesario retenerlos.

Visual Basic, como la mayoría de los lenguajes de programación, usa *variables* para almacenar valores. Las variables tienen un *nombre* (la palabra que se usa para referirse al valor contenido en la variable) y un *tipo de datos* (que determina el tipo de datos que la variable puede almacenar).

## **DECLARACIÓN DE VARIABLES.**

En Visual Basic no es obligatorio declarar o crear las variables explícitamente. Si en el código usa un nombre que no pertenece a ninguna variable existente, Visual Basic crea una variable nueva con ese nombre. Sin embargo, declarar las variables explícitamente es una buena práctica de programación porque puede evitar problemas de depuración posteriormente. Puede especificar que sea obligatorio declarar las variables de la siguiente manera: en el menú **Herramientas**, elija **Opciones**, seleccione la ficha **Entorno** y elija la opción **Declaración de variables requerida**; o bien, agregue la instrucción **Option Explicit** en la sección "Declarations" de cada módulo de formulario, estándar o de clase. Esto le obligará a declarar las variables.

Las variables se declaran con la instrucción **Dim**:





conversiones entre estos tipos de datos cuando los asigna a una variable **Variant**; Visual Basic realiza automáticamente las conversiones necesarias. Sin embargo, si sabe que una variable siempre almacenará datos de un tipo concreto, Visual Basic puede manejar dichos datos más eficientemente si declara una variable de dicho tipo.

## **MÓDULOS.**

El código de Visual Basic se almacena en módulos. Hay tres tipos de módulos: módulos de clase, módulos de formulario (los módulos de formulario son un tipo de módulo de clase) y módulos estándar.

Las aplicaciones sencillas pueden consistir en un solo formulario, y todo el código de la aplicación reside en ese módulo de formulario. A medida que las aplicaciones crecen y se hacen más sofisticadas, se agregan formularios adicionales. En un momento del futuro, podría encontrar que hay código común que desea ejecutar en varios formularios. No quiere duplicar este código en ambos formularios, así que se crea un módulo separado con un procedimiento que implementa el código común. Este módulo separado debe ser un *módulo estándar*. Con el tiempo, podrá generar una biblioteca de módulos que contengan procedimientos compartidos.

Cada módulo estándar, de clase o de formulario puede contener:

**Declaraciones.** En los módulos de formulario, de clase o estándar, puede incluir declaraciones de constante, tipo, variable y procedimiento DLL a nivel de módulo.

**Procedimientos.** Un procedimiento **Sub**, **Function** o **Property** contiene partes de código que pueden ejecutarse como una unidad.

## **MÓDULOS DE FORMULARIO.**

Los módulos de formulario (extensión de nombre de archivo .FRM) son la base de cualquier aplicación Visual Basic. Pueden contener descripciones gráficas del formulario y sus controles, incluyendo la configuración de las propiedades. También pueden contener declaraciones a nivel de formulario de tipos, constantes, variables y procedimientos externos, procedimientos que manejan eventos y procedimientos generales. Casi todo lo que se aplica a los módulos de clase puede aplicarse a los módulos de formulario. Los formularios son simplemente módulos de clase que pueden incluir controles y presentar una ventana de formulario. Los formularios constituyen la parte de su aplicación que es visible para los usuarios en tiempo de ejecución.

## **MÓDULOS ESTÁNDAR.**

Los módulos estándar (extensión de nombre de archivo .BAS) contienen procedimientos y declaraciones usados en común por otras partes de la aplicación. Pueden contener declaraciones a nivel global o a nivel de módulo de tipos, constantes, variables, procedimientos externos y procedimientos globales.

## **MÓDULOS DE CLASE.**

Los módulos de clase (extensión de nombre de archivo .CLS) son la base de la programación orientada a objetos en Visual Basic. Para crear objetos nuevos puede escribir código en módulos de clase. Estos nuevos objetos pueden incluir sus propios métodos y propiedades personalizados, aunque los objetos personalizados no pueden tener eventos propios. Todas las propiedades y métodos que se crean se pueden usar también en otros objetos de la aplicación. También se puede usar la palabra clave **New** para crear múltiples copias de sus objetos.

## **PROCEDIMIENTOS.**

Puede simplificar la programación dividiendo los programas en componentes lógicos más pequeños. Estos componentes (llamados *procedimientos*) pueden transformarse en los bloques básicos que le permiten ampliar y extender Visual Basic.

Los procedimientos son útiles para condensar tareas repetitivas o compartidas, como cálculos usados frecuentemente, manipulación de texto y controles, y operaciones de base de datos.

La programación con procedimientos tiene dos ventajas importantes:

- Los procedimientos le permiten dividir los programas en unidades lógicas separadas, cada una de las cuales puede ser depurada con más facilidad que un programa completo sin procedimientos.

- Los procedimientos usados en un programa pueden actuar como bloques básicos en otros programas, normalmente con una mínima modificación o incluso sin cambios. En Visual Basic hay varios tipos de procedimientos.

Un procedimiento puede ser **Sub**, **Function** o **Property**.

- Los procedimientos **Sub** no devuelven ningún valor.

- Los procedimientos **Function** devuelven un valor.

Los procedimientos **Property** pueden devolver y asignar valores, y establecer referencias a objetos.

## PROCEDIMIENTOS SUB.

La sintaxis de un procedimiento **Sub** es:

```
[Private|Public][Static]Sub nombreDeProcedimiento (argumentos)  
instrucciones
```

### End Sub

Los *argumentos* son una lista de nombres de argumentos, separados por comas cuando hay más de uno. Cada argumento parece una declaración de variable y actúa como una variable en el procedimiento.

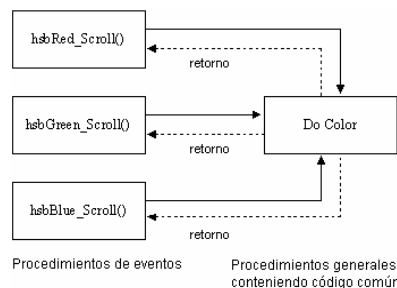
Cada vez que se llama a un procedimiento, se ejecutan las *instrucciones* comprendidas entre **Sub** y **End Sub**. Los procedimientos **Sub** se pueden colocar en módulos estándar, módulos de clase y módulos de formulario. Los procedimientos **Sub** son **Public** de forma predeterminada para todos los módulos, lo que significa que se pueden llamar desde cualquier parte de la aplicación.

En Visual Basic, es útil distinguir entre dos tipos de procedimientos **Sub**, *procedimientos generales* y *procedimientos de evento*.

### Procedimientos generales

Un procedimiento general indica a la aplicación cómo realizar una tarea específica. Una vez definido un procedimiento general, la aplicación debe invocarlo específicamente. Por el contrario, un procedimiento de evento permanece inactivo hasta que se le llama para responder a un evento originado por el usuario o desencadenado por el sistema.

¿Por qué crear procedimientos generales? Una razón es la posibilidad de que varios procedimientos de eventos distintos necesiten realizar las mismas acciones. Una buena estrategia de programación consiste en poner instrucciones comunes en un procedimiento separado (un procedimiento general) y hacer que los procedimientos de evento lo llamen. Esto elimina la necesidad de duplicar código y también facilita el mantenimiento de la aplicación. La figura ilustra el uso de un procedimiento general.



## Figura Los procedimientos de evento llaman a los procedimientos generales

### Procedimientos de evento

Cuando un objeto de Visual Basic reconoce que ha ocurrido un evento, invoca automáticamente al procedimiento de evento cuyo nombre corresponde al evento en cuestión. Como el nombre establece una asociación entre el objeto y el código, se dice que los procedimientos de evento son adjuntos a formularios y controles.

Un procedimiento de evento de un control combina el nombre del control (especificado en la propiedad **Name**), un subrayado (\_) y el nombre del evento. Por ejemplo, si quiere que un botón de comando llamado “MiBotón” invoque un procedimiento de evento cuando se hace clic en él, use el procedimiento **MiBotón\_Click**.

Un procedimiento de evento de un formulario combina la palabra “Form”, un subrayado y el nombre del evento. Si quiere que un formulario invoque un procedimiento de evento cuando se hace clic en él, use el procedimiento **Form\_Click**. (Como los controles, los formularios tienen nombres únicos, pero no se usan en los nombres de los procedimientos de evento). Si usa el formulario MDI, el procedimiento de evento combina la palabra “MDIForm”, un subrayado y el nombre del evento, como por ejemplo **MDIForm\_Load**.

Todos los procedimientos de evento usan la misma sintaxis general.

#### Sintaxis de eventos de controles

#### Sintaxis de eventos de formulario

```
Private Sub nombreDeControl_nombreDeEvento (argumentos )  
    bloqueDeInstrucciones  
End Sub Private Sub Form_nombreDeEvento (argumentos)  
    bloqueDeInstrucciones  
End Sub
```

Aunque puede escribir procedimientos de evento desde cero, es más fácil usar los procedimientos de código proporcionados por Visual Basic, que incluyen automáticamente los nombres de procedimiento correctos. Se puede seleccionar una plantilla en la ventana Código seleccionando un objeto en el cuadro “Objeto” y seleccionando después un procedimiento en el cuadro “Procedimiento”.

También es conveniente cambiar los nombres de los controles *antes* de empezar a escribir sus procedimientos de evento. Si cambia el nombre del control *después* de adjuntarle un procedimiento, también debe cambiar el nombre del procedimiento para que coincida con el nuevo nombre del control. De lo contrario, Visual Basic no podrá relacionar el control con el procedimiento. Cuando el nombre de un procedimiento no coincide con el nombre de un control, se convierte en un procedimiento general: En la ventana

Código, seleccione "(General)" en el cuadro "Objeto", y a continuación seleccione el nombre del procedimiento en el cuadro "Procedimiento".

**Nota** En Visual Basic 4.0, los procedimientos de evento incluyen de manera predeterminada la palabra clave **Private**.

### **Para obtener más información**

Visual Basic reconoce gran variedad de eventos para cada clase de formulario y control. Para consultar la explicación de todos los eventos, busque *procedimientos de evento* en la Ayuda.

## **PROCEDIMIENTOS FUNCIÓN.**

Visual Basic incluye funciones proporcionadas por el sistema, también denominadas intrínsecas, como **Sqr**, **Cos** o **Chr**. Además, puede usar la instrucción **Function** para escribir sus propios procedimientos **Function**.

La sintaxis de un procedimiento **Function** es:

```
[Private|Public][Static]Function nombreDeProcedimiento (argumentos) [As  
tipo]  
    instrucciones  
End Function
```

Al igual que un procedimiento **Sub**, un procedimiento **Function** es un procedimiento independiente que puede aceptar argumentos, ejecutar una serie de instrucciones y cambiar el valor de sus argumentos. Los argumentos de un procedimiento **Function** funcionan exactamente de la misma manera que los argumentos de un procedimiento **Sub**. Además de la palabra clave **Function**, hay tres diferencias entre los procedimientos **Sub** y **Function**:

Generalmente, se llama a una función incluyendo el nombre del procedimiento de la función y los argumentos a la derecha de una instrucción o expresión mayor.

Los procedimientos **Function** tienen tipos de datos, como las variables. Esto determina el tipo del valor devuelto. (En ausencia de la cláusula **As**, el tipo es el predeterminado **Variant**).

El valor devuelto es el que se asigna al *nombreDeProcedimiento*. Cuando el procedimiento **Function** devuelve un valor, este valor puede pasar a formar parte de una expresión mayor.

Por ejemplo, puede escribir una función que calcule el tercer lado, o hipotenusa, de un triángulo rectángulo, dados los valores de los otros dos lados:

```
Function Hypotenuse (A, B)
```

Hypotenuse = Sqr(A ^ 2 + B ^ 2)

End Function

La llamada a un procedimiento **Function** se realiza de la misma manera que a cualquiera de las funciones incorporadas de Visual Basic:

Label1.Caption = Hypotenuse(CInt(Text1.Text), CInt(Text2.Text))

X = Hypotenuse(Width, Height)

### Para obtener más información

Para obtener detalles adicionales sobre el procedimiento **Function**, busque *procedimientos Function* en la Ayuda.

## **PROCEDIMIENTOS PROPERTY.**

Los procedimientos **Property** sirven para crear y manipular propiedades personalizadas.

Los procedimientos **Property** se pueden usar para crear propiedades de sólo lectura de formularios, módulos estándar y módulos de clase.

Los procedimientos **Property** se deben usar en lugar de variables públicas cuando, al definir el valor de una propiedad, tiene que ejecutar código.

Al contrario que las variables públicas, los procedimientos **Property** pueden tener asignadas cadenas de ayuda en el Examinador de objetos.

Cuando se crea un procedimiento **Property**, la nueva propiedad pertenece al módulo que contiene el procedimiento. Visual Basic proporciona tres clases de procedimientos **Property**, como se describe en la siguiente tabla.

<b>Procedimiento Property</b>	<b>Descripción</b>
<b>Property Let</b>	Un procedimiento que establece el valor de una propiedad.
<b>Property Get</b>	Un procedimiento que devuelve el valor de una propiedad.
<b>Property Set</b>	Un procedimiento que establece una referencia a un objeto.

## Propiedades de sólo lectura

Puede usar procedimientos **Property** para agregar propiedades de sólo lectura personalizadas a objetos. Esto es útil en cualquier situación donde:

- desea que la propiedad sea de sólo lectura para los usuarios
- o desea que se ejecute código relacionado al definir la propiedad.

Por ejemplo, un sistema podría establecer una propiedad **WorkOrderPriority** basada en diversos valores como fecha de entrega o estado. En este caso probablemente será necesario que el sistema haga algo antes de determinar el valor de la propiedad **Priority**; y también puede ser necesario que la propiedad sea de sólo lectura.

Este ejemplo demuestra la creación de una propiedad de sólo lectura usando un procedimiento **Property Get**.

Para usar este ejemplo, abra un proyecto nuevo con un único formulario. Es especialmente valioso recorrer esta aplicación instrucción a instrucción. De esta manera, se puede usar **Inspección instantánea** para comprobar el valor de las expresiones y variables en cada paso y ver el orden de proceso.

Declare una sola variable **Date**:

```
Dim datMain as Date
```

El procedimiento de evento **Activate** del formulario inicializa el valor de `datMain` y llama al procedimiento **Property Get**:

```
Private Sub Form_Activate()  
    'La fecha y hora actuales se pasan a la variable.  
    datMain = Now  
    'La variable se pasa al procedimiento Property Get  
    'y se llama al procedimiento.  
    MsgBox Created(datMain)  
End Sub
```

El valor pasado al procedimiento **Property Get** se asigna a la propiedad **Created** y el valor de la propiedad se devuelve a la función **MsgBox** en el evento **Activate** del formulario.

```
Property Get Created(x) As Date
```

'Puede agregar proceso extra al procedimiento.

Created = x

'Adjunta el valor a la propiedad.

End Property

### **Ejecución de código cuando se establecen propiedades**

Se pueden crear procedimientos **Property Let**, **Property Set** y **Property Get** que compartan el mismo nombre. Haciendo esto, puede crear un grupo de procedimientos relacionados que funcionan conjuntamente. Cuando se usa un nombre para un procedimiento **Property**, ese nombre no se puede usar para nombrar un procedimiento **Sub** o **Function**, ni una variable.

La instrucción **Property Let** permite crear un procedimiento que establece el valor de una propiedad. Un ejemplo podría ser un procedimiento **Property** que cree una propiedad **Inverted** para un formulario. Esta sintaxis llamaría al procedimiento **Property Let**:

```
Form1.Inverted = True
```

El trabajo de invertir el mapa de bits del formulario se realizaría dentro del procedimiento **Property Let**:

```
Private IsInverted As Boolean
```

```
Property Let Inverted(X As Boolean)
```

```
    IsInverted = X          ' Almacena el nuevo valor de la  
    propiedad.
```

```
    If IsInverted Then
```

```
        .  
        . (instrucciones)
```

```
    Else
```

```
        .  
        . (instrucciones)
```



End If

End Property

La variable a nivel de formulario **IsInverted** almacena el valor de la propiedad. Se declara **Private** para que el usuario sólo pueda cambiarla usando el procedimiento **Property Let**. Use un nombre que permita reconocer con facilidad que la variable se usa para la propiedad.

El procedimiento **Property Get** se usaría para devolver el estado actual de la propiedad **Inverted**:

Property Get Inverted() As Boolean

Inverted = IsInverted

End Property

Los procedimientos **Property** facilitan la ejecución de código al establecer el valor de una propiedad. Puede usar procedimientos **Property** para lo siguiente:

Efectuar procesos *antes* de que se defina el valor de una propiedad para determinar dicho valor. En el ejemplo de la propiedad **Priority** de la sección anterior, el valor de la propiedad se determina mediante el proceso que depende de la entrada del usuario y de valores incluidos en el código por el programador.

Efectuar procesos *después* de que se defina el valor de una propiedad, basado en el nuevo valor. En el ejemplo de la propiedad **Inverted** de esta sección, dependiendo de si el mapa de bits está invertido o no, se ejecutan diferentes ramificaciones de la estructura **If Then...Else...End If**.

### Declaración de procedimientos Property

La sintaxis para declarar un procedimiento **Property** es:

```
[Public|Private][Static] Property {Get|Let|Set} nombreDePropiedad _  
[(argumentos)][As tipo]  
    instrucciones
```

**End Property**

Los procedimientos **Property** se usan generalmente en parejas: **Property Let** con **Property Get** y **Property Set** con **Property Get**. La declaración de un procedimiento **Property Get** aislado es como la declaración de una propiedad de sólo lectura. Se pueden usar los tres tipos de procedimientos conjuntamente, pero sólo sería útil con **Variants**. Esto se debe a que **Property Get** no está indicado para ser usado con objetos y **Property Set** sí lo está,

siendo **Variant** el único tipo de datos que permite gestionar las dos variedades de datos.

### Argumentos en procedimientos Property

La siguiente tabla demuestra los requisitos de los argumentos en las declaraciones de procedimientos **Property**.

Procedimiento	Sintaxis de declaración
<b>Property Get</b>	<b>Property Get</b> <i>nombreDePropiedad</i> (1,..., <i>n</i> ) <i>As tipo</i>
<b>Property Let</b>	<b>Property Let</b> <i>nombreDePropiedad</i> (1,..., <i>n</i> , <i>n+1</i> )
<b>Property Set</b>	<b>Property Set</b> <i>nombreDePropiedad</i> (1,..., <i>n</i> , <i>n+1</i> )

Todos los argumentos, del primero al penúltimo (1,..., *n*), deben compartir los mismos nombres y tipos de datos en todos los procedimientos **Property** del mismo nombre.

La declaración de un procedimiento **Property Get** requiere un argumento menos que las declaraciones de los procedimientos **Property Let** y **Property Set** relacionados. El tipo de datos de un procedimiento **Property Get** debe ser el mismo que el del último argumento (*n+1*) de las declaraciones **Property Let** y **Property Set** relacionadas. Por ejemplo, si se declara este procedimiento **Property Let**:

```
Property Let Names(intX As Integer, intY As Integer, varZ As Variant)
```

```
    ' Instrucciones.
```

```
End Property
```

La declaración de **Property Get** también debe usar argumentos con el mismo nombre y tipos de datos que los del procedimiento **Property Let**:

```
Property Get Names(intX As Integer, intY As Integer) As Variant
```

```
    ' Instrucciones.
```

```
End Property
```

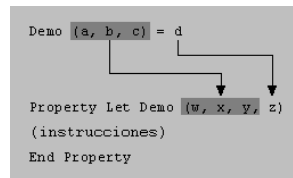
El tipo de datos del último argumento de una declaración **Property Set** debe ser un tipo objeto o **Variant**.

### Llamada a procedimientos Property

La siguiente tabla muestra la sintaxis de la llamada a procedimientos **Property**.

Procedimiento Property	Sintaxis
<b>Property Let</b>	<i>[objeto.]nombreDePropiedad[(argumentos)] = argumento</i>
<b>Property Get</b>	<i>variable = [objeto.]nombreDePropiedad[(argumentos)]</i>
<b>Property Set</b>	<b>Set</b> <i>[objeto.]nombreDePropiedad[(argumentos)] = variable</i>
Quando se llama un procedimiento <b>Property Let</b> o <b>Property Set</b> , siempre aparece un argumento a la derecha del signo igual.	

Quando se declara un procedimiento **Property Let** o un procedimiento **Property Set** con múltiples argumentos, Visual Basic pasa el argumento situado a la derecha de la llamada al último argumento de la declaración **Property Let** o **Property Set**. Por ejemplo, en la figura, Visual Basic pasa el argumento *a* al argumento *w*, *b* al *x*, *c* al *y*, y *d* al *z*.



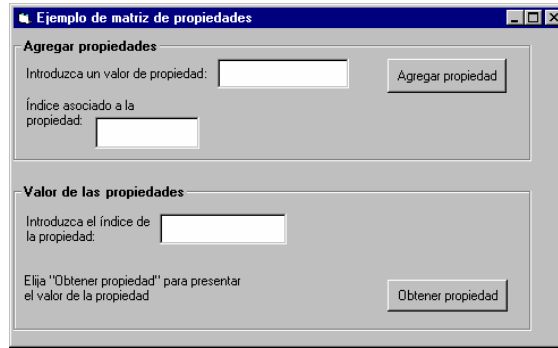
**Figura Llamada a un procedimiento Property**

En la práctica, el único uso de procedimientos **Property** con múltiples argumentos es la creación de matrices de propiedades, como se explica en la siguiente sección.

### Ejemplo de matriz de propiedades

Este ejemplo demuestra como los procedimientos **Property** almacenan propiedades en matrices. Para usar este ejemplo, ejecute la aplicación, escriba un valor en el cuadro de texto y haga clic en el botón “Agregar propiedad”. Después de hacer clic en este botón, se presentará el índice asociado con la nueva propiedad. Para recuperar el valor de una propiedad, escriba el índice de la propiedad y después haga clic en el botón de comando “Obtener prioridad”. Un cuadro de mensaje presenta el valor de la propiedad devuelta.

Dibuje dos marcos, cinco etiquetas, dos botones de comando y dos cuadros de texto en un formulario, como se muestra en la figura.



**Figura Ejemplo de matriz de propiedades**

La siguiente tabla muestra la configuración de propiedades de los objetos de este ejemplo.

Objeto	Propiedad	Valor
Formulario	Caption	Ejemplo de matriz de propiedades
Primer marco	Caption	
Font	Agregar propiedades	
Font Style = Bold		
Font Size = 10		
Segundo marco	Caption	
Font	Valor de propiedades	
Font Style = Bold		
Font Size = 10		
Primer botón de comando	Name	
Caption	cmdAddProp	
Agregar propiedad		
Segundo botón de comando	Name	
Caption	cmdGetprop	
Obtener propiedad		
Primer cuadro de texto	Name	
Text	txtNewValue	
(vacío)		
Segundo cuadro de texto	Name	
Text	txtNewKey	
(vacío)		
Primera etiqueta	Name	
Caption		
BorderStyle	lblIndex	
(vacío)		
Fixed Single		
Segunda etiqueta	Caption	Introduzca un valor de propiedad:

Tercera etiqueta	Caption	Índice asociado a la propiedad:
Cuarta etiqueta	Caption	Introduzca el índice del valor de la propiedad:
Quinta etiqueta	Caption	Elija "Obtener propiedad" para presentar el valor de la propiedad

Declare una matriz **Variant** y un contador de índice entero:

' Esta variable de matriz almacena los valores de la matriz de' propiedades. Es Private, así que la única manera de definir sus valores es mediante Property Let.

```
Private vntDemo() As Variant
```

```
Private intIndex As Integer
```

El evento **Load** del formulario dimensiona una matriz para almacenar los valores de las propiedades:

```
Private Sub Form_Load()
```

```
    ReDim vntDemo(1)
```

```
End Sub
```

El evento **cmdAddProp\_Click** restablece el contador de índice de la matriz y llama al procedimiento **Property Let**:

```
Private Sub cmdAddProp_Click()
```

```
    intIndex = intIndex + 1           ' Restablece el
    contador                          ' de
```

```
    índice.
```

```
    Demo(intIndex) = txtNewValue.Text ' Llama Property Let.
```

```
    lblIndex.Caption = intIndex      ' Presenta el índice.
```

```
    txtNewValue.Text = ""           ' Borra el cuadro de
    texto.
```

```
End Sub
```

El procedimiento **Property Let** asigna a la matriz el valor transferido:

```
Property Let Demo(ByVal intIndex As Integer, vntValue As Variant)
```

```
    If intIndex > UBound(vntDemo) Then
```

```
        ' Redimensiona la matriz.
```

```
        ReDim Preserve vntDemo(intIndex)
```

```
    End If
```

```
    ' Asigna el valor de la propiedad.
```

```
    vntDemo(intIndex) = vntValue
```

```
End Property
```

El procedimiento **Property Get** asigna un valor de propiedad adecuado a la propiedad **Demo**:

```
Property Get Demo(ByVal intIndex As Integer) As Variant
```

```
    If intIndex <= UBound(vntDemo) And intIndex >= LBound(vntDemo) Then
```

```
        Demo = vntDemo(intIndex)           ' Asigna a la  
propiedad.
```

```
    Else
```

```
        Demo = "Error!"
```

```
    End If
```

```
End Property
```

El evento **cmdGetProp\_Click** llama al procedimiento **Property Get** y transfiere el valor de la propiedad al cuadro de mensaje:

```
Private Sub cmdGetProp_Click()
```

```
    MsgBox Demo(CInt(txtNewKey.Text))
```

```
End Sub
```

Una matriz de valores de propiedades también puede implementarse usando el objeto Collection. En muchos casos, esto producirá un código más claro y fácil de mantener.

## **CREACIÓN DE PROCEDIMIENTOS NUEVOS.**

### **Para crear un nuevo procedimiento general**

En el menú **Insertar**, elija **Procedimiento**.

O bien,

Escriba un encabezado de procedimiento en la ventana Código y presione ENTRAR. El encabezado del procedimiento puede ser tan sencillo como **Sub** o **Function** seguido de un nombre. Por ejemplo, puede escribir uno de los siguientes:

Sub UpdateForm ()

Function GetCoord ()

1. Visual Basic responde completando la plantilla del nuevo procedimiento.

## **SELECCIÓN DE PROCEDIMIENTOS EXISTENTES.**

### **Para ver un procedimiento del módulo actual**

Para ver un *procedimiento general* existente, seleccione "(General)" en el cuadro "Objeto" de la ventana Código y después seleccione el procedimiento en el cuadro "Procedimiento".

O bien,

Para ver un *procedimiento de evento*, seleccione el objeto apropiado en el cuadro "Objeto" de la ventana Código y después seleccione el evento en el cuadro "Procedimiento".

### **Para ver un procedimiento de otro módulo**

1. En el menú **Ver**, elija **Examinador de objetos**.
2. Seleccione el proyecto en el cuadro "Bibliotecas/Proyectos".
3. Seleccione el módulo en la lista "Clases/Módulos", y el procedimiento en la lista "Métodos/Propiedades".
4. Elija "Mostrar".

## **LLAMADA A PROCEDIMIENTOS.**

Las técnicas de llamada a procedimientos varían según el tipo de procedimiento, dónde se encuentra y cómo se usa en la aplicación. Las siguientes secciones describen cómo llamar a procedimientos **Sub** y **Function**.

### **Llamada a procedimientos Sub**

Un procedimiento **Sub** difiere de un procedimiento **Function** en que no se le puede llamar usando su nombre en una expresión. Una llamada a **Sub** es una instrucción independiente. Además, un **Sub** no devuelve un valor dentro de un nombre, como lo hace una función. Sin embargo, al igual que un **Function**, un **Sub** puede modificar los valores de cualquier variable que se le transfiera.

Hay dos maneras de llamar a un procedimiento **Sub**:

```
' Estas dos instrucciones llaman a un procedimiento Sub llamado MiProc.
```

```
Call MiProc (primerArgumento, segundoArgumento)
```

```
MiProc primerArgumento, segundoArgumento
```

Observe que cuando usa la sintaxis **Call**, los argumentos deben ir entre paréntesis. Si omite la palabra clave **Call**, también debe omitir los paréntesis que delimitan los argumentos.

### **Llamada a procedimientos Function**

Normalmente, llamará a un procedimiento función que haya escrito personalmente de la misma manera en que se llama a una función intrínseca de Visual Basic como **Abs**, es decir, usando su nombre en una expresión:

```
' Las siguientes instrucciones llaman a una función llamada ToDec.
```

```
Print 10 * ToDec
```

```
X = ToDec
```

```
If ToDec = 10 Then Debug.Print "Fuera de rango"
```

También puede llamar a una función de la misma manera que a un procedimiento **Sub**. Las siguientes instrucciones llaman a la misma función:

```
Call Year(Now)
```

```
Year Now
```



Cuando se llama a una función de esta manera, Visual Basic no utiliza el valor devuelto.

## Llamada a procedimientos de otros módulos

Puede llamar a los procedimientos de otros módulos desde cualquier parte del proyecto. Quizás necesite especificar el módulo que contiene el procedimiento que se va a llamar. Las técnicas de estas llamadas varían dependiendo de si el procedimiento se encuentra en un módulo de formulario, de clase o estándar.

## Procedimientos en formularios

Todas las llamadas desde fuera del módulo de formulario deben indicar el módulo de formulario que contiene el procedimiento. Si un procedimiento llamado **SomeSub** está en un formulario llamado **Form1**, puede llamar al procedimiento de **Form1** usando esta instrucción:

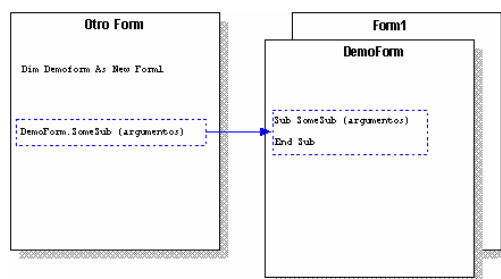
```
Call Form1.SomeSub(argumentos)
```

Para llamar al procedimiento en una instancia (o copia) distinta del formulario, use la variable que indica la instancia para identificar el nombre del procedimiento. El siguiente código llama al procedimiento **SomeSub** contenido en **DemoForm**, que es una instancia de **Form1**.

```
Dim DemoForm As New Form1
```

```
DemoForm.SomeSub(argumentos) ' Donde DemoForm es una instancia de Form1.
```

La figura ilustra cómo se llama al procedimiento **SomeSub**.



**Figura** Llamada a un procedimiento de otro módulo de formulario diferente

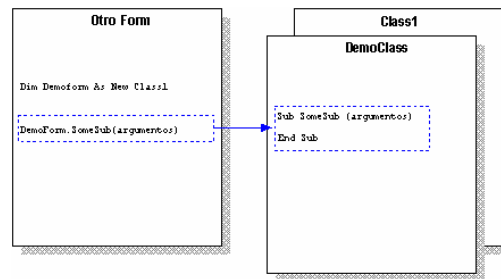
## Procedimientos en módulos de clase

De igual manera que las llamadas a procedimientos en una instancia de un formulario, las llamadas a procedimientos en un módulo de clase requieren que la llamada al procedimiento incluya una variable de instancia de la clase. Por ejemplo, **DemoClass** es una instancia de **Class1**:

```
Dim DemoClass as New Class1
```

```
DemoClass.SomeSub
```

Sin embargo, al contrario que en los formularios, el nombre de clase no puede usarse como cualificador al hacer referencia a una instancia de la clase. El código de la figura llama al procedimiento **SomeSub** en el módulo de clase **DemoClass**.

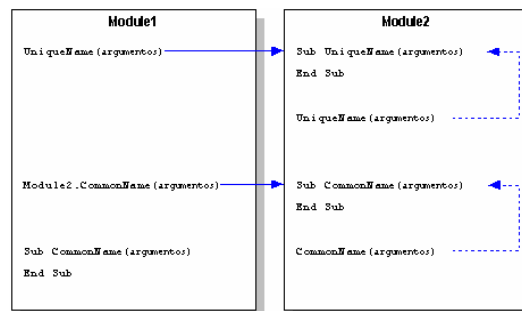


**Figura Llamada a procedimientos en módulos de clase**

**Para obtener más información**

### Procedimientos en módulos estándar

Si el nombre del procedimiento es único, no es necesario incluir el nombre del módulo en la llamada. Una llamada desde dentro o fuera del módulo se referirá a ese único procedimiento. Un procedimiento es *único* si aparece en un solo lugar. Un procedimiento es *común* si aparece en varios lugares, por ejemplo, en dos módulos diferentes. Por ejemplo, en la figura, una llamada a **UniqueName** siempre encontrará el procedimiento.



**Figura Llamada a procedimientos en módulos estándar**

Una llamada a un procedimiento común desde el mismo módulo ejecuta el procedimiento de dicho módulo. Por ejemplo, una llamada a **CommonName**

desde **Module2** ejecutará el procedimiento **CommonName** de **Module2**, no el procedimiento **CommonName** de **Module1**, como se muestra en la figura .

Una llamada a un nombre de procedimiento común desde otro módulo debe especificar el módulo, como se muestra en la figura. Por ejemplo, si quiere llamar al procedimiento **CommonName** de **Module2** desde **Module1**, debe especificar el módulo:

```
Module2.CommonName(argumentos)
```

## **TRANSFERENCIA DE ARGUMENTOS A PROCEDIMIENTOS.**

Normalmente, el código de un procedimiento necesita alguna información sobre el estado del programa para hacer su trabajo. Esta información se transfiere al procedimiento en forma de argumentos cuando se llama a dicho procedimiento.

### **Tipos de datos de argumentos**

Los argumentos de los procedimientos que se escriben tienen el tipo de datos predeterminado **Variant**. Sin embargo, puede declarar otros tipos de datos para los argumentos. Por ejemplo, la siguiente función acepta una cadena y un entero:

```
Function Reverse (S As String, ByVal n As Integer)
```

```
    ' Invierte los n primeros caracteres de S.
```

```
    Dim Temp As String, i As Integer
```

```
    If n > Len(S) Then n = Len(S)
```

```
    For i = n To 1 Step -1
```

```
        Temp = Temp & Mid(S, i, 1)
```

```
    Next
```

```
    Reverse = Temp & Right(S, Len(S) - n)
```

```
End Function
```

### **Transferencia de argumentos por valor**

Cuando una variable se pasa por valor sólo se transfiere una copia de la variable. Si el procedimiento cambia su valor, el cambio afecta sólo a la copia y no a la variable misma. Para indicar que una variable se pasa por valor, se usa la palabra clave **ByVal**.

Por ejemplo:

```
Sub PostAccounts(ByVal intAcctNum as Integer)
```

```
    .
```

```
    . ' Instrucciones.
```

```
    .
```

End Sub

### Transferencia de argumentos por referencia

La transferencia de argumentos por referencia, ofrece al procedimiento el acceso al contenido real de la variable en su posición de memoria. Como resultado, el valor de la variable puede ser modificado permanentemente por el procedimiento al cual se pasa. La transferencia de argumentos por referencia es la predeterminada en Visual Basic.

Si especifica un tipo de datos para un argumento pasado por referencia, debe transferir un valor de ese tipo como argumento. Puede evitar esta limitación usando una expresión, en lugar de tipo de datos, como argumento. Visual Basic evalúa una expresión y, si puede, la pasa como si fuera del tipo requerido. La manera más sencilla de convertir una variable en expresión es incluirla entre paréntesis. Usando la función **Reverse** del ejemplo de “Tipos de datos de argumentos”, el siguiente código convierte una variable en expresión incluyéndola entre paréntesis:

```
Debug.Print Reverse((V), 4) ' Convierte la variable en una expresión.
```

### Uso de argumentos opcionales

Puede especificar que los argumentos de un procedimiento sean opcionales incluyendo la palabra clave **Optional** en la lista de argumentos. Si especifica un argumento opcional, todos los argumentos siguientes de la lista de argumentos también deben ser opcionales, declarándolos con la palabra clave **Optional**. Los argumentos opcionales deben ser del tipo de datos **Variant**. Las dos partes del código de ejemplo siguiente suponen que hay un formulario con un botón de comando y un cuadro de lista.

Por ejemplo, este código proporciona todos los argumentos opcionales:

```
Dim strName As String
```

```
Dim varAddress As Variant
```

```
Sub ListText(Optional x As Variant, Optional y As Variant)
```

```
    List1.AddItem x
```

```
    List1.AddItem y
```

```
End Sub
```

```

Private Sub Command1_Click()
    strName = "SuNombre"

    varAddress = 12345          ' Se proporcionan ambos
argumentos.

    Call ListText(strName, varAddress)
End Sub

```

Este código, sin embargo, no proporciona todos los argumentos opcionales:

```
Dim strName As String
```

```
Dim varAddress As Variant
```

```
Sub ListText(x As String, Optional y As Variant)
```

```
    List1.AddItem x
```

```
    If Not IsMissing(y) Then
```

```
        List1.AddItem y
```

```
    End If
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    strName = "SuNombre"
segundo
```

```
    ' No se proporciona el
' argumento.
```

```
    Call ListText(strName)
```

```
End Sub
```

## Uso de un número indefinido de argumentos

Generalmente, el número de argumentos en una llamada a un procedimiento debe ser el mismo que en la especificación del procedimiento. El uso de la palabra clave **ParamArray** le permite especificar que un procedimiento acepte un número arbitrario de argumentos. Esto le permite escribir funciones como **Sum**:

```
Dim x As Variant
```

```
Dim y As Integer
```

```
Dim intSum As Integer
```

```
Sub Sum(ParamArray intNums())
```

```
    For Each x In intNums
```

```
        y = y + x
```

```
    Next x
```

```
    intSum = y
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    Sum 1, 3, 5, 7, 8
```

```
    List1.AddItem intSum
```

```
End Sub
```

## Creación de instrucciones más sencillas con argumentos con nombre

Visual Basic proporciona la opción de usar *argumentos con nombre* para muchas funciones, instrucciones y métodos incorporados como método abreviado para escribir los valores de numerosos argumentos. Con los argumentos con nombre se pueden proporcionar alguno o todos los argumentos, en cualquier orden, asignando un valor al argumento. Esto se consigue escribiendo dos puntos seguidos de un signo igual (:=) y situando la asignación en cualquier secuencia delimitada por el carácter separador de lista. Observe que los argumentos del siguiente ejemplo están en orden inverso al de la sintaxis real:

```
Set CurDB = OpenDatabase(exclusive:= False, source:= "Fox 2.5", _  
readonly:= False, dbname:="C:\FoxData")
```

También puede usar argumentos con nombre con las funciones y los procedimientos que cree personalmente. Visual Basic asocia automáticamente los nombres de los argumentos con sus correspondientes procedimientos.

```
Function ListText(strName As String, Optional varAddress As Variant)
```

```
    List1.AddItem varAddress
```

```
    List1.AddItem strName
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    ListText varAddress:=12345, strName:="Su nombre"
```

```
End Sub
```

Esto es especialmente útil si sus procedimientos tienen varios argumentos opcionales que no es necesario especificar siempre.

### **Determinación de la aceptación de argumentos con nombre**

Para determinar qué funciones, instrucciones y métodos aceptan argumentos con nombre, compruebe el Examinador de objetos o consulte la Ayuda en pantalla. Cuando trabaje con argumentos con nombre, tenga en cuenta lo siguiente:

Los argumentos con nombre no se aceptan en los métodos de los objetos de la biblioteca de objetos de Visual Basic (VB). Todas las *palabras clave restringidas* del lenguaje de la biblioteca de objetos de Visual Basic para aplicaciones (VBA) y los métodos de la biblioteca de objetos de acceso a datos (DAO) sí los aceptan.

En la sintaxis de la Ayuda en pantalla, los argumentos con nombre se muestran en negrita y cursiva. El resto de los argumentos se muestran sólo en cursiva.

**Importante** No puede utilizar argumentos con nombre para evitar la introducción de los argumentos requeridos. Sólo puede omitir los argumentos opcionales. En las bibliotecas de objetos de Visual Basic 4.0 (VB) y Visual Basic para aplicaciones (VBA), el Examinador de objetos incluye entre corchetes [ ] los argumentos opcionales.



## Para obtener más información

Busque *ByVal*, *ByRef*, *Optional*, *ParamArray* y *argumentos* en la Ayuda

## **ESTRUCTURAS DE CONTROL.**

Las estructuras de control le permiten controlar el flujo de la ejecución de su programa. Si no hay instrucciones de control del flujo que la comprueben, la lógica de un programa recorre las instrucciones de izquierda a derecha y de arriba a abajo. Aunque muchos programas muy sencillos se pueden escribir sólo con ese flujo unidireccional, y aunque parte del flujo puede controlarse usando operadores para regular la preferencia de las operaciones, la mayor parte de la potencia y utilidad de cualquier lenguaje de programación reside en su capacidad para cambiar el orden de las instrucciones con estructuras y bucles.

## **ESTRUCTURAS DE DECISIÓN.**

Al igual que las macros, los procedimientos de Visual Basic pueden comprobar condiciones y después, dependiendo de los resultados de esa comprobación, realizar diferentes operaciones. Las estructuras de decisión aceptadas por Visual Basic incluyen:

**If...Then**

**If...Then...Else**

**Select Case**

### **If...Then**

Use un bloque **If...Then** para ejecutar condicionalmente una o varias instrucciones. Se puede usar una sintaxis de una sola línea o sintaxis de “bloque” de múltiples líneas:

**If** *condición* **Then** *instrucción*

**If** *condición* **Then**

*instrucciones*

**End If**

La *condición* suele ser una comparación, pero puede ser cualquier expresión que se evalúe a un resultado numérico. Visual Basic interpreta este valor como **True** o **False**, un valor numérico cero es **False** y si es distinto de cero se considera **True**. Si la *condición* es **True**, Visual Basic ejecuta todas las instrucciones que siguen a la palabra clave **Then**. Para ejecutar

condicionalmente una sola instrucción puede usar la sintaxis de una sola línea o de múltiples líneas (estos dos ejemplos son equivalentes):

```
If anyDate < Now Then anyDate = Now
```

```
If anyDate < Now Then
```

```
    anyDate = Now
```

```
End If
```

Observe que la forma de línea única de **If...Then** no usa la instrucción **End If**. Cuando la *condición* es **True** y desea ejecutar más de una línea de código, debe usar la sintaxis de bloque de múltiples líneas **If...Then...End If**.

```
If anyDate < Now Then
```

```
    anyDate = Now
```

```
    Timer1.Enabled = False      ' Desactiva el control Timer.
```

```
End If
```

### **If...Then...Else**

Use un bloque **If...Then...Else** para definir varios bloques de instrucciones, de los que sólo uno se ejecutará:

```
If                                condición1                                Then  
    [bloqueDeInstrucciones-1]  
[Elseif                            condición2                            Then  
    [bloqueDeInstrucciones-2]  
    ...  
[Else  
    [bloqueDeInstrucciones-n]  
End If
```

Visual Basic comprueba primero la *condición1*. Si es **False**, Visual Basic procede a comprobar la *condición2*, y así hasta que encuentre una condición **True**. Cuando encuentra una condición **True**, Visual Basic ejecuta el bloque de instrucciones correspondiente y continúa la ejecución a partir de **End If**. Como opción, puede incluir un bloque de instrucciones **Else**, que Visual Basic ejecuta si ninguna de las condiciones es **True**.

**If...Then** no es más que un caso especial de **If...Then...Else**. Observe que puede tener cualquier número de cláusulas **Elseif**, o ninguna. Puede incluir una cláusula **Else** independientemente de las cláusulas **Elseif** presentes.

Por ejemplo, su aplicación podría realizar acciones diferentes dependiendo del control de una matriz de control de menús donde se hace clic:

```
Private Sub mnuCut_Click (Index As Integer)
```

```
    If Index = 0 Then                                ' Comando Cortar.
        CopyActiveControl                            ' Llama a procedimientos
generales.                                         ClearActiveControl
    ElseIf Index = 1 Then                            ' Comando Copiar.
        CopyActiveControl
    ElseIf Index = 2 Then                            ' Comando Borrar.
        ClearActiveControl
    Else                                             ' Comando Pegar.
        PasteActiveControl
    End If
End Sub
```

Fíjese que siempre puede agregar más partes **Elseif** a su estructura **If...Then**. Sin embargo, esta sintaxis puede resultar aburrida de escribir cuando cada **Elseif** compara la misma expresión con un valor diferente. En esta situación, puede usar la estructura de decisión **Select Case**.

### Para obtener más información

Para obtener información adicional sobre **If...Then...Else**, busque *If* en la Ayuda.

### Select Case

Visual Basic proporciona la estructura **Select Case** como alternativa a **If...Then...Elseif** para ejecutar selectivamente un bloque de instrucciones entre varios. Una instrucción **Select Case** proporciona una funcionalidad similar a la instrucción **If...Then...Else**, pero hace que el código sea más eficiente y legible.

La estructura **Select Case** funciona con una única expresión de comparación que se evalúa una sola vez, al principio de la estructura. Visual Basic compara entonces el resultado de esta expresión con los valores de cada **Case** de la

estructura. Si hay una coincidencia, ejecuta el bloque de instrucciones asociado con dicho **Case**:

```

Select                                Case                                expresiónDeComparación
    [Case                                listaDeExpresiones1
        [bloqueDeInstrucciones-1]
    [Case                                listaDeExpresiones2
        [bloqueDeInstrucciones-2]
        .
        .
        .
    [Case                                Else
        [bloqueDeInstrucciones-n]
End Select

```

Cada *listaDeExpresiones* es una lista de uno o más valores. Si hay más de un valor en una lista, los valores se separan por comas. Cada *bloqueDeInstrucciones* contiene cero o más instrucciones. Si más de un **Case** coincide con la expresión de comparación, sólo se ejecuta el bloque de instrucciones asociado con el primer **Case** que coincida. Visual Basic ejecuta las instrucciones de la cláusula **Case Else** (que es opcional) si ninguno de los valores de las listas de expresiones coincide con la expresión de comparación.

Por ejemplo, suponga que agregó otro comando al menú **Edición** del ejemplo de **If...Then...Elseif**. Podría agregar otra cláusula **Elseif**, o bien escribir la función con **Select Case**:

```
Private Sub mnuCut_Click (Index As Integer)
```

```

    Select Case Index
        Case 0
            CopyActiveControl
            ClearActiveControl
            ' Comando Cortar.
            ' Llama a procedimientos
            ' generales.
        Case 1
            CopyActiveControl
            ' Comando Copiar.
        Case 2
            ClearActiveControl
            ' Comando Borrar.
        Case 3
            PasteActiveControl
            ' Comando Pegar.
        Case Else

```

frmFind.Show  
de diálogo Mostrar.

' Muestra el cuadro

End Select

End Sub

Observe que la estructura **Select Case** evalúa la expresión una vez al principio de la estructura. En contraste, la estructura **If...Then...Elseif** puede evaluar una expresión diferente en cada instrucción **Elseif**. Solamente puede reemplazar una estructura **If...Then...Elseif** por una estructura **Select Case** si todas las instrucciones **Elseif** evalúan la misma expresión.

## **ESTRUCTURAS DE BUCLES.**

Las estructuras de bucles le permiten ejecutar repetidamente una o más líneas de código. Las estructuras de bucles que Visual Basic acepta incluyen:

**Do...Loop**

**For...Next**

**For Each...Next**

### **Do...Loop**

Use un bucle **Do** para ejecutar un bloque de instrucciones un número indefinido de veces. Hay diferentes variaciones de la instrucción **Do...Loop**, pero todas evalúan una condición numérica para determinar si debe continuar la ejecución. Como con **If...Then**, la *condición* debe ser un valor o una expresión que se evalúe como **False** (cero) o **True** (distinto de cero).

En el siguiente **Do...Loop**, las *instrucciones* se ejecutan mientras que la *condición* sea **True**:

<b>Do</b>	<b>While</b>	<i>condición</i>
<i>instrucciones</i>		
<b>Loop</b>		

Cuando Visual Basic ejecuta este bucle **Do**, primero comprueba el valor de *condición*. Si *condición* es **False** (cero), salta las instrucciones. Si es **True** (distinto de cero), Visual Basic ejecuta las instrucciones y después vuelve a la instrucción **Do While** y comprueba la condición otra vez.

Consecuentemente, el bucle se puede ejecutar cualquier número de veces, mientras *condición* sea distinto de cero o **True**. Las instrucciones no se ejecutan nunca si *condición* es inicialmente **False**. Por ejemplo, este procedimiento cuenta las veces que aparece una cadena de destino dentro de otra cadena efectuando un bucle hasta encontrar la cadena de destino:

Function CountStrings (longstring, target)

Dim position, count

position = 1

Do While InStr(position, longstring, target)

position = InStr(position, longstring, target) + 1

count = count + 1

Loop

CountStrings = count

End Function

Si la cadena de destino no aparece ninguna vez en la otra cadena, **InStr** devuelve 0, y el bucle no se ejecuta.

Otra variación de la instrucción **Do...Loop** ejecuta primero las instrucciones y después comprueba el valor de *condición* después de cada ejecución. Esta variación garantiza al menos una ejecución de *instrucciones*:

**Do**

*instrucciones*

**Loop While** *condición*

Otras dos variaciones son análogas a las dos anteriores, excepto en que ejecutan el bucle mientras que *condición* sea **False** en lugar de **True**.

**Bucle de cero o más vueltas**      **Bucle de al menos una vuelta**

**Do**

*instrucciones*

**Until**

*condición*

**Loop Do**

*instrucciones*

**Loop Until** *condición*

Observe que **Do Until** *condición* es equivalente a **Do While Not** *condición*.

**For...Next**

Los bucles **Do** son adecuados cuando no se sabe cuántas veces se necesita ejecutar las instrucciones del bucle. Sin embargo, cuando sabe que hay que ejecutar las instrucciones un número específico de veces, el código es más eficiente si se usa un bucle **For**. Al contrario que un bucle **Do**, un bucle **For** usa una variable *contador* que incrementa o reduce su valor en cada repetición del bucle. La sintaxis es:

```

For   contador   =   inicio   To   fin   [Step   incremento]
      instrucciones
Next [contador]

```

Los argumentos *contador*, *inicio*, *fin* e *incremento* son numéricos.

**Nota** El argumento *incremento* puede ser positivo o negativo. Si *incremento* es positivo, *inicio* debe ser menor o igual que *fin* o las instrucciones del bucle no se ejecutarán. Si *incremento* es negativo, *inicio* debe ser mayor o igual que *fin* para que se ejecute el cuerpo del bucle. Si **Step** no se define, el *incremento* toma el valor predeterminado 1.

Al ejecutar el bucle **For**, Visual Basic:

1. Asigna a *contador* el valor *inicio*.
2. Comprueba si *contador* es mayor que *fin*. Si es así, Visual Basic sale del bucle.  
(Si *incremento* es negativo, Visual Basic comprueba si *contador* es menor que *fin*.)
3. Ejecuta las *instrucciones*.
4. Incrementa *contador* en 1 (o en *incremento* si se especifica).
5. Repite los pasos 2 a 4.

Este código imprime los nombres de todas las fuentes de pantalla disponibles:

```

Private Sub Form_Click ()
    Dim i
        For i = 0 To Screen.FontCount - 1
            Print Screen.Fonts(i)
        Next
    End Sub

```

Otro ejemplo, usa un bucle **For...Next** para recorrer las líneas de la propiedad **Selected** de un cuadro de lista de columna múltiple:

```

Private Sub cmdTransfer_Click ()
    For n = 0 To (lstTop.ListCount - 1)
        If lstTop.Selected(n) = True Then           ' Si está
seleccionado,
            lstBottom.AddItem lstTop.List(n)      ' agrega a la
lista.
    End For

```

```
End If
Next
End Sub
```

### **For Each...Next**

Un bucle **For Each...Next** es similar a un bucle **For...Next**, pero repite un grupo de instrucciones para cada elemento de una colección de objetos o de una matriz, en lugar de repetir las instrucciones un número específico de veces. Esto es especialmente útil si no sabe cuántos elementos hay en una colección.

Esta es la sintaxis de **For Each...Next**:

```
For          Each          elemento          In          grupo
    instrucciones
Next elemento
```

Por ejemplo, el siguiente procedimiento **Sub** abre BIBLIO.MDB y agrega el nombre de cada tabla a un cuadro de lista.

```
Dim objDb As Database
Set objDb = OpenDatabase("c:\vb\biblio.mdb", True, False)
For Each TableDef In objDb.TableDefs()
    List1.AddItem TableDef.Name
Next TableDef
```

Tenga presentes las siguientes restricciones cuando utilice **For Each...Next**:

Para las colecciones, *elemento* sólo puede ser una variable **Variant**, una variable **Object** genérica o un objeto enumerado en el Examinador de objetos.

Para las matrices, *elemento* sólo puede ser una variable **Variant**.

No puede usar **For Each...Next** con una matriz de tipos definidos por el usuario, porque **Variant** no puede contener un tipo definido por el usuario.



## **ESTRUCTURAS DE CONTROL ANIDADAS.**

Como demuestra el ejemplo anterior, se pueden colocar estructuras de control dentro de otras estructuras de control (como un bloque **If...Then** dentro de un bucle **For...Next**). A una estructura de control situada dentro de otra estructura de control se le denomina *anidada*.

Las estructuras de control de Visual Basic se pueden anidar tantos niveles como se desee. Es una práctica habitual aumentar la legibilidad de las estructuras de decisión y de bucles anidadas desplazando el cuerpo de la estructura de decisión o bucle.

Por ejemplo, este procedimiento imprime todos los nombres de fuentes que son comunes a la impresora y la pantalla:

```
Private Sub Form_Click()  
  
Dim SFont, PFont  
  
    For Each SFont In Screen.Fonts()  
  
        For Each PFont In Printer.Fonts()  
  
            If SFont = PFont Then  
  
                Print SFont  
  
            End If  
  
        Next PFont  
  
    Next SFont  
  
End Sub
```

Observe que el primer **Next** cierra el bucle **For** interno y el último **Next** cierra el bucle **For** externo. De igual manera, en instrucciones **If** anidadas, las instrucciones **End If** se aplican automáticamente a la instrucción **If** anterior más próxima. Las estructuras anidadas **Do...Loop** funcionan de manera similar, ya que la instrucción **Loop** interna se corresponde con la instrucción **Do** interna.

## **SALIDA DE UNA ESTRUCTURA DE CONTROL.**

La instrucción **Exit** le permite salir directamente de un bucle **For**, un bucle **Do**, un procedimiento **Sub** o un procedimiento **Function**. La sintaxis de la instrucción **Exit** es sencilla: **Exit For** puede aparecer tantas veces como sea necesario dentro de un bucle **For**, y **Exit Do** puede aparecer tantas veces como sea necesario dentro de un bucle **Do**:

```
For    contador    =    inicio    To    fin    [Step    incremento]
    [bloqueDeInstrucciones]
    [Exit                                For]
    [bloqueDeInstrucciones]
Next [contador[, contador] [,...]]
Do    [{While          |          Until}    condición]
    [bloqueDeInstrucciones]
    [Exit                                Do]
    [bloqueDeInstrucciones]
Loop
```

La instrucción **Exit Do** funciona con todas las versiones de la sintaxis del bucle **Do**.

**Exit For** y **Exit Do** son útiles porque algunas veces es apropiado salir de un bucle inmediatamente, sin efectuar más iteraciones o instrucciones del bucle. Por ejemplo, en el ejemplo anterior que imprimía las fuentes comunes a la pantalla y la impresora, el código continúa comparando las fuentes de impresora con una fuente de pantalla específica incluso cuando ya ha habido una coincidencia con una fuente de impresora anterior. Una versión más eficiente de la función sale del bucle tan pronto como se encuentre una coincidencia:

```
Private Sub Form_Click()
```

```
Dim SFont, PFont
```

```
    For Each SFont In Screen.Fonts()
```

```
        For Each PFont In Printer.Fonts()
```

```
            If SFont = PFont Then
```

```
                Print Sfont
```

```
                Exit For ' Salir del bucle
interno.
```

```
            End If
```

Next PFont

Next SFont

End Sub

Como se muestra en este ejemplo, una instrucción **Exit** casi siempre aparece dentro de una instrucción **If** o **Select Case** anidada dentro del bucle.

Cuando se usa la instrucción **Exit** para salir de un bucle, el valor de la variable de control difiere, dependiendo de cómo se salga del bucle:

Cuando se *completa* un bucle, la variable de control contiene el valor del límite superior más el incremento.

Cuando se *sale de un bucle prematuramente*, la variable de control mantiene su valor según las reglas normales.

Cuando se *termina la iteración sobre una colección*, la variable de control contiene **Nothing** si es del tipo de datos **Object**, y contiene **Empty** si es del tipo de datos **Variant**.

### **SALIDA DE UN PROCEDIMIENTO SUB O FUNCTION.**

La sintaxis de **Exit Sub** y **Exit Function** es similar a la de **Exit For** y **Exit Do** de la sección anterior, "Salida de una estructura de control". **Exit Sub** puede aparecer tantas veces como sea necesario, en cualquier parte del cuerpo de un procedimiento **Sub**. **Exit Function** puede aparecer tantas veces como sea necesario, en cualquier parte del cuerpo de un procedimiento **Function**.

**Exit Sub** y **Exit Function** son útiles cuando el procedimiento ha hecho todo lo que tenía que hacer y puede terminar inmediatamente. Por ejemplo, si desea cambiar el ejemplo anterior de manera que imprima sólo la primera fuente común a la impresora y la pantalla que encuentre, usaría **Exit Sub**:

```
Private Sub Form_Click()
```

```
Dim SFont, PFont
```

```
For Each SFont In Screen.Fonts()
```

```
For Each PFont In Printer.Fonts()
```

```
    If SFont = PFont Then
```

```
        Print Sfont
```

```
Exit Sub ' Sale del procedimiento.
```

```
End If
```

```
Next PFont
```

```
Next SFont
```

```
End Sub
```

## COMBINACIÓN DE INSTRUCCIONES EN UNA LÍNEA.

Normalmente, en Visual Basic hay una instrucción por línea, y no hay un indicador de finalización de instrucción. Sin embargo, se pueden colocar dos o más instrucciones en una línea si se usan dos puntos (:) para separarlas:

```
Text1.Text = "Hola" : Red = 255 : Text1.BackColor = Red
```

## ***DIVISIÓN DE UNA INSTRUCCIÓN EN MÚLTIPLES LÍNEAS.***

Puede dividir una instrucción larga en varias líneas en la ventana Código usando el *carácter de continuación de línea* (un espacio seguido de un subrayado). Un carácter de continuación de línea no puede ir seguido de un comentario en la misma línea. El uso del carácter de continuación de línea puede ayudar a aumentar la legibilidad del código, tanto en pantalla como impreso. El siguiente código se divide en dos líneas con el carácter de continuación de línea ( \_):

```
Data1.RecordSource = "SELECT * FROM Titles, Publishers" _
```

```
& "WHERE Publishers.PubId = Titles.PubID AND Publishers.State = 'CA'"
```

## ***ADICIÓN DE COMENTARIOS AL CÓDIGO.***

Al leer los ejemplos de esta guía, a menudo se encuentra el símbolo de comentario ('). Este símbolo indica a Visual Basic que no debe hacer nada con las palabras que lo siguen. Dichas palabras son comentarios situados en el código en beneficio del programador, y otros usuarios que puedan examinar el código en el futuro. Por ejemplo:

```
' Este comentario comienza en el borde izquierdo de la pantalla.
```

```
Text1.Text = "Hola" ' Coloca un saludo en el cuadro de texto.
```

Los comentarios pueden seguir a una instrucción, o pueden ocupar una línea entera. Los dos casos se muestran en el código anterior. Los comentarios no pueden seguir a un carácter de continuación de línea dentro de una línea.

## **COMPRESIÓN DE SISTEMAS DE NUMERACIÓN.**

La mayoría de los números de este manual son decimales (base 10). Pero ocasionalmente es conveniente usar números hexadecimales (base 16). Visual Basic representa los números en hexadecimal con el prefijo &H y en octal (base 8) con &O. La siguiente tabla muestra los mismos números en decimal, octal y hexadecimal.

<b>Decimal</b>	<b>Octal</b>	<b>Hexadecimal</b>
9	&O11	&H9
15	&O17	&HF
16	&O20	&H10
20	&O24	&H14
255	&O377	&HFF

Generalmente, no es necesario aprender el sistema de numeración hexadecimal ni octal, porque su PC puede trabajar con números introducidos en cualquier sistema de numeración. Sin embargo, algunos sistemas de numeración son más apropiados para ciertas tareas, como la definición de colores de pantalla y de controles.

**Nota** Si desea usar un número hexadecimal u octal en situaciones en las que se manejan números grandes, agregue un signo (&) al número. Esto hace que el valor se almacene correctamente como entero **Long**.

## **CONVENCIÓN DE NOMBRES EN VISUAL BASIC.**

Al escribir código Visual Basic, se declaran y asignan nombres a muchos elementos (procedimientos **Sub** y **Function**, variables y constantes, etc.). Los nombres de los procedimientos, variables y constantes que se declaran en código Visual Basic deben seguir estas reglas:

- Deben empezar por una letra.
- No pueden contener puntos ni caracteres de declaración de tipos.
- Los nombres de variables, constantes y procedimientos no pueden tener una longitud mayor que aproximadamente 200 caracteres. Los nombres de controles, formularios, clases y módulos no deben superar los 40 caracteres.

- Los nombres de los elementos declarados en el código no pueden ser palabras clave restringidas.

Una *palabra clave* es una palabra que Visual Basic usa como parte de su lenguaje. Esto incluye instrucciones predefinidas (como **If** y **Loop**), funciones (como **Len** y **Abs**) y operadores (como **Or** y **Mod**).

### Para obtener más información

Para obtener la lista completa de las palabras clave restringidas, busque *palabras clave restringidas* en la Ayuda.

Los formularios y los controles pueden tener el nombre de una palabra clave. Por ejemplo, puede tener un control llamado Loop. Sin embargo, en el código no puede hacer referencia a dicho control de la manera habitual, porque Visual Basic asume que es la palabra clave **Loop**. Por ejemplo, este código provoca un error:

```
Loop.Visible = True ' Provoca un error.
```

Para hacer referencia a un formulario o control que tiene el nombre de una palabra clave, debe incluirlo entre corchetes: [ ]. Por ejemplo, este código no provoca error:

```
[Loop].Visible = True ' Con corchetes funciona.
```

- Puede usar corchetes de esta manera para referirse a formularios y controles que ha nombrado con una palabra clave, pero no para declarar una variable ni definir un procedimiento con el mismo nombre que una palabra clave.
- Los corchetes también se pueden usar para obligar a Visual Basic a aceptar nombres proporcionados por otras bibliotecas de tipos que infringen las reglas normales.

**Nota** Puesto que escribir corchetes puede ser tedioso, existe la posibilidad de restringir el uso de *palabras clave restringidas* como nombres de formularios y controles. Sin embargo, la técnica de los corchetes se puede usar si en futuras versiones de Visual Basic se definen nuevas *palabras clave restringidas* que provoquen conflictos con nombres de formularios o controles existentes al actualizar el código para que funcione con la nueva versión.

# REDES Y COMUNICACIONES ELECTRONICAS<sup>1</sup>

## LAS REDES Y SUS ORIGENES

### ANTECEDENTES

Una Red es un conjunto de computadoras independientes (hosts) capaces de comunicarse electrónicamente. Los orígenes de las redes de computadoras se remontan a los primeros sistemas de tiempo compartido, al principio de los años sesenta, cuando una computadora era un recurso caro y escaso.

La idea que encierra el tiempo compartido es simple. Puesto que muchas tareas requieren solo una pequeña fracción de la capacidad de una gran computadora, se saca mayor rendimiento de esta, si presta servicios a más de un usuario al mismo tiempo. Del tiempo compartido a las redes hay solo un pequeño escalón.

Una vez demostrado que un grupo de usuarios más o menos reducido podía compartir una misma computadora, era natural preguntarse si muchas personas muy distantes podrían compartir los recursos disponibles (discos, terminales, impresoras, e incluso programas especializados y bases de datos) en sus respectivas computadoras de tiempo compartido.

Posteriormente de estos servicios saldrían redes de datos públicas como Tymnet y Telenet. Las redes de las grandes corporaciones (Xerox, General Motors, IBM, Digital Equipment Corporation, AT&T y Burroughs), y las redes de investigación (SERCnet y NPL, inglesas de 1966-1968; HMI-NET de Berlín 1974; CYCLADES, Francia 1972), las redes comerciales, los sistemas de conferencia y las comunidades virtuales (especialmente USENET y FIDOnet).

A medida que las redes de computadoras fueron captando más adeptos, compañías tales como XEROX e IBM comenzaron a desarrollar su propia tecnología en redes de computadoras, comenzando por lo general, con redes de área local. Las redes de amplio alcance entonces, pasaron a ser usadas no solo para la comunicación entre computadoras conectadas directamente sino también para comunicar las redes de área local.

Con el establecimiento de ARPAnet, en U.S.A.-1968, comenzó a entremezclarse el impacto social de la telemática. La tecnología de ARPAnet fue utilizada para construir en 1976, la red comercial TELENET. En Europa, las compañías de teléfono, que controlan las redes públicas de transmisión de datos en cada país, adoptaron el protocolo X-25.

En 1987 la red ARPAnet -dependiente del departamento de Defensa norteamericano- utilizada al principio, exclusivamente para la investigación y desbordada por el interés demostrado por sus usuarios por el correo electrónico, necesitó transmitir datos que usaban gran espectro de banda (sonidos, imágenes y videos) y sufrió tal congestión que tuvo que declarar obsoletas sus redes de transmisión de 56.000 baudios por segundo

---

Amenta, Gabriela y Navarro, Gustavo. Manual del Usuario Redes y Comunicaciones Electrónicas. Consejo Latinoamericano de Ciencias Sociales - CLACSO  
<http://lanic.utexas.edu/la/region/networking/clacsoman.html><sup>1</sup>

(5.000 palabras por minuto). Posteriormente se convirtió en la espina dorsal de las telecomunicaciones en U.S.A. bajo su forma actual de INTERNET, una vez que quedó demostrada la viabilidad de redes de paquetes conmutados de alta velocidad.

Los servicios comerciales que concentraron una cantidad de bases de datos como DIALOG, empezaron alrededor de 1972. Los sistemas de conferencia computarizada comenzaron en 1976 y posteriormente encontraron viabilidad comercial en servicios centralizados como Delphi así como en sistemas algo más distribuidos como CompuServe.

Mientras tanto, se fue desarrollando otra tecnología, basada en conexiones por líneas telefónicas en lugar de conexiones dedicadas. Dos de los primeros productos de esta tecnología fueron ACSNET y UUCP, que sobreviven en una forma modificada. Las redes a través de líneas telefónicas produjeron el más distribuido de los sistemas de conferencia: USENET. También BITNET puso a disposición de la comunidad académica la tecnología en redes de computadoras de IBM y lo difundió aun entre computadoras de otras marcas.

Los servicios prestados por las redes de computadoras se han difundido ampliamente y alcanzan ya a la mayoría en las naciones. A medida que su diversidad continúa en aumento, la mayoría de las redes académicas, se conectan entre sí, por lo menos con el propósito de intercambiar correo electrónico.

La comunicación mediante computadoras es una tecnología que facilita el acceso a la información científica y técnica a partir de recursos informáticos y de telecomunicaciones. Por eso, decimos que una red es, fundamentalmente, una forma de trabajo en común, en la que son esenciales tanto la colaboración de cada miembro en tareas concretas, como un buen nivel de comunicación que permita que la información circule con fluidez y que pueda llevarse a cabo el intercambio de experiencias.

## CLASIFICACION DE REDES E INTERFASES

Podemos clasificar las redes en tres tipos, según el área geográfica que abarquen. Estas son:

LAN - Red de Área Local (Local Area Network) Red formada por computadoras que se encuentran en un mismo edificio, fábrica o campus universitario, es decir en un radio de unos pocos kilómetros cuadrados.

MAN -Red de Área Metropolitana (Metropolitan Area Network)

Red que abarca el área de una ciudad.

WAN -Red de Área Amplia (World Area Network) Red que abarca países enteros y hasta todo el mundo.

Como hemos mencionado anteriormente, en la actualidad llamamos RED a un conjunto de computadoras interconectadas entre sí. La INTERNET, por ejemplo, es una red de redes. Toda la tecnología actual de internet-working se basa en la interconexión de redes rápidas. Las redes rápidas son las redes locales o redes de área local (en inglés LAN). Dentro de las LANs, hay varios tipos. Las más comunes son "Ethernet" y "Token-ring".



Las redes "Ethernet" se basan en una "topología de bus" donde todas las interfases conectadas al cable "escuchan" lo que otras interfases "escriben". Cuando una interfase desea enviar un dato a través de la red, simplemente comienza a transmitir.

Las redes "token-ring" se basan en una "topología de anillo", donde los datos circulan a través del anillo, de interfase en interfase. En esta topología, las interfases no transmiten cuando lo desean, sino que esperan a recibir el token. Este token, va circulando por todas las interfases dándole a cada una la posibilidad de transmitir.

Para acceder a las redes existen, básicamente, dos formas de acceso tecnológicas distintas :

"Lineas dedicadas": Estas líneas tienen un costo fijo mensual y sobre ellas se puede transmitir sincrónica o asincrónicamente, analógica o digitalmente, con velocidades variando entre 300 baudios y alguna cantidad (muy alta) de Mbits/seg, dependiendo del tipo de material que se use como transmisor, así como de la forma de transmisión. Por ejemplo, en una línea dedicada analógica sobre cable de cobre común (utilizado por el sistema telefónico) y transmitiendo analógicamente, se pueden obtener 19200 Kbits/seg, mientras que la misma línea, transmitiendo en modo digital puede alcanzar los 128 Kbits/seg

"Redes públicas": La conexión a estas redes se hace, en general, a través de líneas dedicadas, aunque en algunos casos se puede acceder vía el sistema telefónico conmutado. La diferencia con el caso anterior es que la línea dedicada conecta al usuario con el proveedor del servicio. El proveedor del servicio opera una WAN que interconecta a muchos usuarios entre sí, pero en una WAN, las conexiones son virtuales (comúnmente denominados circuitos virtuales). Esto significa que por la misma línea, una interfase puede mantener varias conexiones simultáneamente.

## REDES EN AMERICA LATINA

En los últimos años diversas instituciones han manifestado su interés por crear e integrarse a redes de comunicación. Numerosas redes funcionan con éxito y han sido fundamentales para las diversas áreas del conocimiento y programas de cooperación en la medida que la automatización de los datos permite a investigadores y profesionales tener una visión más amplia de la producción en los más variados sectores.

Por otra parte, cabe señalar que a pesar de un desarrollo tecnológico acelerado, la práctica de comunicación por redes en nuestros países es aún incipiente, sin embargo las que están interconectadas y que intercambian correo electrónico y/o "noticias" como CCC (Centro de Comunicación Científica de la Universidad de Buenos Aires) y CLACSO en Argentina están creciendo con la aparición de INTERNET.

BITNET, FIDONet, UUCP, e INTERNET son las redes mundiales más grandes y de estas, INTERNET es la mayor en cuanto a su crecimiento, su alcance y el volumen de computadoras conectadas, es una de las redes mayor recomendada para mantener tráfico de transferencia de grandes archivos.

Sin embargo, desde cualquiera de estas redes es posible comunicarse con cualquiera de los aprox. 5 a 30 millones de usuarios activos en el mundo de las comunicaciones electronicas a traves de correo electronico.

## APLICACION DE LAS REDES AL TRABAJO

La forma en que las redes son usadas ha estado cambiando y han afectado la forma de trabajo, incluso a los academicos. El antiguo modelo de una gran computadora, centralizada, ya es cosa del pasado. Ahora la mayoría de las instalaciones tienen diferentes tipos de computadoras, desde computadoras personales y estaciones de trabajo, a super computadoras.

Las computadoras, por lo general, estan configuradas para realizar tareas particulares. Aunque la gente suele trabajar con una computadora especifica, las computadoras pueden llamar a otros sistemas en la red para servicios especializados. Esto ha dado origen al modelo de servicios de red "SERVIDOR-CLIENTE". El servidor y el cliente no tienen, necesariamente, que estar en distintas computadoras, podrian usar distintos programas en la misma computadora.

El trabajo a distancia entre instituciones y personas muy diversas, separadas geograficamente como es el caso de CLACSO, ha recibido un gran impulso gracias a la introduccion del fax y del correo electronico. Ello esta acelerando el ritmo del intercambio a tal punto que podemos plantearnos acciones concretas e investigaciones de todo tipo coordinadas a distancia. Tal como lo senalo A. Toffler: "lo que esta cambiando el equilibrio del poder en el mundo es la combinacion de nuevas tecnologias de comunicacion cada vez mas accesibles (computadoras, telefonos, modems, satelites), que se traducen en autenticas "autopistas electronicas".

Las nueva tecnologias permiten trabajar sin salir de nuestras casas. El teletrabajo ha dejado de ser un mito lejano. Ocho millones de tele-trabajadores europeos y veinticinco en Estados Unidos son los primeros tecnomadas del ciberespacio. No importa el lugar de residencia, los tecnomadas asumen su condicion de pioneros. Las telecomunicaciones les permiten adquirir el don de la ubicuidad.

La revolucion del teletrabajo no ha hecho mas que empezar, como muy bien sugiere Dennis Ettinghoffer en su libro La empresa virtual: "El hombre contemporaneo entra en el siglo XXI con la perspectiva de ver como se modifica su relacion con las cosas, con su trabajo, con su empresa y con los otros. Esta en curso una formidable mutacion en nuestra evolucion". Recientes estudios confirman que la computadora modifica el lenguaje de las personas que lo emplean en su actividad productiva, "delante del monitor, la gente tiende a ser mas desinhibida y espontanea", dice Lee Sproul, profesora de sociologia de la Universidad de Boston.

No cabe duda de que la autonomia que ofrece esta nueva forma de trabajo podria servir para mejorar las relaciones familiares, ampliar el tiempo libre, cuidar mejor la imagen individual y, sobre todo, mejorar la productividad al racionalizar el trabajo. Ademas, es una forma mucho mas ecologica de dedicarse al trabajo cotidiano. Sin embargo, tambien cuenta con sus desventajas: no es demasiado economico comprarse una computadora, un modem y un fax. Por otra parte, se sufre una

mayor tendencia al aislamiento y una menor integración en la empresa, lo que redundaría en la dificultad de controlar el trabajo.

Pero la posibilidad de trabajar en la propia casa, sin largos desplazamientos o madrugones, como vaticinaba Toffler en su obra *La tercera ola*, parece bastante tentadora. Además, ya está a la vuelta de la esquina.

## REDES Y CIENCIAS SOCIALES

La telemática (telefonía + computación) está produciendo transformaciones profundas tanto en las formas de realizar investigación cuanto en el proceso de conversión de la información de datos en bruto a los registros interpretativos y la difusión de los resultados. Existe un potencial en algunos usos de las telecomunicaciones y teleinformática que puede cambiar en forma radical el modo de hacer investigación en las ciencias sociales.

Estas innovaciones se plantean a distintos niveles y son una consecuencia directa del teletrabajo antes mencionado. En un primer momento permitirán que una cantidad importante de investigadores interactúen frecuentemente unos con otros. Al mismo tiempo permiten que gran cantidad de investigadores dispersos y situados en sitios de difícil acceso en la región, y a los cuales les está vedado el consumo de información primaria, puedan mantener el contacto con la palabra impresa a costos accesibles.

El aspecto más interesante para un proyecto de estas características consiste en la coordinación en tiempo real de una cantidad significativa de investigaciones simultáneas tanto a nivel regional como inter-continental.

Por otro lado, dada la necesidad de alta tecnología de los países latinoamericanos, una forma de recuperar parte del camino perdido consistiría en la repatriación de científicos latinoamericanos residentes en el exterior. Los ejemplos conocidos de programas de este tipo han sido emperados y fracasados. Una forma alternativa de esta repatriación física es el contacto electrónico permanente con los mismos.

Cada día miles de personas se anotan en la nueva moda de las autopistas de la comunicación. A medida que avanza el tiempo, el estar conectado será entonces una verdadera necesidad, cualquiera que no lo haga quedará definitivamente aislado del mundo. El mundo, cada vez más interrelacionado, está cambiando a gran velocidad. Junto con ello también lo hacen las formas de trabajo de los investigadores. El aislamiento personal o institucional no ayudan al fortalecimiento de las ciencias sociales en este nuevo contexto mundial.

El modelo clásico de procesamiento de la información (emisor-mensaje-receptor) que ha guiado durante décadas gran parte de la institucionalización y comunicación de los resultados de la investigación, está siendo reformulado aceleradamente.

El "tratamiento" de la información -incluyendo el procesamiento visual tan poco atendido en la literatura académica- está siendo abordado desde nuevas perspectivas, teniendo en cuenta conceptos como el de "conversación multidireccional" que hasta hace poco no estaban demasiado difundidos ni eran técnicamente factibles.

Las posibilidades de acumular y recuperar cantidades importantes de información y de compartirla con usuarios a larga distancia, permite imaginar escenarios de

"dialogo de alta precision", que den lugar a nuevas redes de comunicacion mas rapidas y eficientes. Por ejemplo, el correo electronico nos permite, aunque estemos muy alejados geograficamente, intercambiar mensajes e informacion almacenada en computadoras por medio de una conexion a redes telefonicas. Las nuevas tecnologias informaticas y en telecomunicaciones pueden transformarse en herramientas utiles de trabajo para los cientificos sociales de la region. Es mucho lo que se puede ganar al aprender a usarlas.

La comunidad de redes electronicas ha estado creciendo en los ultimos cinco anos a un ritmo constante. Las ciencias sociales de America Latina no pueden "padecer" pasivamente este fenomeno. CLACSO se propone estar presente y ser protagonista en este nuevo modo de produccion; de ahi que el uso de nuevas tecnologias de la comunicacion constituya un elemento importante de la renovacion institucional. El Plan de Trabajo 1992-1995 elaborado por CLACSO, contemplo partir de los puntos focales de comunicacion establecidos en los ultimos anos y ofrecer adiestramiento a personal de los centros para agilizar nuestra comunicacion interna y estimular la colaboracion y el intercambio entre investigadores.

Las comunicaciones electronicas aceleraron tambien la atencion de consultas entre regiones como dentro de la region. Esto permitio avanzar un gran paso mas en la integracion regional e internacional de las ciencias sociales. Aun queda mucho por hacer, pero la cuestion, entonces, es organizarnos para que compartamos y hagamos uso de estas tecnologias en forma cotidiana. En America Latina el area de la comunicacion internacional de datos en Ciencias Sociales se encuentra en un nivel incipiente y el exito de su implementacion depende mucho de la coordinacion internacional y la concertacion entre todas las partes involucradas.

Actualmente, se estan llevando a cabo varios esfuerzos importantes en la region, por ejemplo, las Redes de Organismos no Gubernamentales (APC-Alternex) en Rio de Janeiro y otras redes que estan surgiendo a diario.

La iniciativa de CLACSO demuestra que el mundo de las redes electronicas es enorme y complejo y que con el uso de nuevas tecnologias de computacion se beneficiara una mejor cooperacion e intercambio de informaciones entre los paises latinoamericanos y las organizaciones internacionales.

La integracion con las redes academicas existentes facilitan el acceso para los expertos y centros de excelencia, las organizaciones no gubernamentales, el sector academico, las companias publicas de telecomunicacion y el sector privado. A fin de comprender la complejidad del campo de las redes en la region, podemos clasificar a los paises de America Latina y el Caribe, segun su nivel de conectividad a Internet, el mas sofisticado en los casos de comunicacion en redes. Estos son:

- a) Paises sin conectividad. (Guyana, Surinan, Haiti y otros pocos)
- b) Paises donde existe una red publica (redes con protocolos X-25 de conmutacion de paquetes), pero que por lo general es muy cara para un uso extensivo de la comunidad academica y de investigacion (Guatemala, Honduras, El Salvador y la mayoria de paises del Caribe).
- c) Paises en los que existe un nivel basico de conectividad. Generalmente una o mas estaciones conectadas a Internet usando UUCP (el protocolo de copia de

Unix a Unix) sobre líneas telefónicas estándar (Bolivia, Paraguay, Uruguay, Nicaragua, Perú, etc).

d) Países con enlaces satelitales dedicados a Internet (Chile, Argentina, Venezuela, Ecuador, Costa Rica, etc.).

## COMUNICACIONES ELECTRONICAS

Este manual describe, brevemente, que son las comunicaciones electrónicas, que tipo de sistemas existen actualmente y que servicios ofrecen las Redes Electrónicas de información y comunicación. No es una lista completa de todas las posibilidades, tampoco se explicarán todos los comandos de los sistemas ni abarcarán aun todas las áreas posibles de trabajo con las comunicaciones electrónicas. Una completa revisión y explicación de todas las posibilidades requiere un gran libro más que un útil manual. No obstante, este manual es algo más que una idea sobre los sistemas de comunicación electrónica, constituye una guía sobre lo básico del trabajo en Red, sobre sus alcances, sus posibilidades, sobre la forma de trabajar con una Red Electrónica de comunicación e información y de cómo beneficiarse.

Las redes mundiales de computadoras que intercambian comunicación electrónica proveen un campo fértil para compartir recursos y cooperación. La mayoría de las redes están creciendo exponencialmente, con tasas cerca del 100% anual. Las redes de computadoras se han expandido y continúan su expansión por todo el mundo y en áreas muy diversas, por ejemplo las redes de hospitales en Argentina. Los usos de estas redes pueden agruparse en dos propósitos: "compartir recursos y comunicación mediante computadoras (CMC)".

Los datos a continuación, muestran la cantidad de computadoras existentes en algunas Redes:

BITNET 3,477

FIDONet 16,303

UUCP 14,805

INTERNET 992,000

IP Privado 600,000

Total 1,612,000

Para trabajar en Red, los sistemas de comunicación electrónica ofrecen diferentes posibilidades, las más utilizadas son: el Correo Electrónico (e.mail), las Conferencias Electrónicas, los Bulletin Boards Systems (BBS) y las Listas de Interés. A continuación, Ud. encontrará una breve descripción de las utilidades de cada sistema de comunicación y sus posibilidades.

## CORREO ELECTRONICO (E.MAIL)

El correo electrónico (e.mail: electronic mail) es una técnica de comunicación en donde Ud. puede intercambiar mensajes entre dos computadoras situadas en cualquier lugar del mundo. El correo electrónico es rápido, conveniente, versátil y usualmente económico, es por eso que los usuarios de las comunicaciones electrónicas usan, en su mayoría, el envío y la recepción de mensajes, es decir, el correo electrónico.

No se necesita saber demasiado ni acumular gran información sobre computadoras para poder usar el correo electrónico; tampoco se necesita conocer demasiado sobre teléfonos, conexiones, uso electrónico de los teléfonos y las líneas. De hecho, los temas que se discuten vía correo electrónico no son necesariamente sobre las computadoras o aspectos técnicos de estas. En el correo electrónico Ud. podrá discutir, hablar o intercambiar lo mismo que si estuviera hablando por teléfono o a través de una carta escrita.

Ud. se preguntara, como hacen los mensajes que yo escribo en una computadora para llegar a otra computadora en otro país? Si lo pensamos como si fuese una carta escrita, sabemos que nosotros despachamos en el buzón un sobre cerrado, este es sellado y el correo lo envía, ya sea por avión, tren, etc. a otra casilla de correo o dirección.

Pero, que sucede con los mensajes electrónicos, cual es la forma y como los mensajes son transportados en un sistema tan complejo como las redes de comunicación electrónica? El "nombre" corresponde a un casillero o una máquina; la "dirección" le indica al programa de computadora la ubicación de ese recurso y la "ruta" es lo que el programa utiliza para llegar a ese punto específico de la red. Puede haber más de una ruta para llegar a un mismo destino, y en algunos casos, la dirección variara dependiendo de la posición desde la cual se la quiera alcanzar. Es por eso, que para el uso personal del correo electrónico, lo primero que Ud. debe saber es si la computadora en la cual trabajara esta conectada a una Red Electrónica. Si se obtiene el permiso de trabajar en Red, para ello necesitara un "casillero" propio, es decir, una casilla postal electrónica.

A través del correo electrónico los mensajes que Ud. envía a otro usuario serán almacenados en un casillero, llamado "mailbox" (casilla postal), el cual será consultado por el usuario al que se le ha enviado el mensaje. Es lo mismo que Ud. escriba una carta a una casilla postal. Luego este mensaje recibido será leído por el destinatario el cual podrá responderle y enviarle de nuevo un mensaje que será depositado en su "mailbox", Ud. abra luego su mailbox y podrá leer el mensaje recibido y responderlo, y así sucesivamente.

En cuanto a la rapidez con la que llegan los mensajes, en la mayoría de los casos, el mensaje electrónico puede transmitirse en segundos o minutos. El mensaje que Ud. envía viaja entre dos computadoras y el tiempo depende de la conexión, ya sea, enviando el mensaje a alguien en su propia organización o institución o enviando el mensaje a otra institución u organización. Es poco probable que el mensaje tarde más de un día en llegar.

Como dijimos anteriormente, cuando Ud. conoce a alguien que posee dirección electrónica, enviarle un mensaje implica la ejecución de unos simples comandos que dependerán del sistema instalado en su computadora. Ud. podrá enviarle mensajes, documentos, etc. y también podrá responder automáticamente sobre el mensaje recibido, ya sea intercalando entre líneas o al final de mensaje. También muchos de los servicios de correo electrónico permiten enviar un mensaje a varios usuarios a la vez, como así también copias de mensajes a otros usuarios.

## CONFERENCIAS ELECTRONICAS

Cuando un grupo de personas con intereses comunes o afines comienzan un intercambio de mensajes, pueden hacerlo a través de una "CONFERENCIA", es decir, enviando mensajes al conjunto del grupo. Una conferencia es una estructura utilizada para mantener una discusión o un espacio para el intercambio de información entre usuarios que participan desde diferentes lugares y en diferentes momentos o tiempos.

Hay muchas conferencias electrónicas sobre temas muy variados y Ud. puede participar de ellas enviando mensajes de la misma manera que si lo hiciese a una sola persona, la dirección electrónica en este caso es la de la conferencia y no un mailbox privado. Ud. debe suscribirse a los servicios de la conferencia, solicitando individualmente su incorporación, y a partir de su inscripción, Ud. como usuario de dicha conferencia recibirá todos los mensajes de la conferencia y podrá enviar sus comentarios o mensajes a todos los participantes.

Existen ciertas condiciones que serán delimitadas al comienzo de la conferencia y a través de estas se conformarán sus diferentes características. Por ejemplo, puede ser local, si existe apenas en un nodo de la Red, o en Red si es reproducida en más de un nodo de la Red. También puede ser pública, es decir, abierta a todos los usuarios del sistema, ya sea para participar activamente o solo para su lectura, o privada, significa que solo un grupo limitado de usuarios acceden y participan de ella.

Las conferencias son utilizadas para facilitar la comunicación en grupo, sirviendo de espacios electrónicos para discusiones públicas o privadas entre grupos de usuarios. Siempre que sea necesario compartir información con un grupo de usuarios en forma regular, se debe optar por una conferencia en vez del correo electrónico. La información en una conferencia puede ser estructurada, por ejemplo, para facilitar los procesos de toma de decisión en grupo, para modificaciones en un proyecto conjunto, etc.

Es por esto que todas las conferencias son organizadas en tópicos de discusión, textos enviados por los usuarios para que sean leídos por otros usuarios y comentados. Cada tópico puede tener muchas respuestas y a su vez, cada conferencia puede contener muchos tópicos. Por ejemplo, una conferencia electrónica puede ser organizada de modo que solamente algunos usuarios pueden abrir nuevos tópicos de discusión, mientras otros pueden, solamente, enviar respuestas a estos tópicos.

## BULLETIN BOARDS SYSTEMS (BBS)

Los Bulletin Boards Systems (BBS), también son sistemas que permiten el intercambio de información electrónica. Trabajan en un computador brindando servicios de mensajería, como ser el correo y conferencias electrónicas, listas de interés e intercambio de archivos. El BBS tiene su razón de ser en el "post" de información (sistema para envío de mensajes electrónicos organizados por temas de interés), donde los comentarios son tratados como nuevos ítems de información y pueden ser vistos por cualquiera que visita el BBS. Una password (clave secreta) es requerida para acceder al BBS.

En forma simple, la idea del BBS es que cualquier usuario del sistema de correo electrónico puede enviar mensajes a un área pública y también recibirlos. En los

BBS es posible que los usuarios agreguen comentarios a artículos enviados por otros usuarios al BBS. Todas las redes de correo electrónico contienen algunos métodos para identificar cuáles BBS tienen nuevos ítems adicionados desde la última vez que algún usuario ingreso información. En este caso no es necesario leer todos los mensajes del BBS ya que solo se mostrarán los nuevos.

Hay muchas variantes dentro de los BBS y dependen de cada sistema. El derecho de enviar mensajes a un BBS puede, por ejemplo ser restrictiva a organismos específicos o individuos (solo algunos pueden leer los índices). En este caso, es similar a las conferencias electrónicas, cuando se inicia un BBS, se determinan las condiciones de trabajo dentro del BBS y por consiguiente sus características específicas. Por ejemplo, ciertos BBS son establecidos por algunas organizaciones para el uso, únicamente, de sus miembros y puede ser muy útil para desarrollar y realizar debates internos.

Los sistemas de correo electrónico contienen una variedad de BBS divididos por regiones geográficas o temas de interés. Por ejemplo BBS sobre América Latina, Ciencia, Sexualidad, etc. Otros sistemas de correo electrónico, por ejemplo en APC (Asociación para el Progreso de las Comunicaciones) son conocidos como conferencias, el principio es exactamente igual al BBS, pero su operación es diferente. En las redes APC, los sistemas de conferencia están divididos en diferentes tópicos y cualquier tópico puede tener una serie de comentarios que son generados por los participantes en la discusión. Estos comentarios no son visibles a menos que el tópico sea emitido.

Los BBS pueden funcionar bajo sistemas operativos diversos, como ser Unix, DOS, Macintosh, Windows, etc. Algunos pueden recibir noticias o news (USENET). Generalmente para la recepción de las mismas se realiza la conexión bajo protocolos UUCP (Unix a Unix CoPy). Por ejemplo, para el intercambio de correo electrónico, ya sea en forma local o internacional, algunos están conectados a Internet vía protocolos UUCP .

## LISTAS DE INTERES

Las listas de interés son sistemas de discusión electrónica, consisten en foros de discusión sobre variados temas. Toda lista posee un responsable para el envío y recepción de mensajes. Utilizan grandes sistemas para almacenar y enviar información. Cada sistema reside en una computadora, la cual posee una dirección de correo electrónico y demás datos para la suscripción a la lista. Existen gran cantidad de listas, los temas abarcan desde las ciencias duras a las ciencias sociales. Una vez que se ha identificado una lista de interés, solo se deberá seguir las indicaciones del responsable de la misma para la suscripción. La suscripción se realiza con un simple mensaje a la dirección indicada por el responsable. A vuelta de correo se recibe la bienvenida a la lista y, periódicamente, se pueden leer todos los mensajes que generan otros usuarios incritos a la lista correspondiente. Por ejemplo, si Ud. es un usuario de una lista de interés y envía un mensaje electrónico a dicha lista, este será compartido por el resto de los usuarios suscritos a la lista.

## BITNET



BITNET cuyo origen se remonta a 1981, es una de las redes mas antiguas y la mas familiar para muchos academicos en disciplinas que no pertenezcan a campos relacionados con las ciencias de la computacion. Dicha red se encuentra conformada por todas las computadoras que utilizan los protocolos NJE, Network Job Entry (Entrada de tareas a la red) y que aparecen en el archivo de ruteo llamado "BITEARN NODES" (Nodos BITEARN). Su implementacion mas comun es el RSCS, Remote Spooling Control System (Sistema remoto de control de spooling).

Administrativamente, BITNET se encuentra dividida en varias partes, segun el area geografica. En los Estados Unidos es administrado por CREN, la Corporacion para Investigacion y Redes Educativas (Corporation for Research and Educational Networking). En Europa, la red y la organizacion es llamada EARN, la Red Europea Academica y de Investigacion (European Academic and Research Network). En Canada, es NetNorth, y el Consorcio NetNorth. Otras organizaciones en otras areas administran otras redes NJE miembros.

Tradicionalmente, BITNET estaba soportada en lineas dedicadas de 9600 bps, que hacen lenta y "dolorosa" la transferencia de archivos grandes. Sin embargo, en anos recientes, porciones grandes de BITNET estan siendo soportadas por NJE sobre conexiones IP mucho mas rapidas, a traves de INTERNET.

## FIDONET

FIDONet es una red que esta conformada, en su mayoria, por computadoras personales basadas en MS-DOS, comunicandose a traves de lineas discadas, muchas a 2400 bps. Soporta correo electronico y un sistema de conferencias llamado echomail.

La mayor ventaja de FidoNet es que es un sistema muy barato, que ha llegado a muchas partes del mundo antes que cualquier otra red. Actualmente existen aproximadamente 16.000 nodos de FidoNet en el mundo.

## UUCP

El nombre de la red UUCP esta dado por su protocolo principal, UUCP (UNIX a UNIX CoPy). Dicha red conecta, principalmente, maquinas UNIX, sobre lineas discadas a velocidades entre 2400 bps y 19200 bps. El unico servicio soportado por la red UUCP es el de correo electronico. Las conexiones a la red UUCP son mucho mas economicas que las anteriores. La red de correo UUCP esta muy relacionada con la red USENET de noticias (servicio ofrecido por INTERNET)

## INTERNET

INTERNET es una gran Red de computadoras unidas entre si por protocolos de comunicacion. Es la red electronica de computadoras mas grande del mundo, a excepcion de la Red Telefonica.

Esta gran red, INTERNET, se extiende por America del Norte, Europa, Japon, Australia, Nueva Zelandia, y otros paises, y aun continua creciendo por el resto de

Asia, America Latina, Antartida, Europa del Este y Africa. Para julio de 1992, por ejemplo, ya existian unos 992.000 nodos de INTERNET distribuidos en el mundo. Cuando la red precursora de INTERNET, ARPANET (1969-1990) fue propuesta, no todos los usuarios utilizaron el servicio de correo electronico. No obstante, al inaugurar la red de redes, INTERNET, en enero de 1983, el correo electronico paso a ser uno de los servicios mas basicos.

Dado que la mayor parte del correo electronico y las noticias o news son textos, y por lo tanto ya han pasado por la compresion de su funcionamiento en el area academica y cientifica, queda acentado que la comunicacion electronica es el servicio mas utilizado en Red. Adicionalmente, el correo electronico es el servicio que cualquier nuevo usuario probablemente usara primero y posteriormente a diario, ya que esta nueva forma de comunicacion provee acceso a millones de usuarios que aun no estan conectados directamente en INTERNET.

Los servicios de INTERNET pueden ser usados desde una computadora personal hasta grandes mainframes (conjunto de computadoras). Los accesos a esta Red varian de acuerdo a las telecomunicaciones locales con las que se cuenta, que pueden ser via dial-up (modem y linea telefonica), slip, linea dedicada, X.25 y ISDN.

Esta Red de Redes, incluye todas las computadoras y todas las redes que trabajan con el protocolo IP (Internet Protocol) y que estan interconectadas con acceso directo de punto a punto. Los protocolos TCP/IP, utilizados por INTERNET, estan siendo incorporados dentro de muchas companias y organizaciones para integrar diversos tipos de servicios computacionales y lograr sistemas distribuidos que utilizan la red para hacer accesibles recursos y comunicaciones, no solo a los sistemas mas grandes de tiempo compartido, sino tambien a estaciones de trabajo (workstations) y computadoras personales. Esto representa la union de areas previamente separadas como UNIX, protocolos de redes como TCP/IP y standards

En la actualidad, los protocolos TCP/IP han dejado atras sus origenes en los laboratorios de investigacion y se encuentran hoy dia ampliamente utilizados en la industria. Aunque no se sabe con certeza cuantos nodos hay en estas redes intra-compania, una estimacion alcanza los 600.000.

Usando los protocolos TCP/IP, INTERNET provee un enlace directo entre computadoras que se encuentran al otro lado del mundo. Si Ud. puede ejecutar el comando que sirve para la transferencia de archivos, denominado FTP (File Transfer Protocol) a las siguientes direcciones: nis.nsf.net, ftp.psi.com, o ftp.uu.net, esta en INTERNET.

En su mayoria, INTERNET esta conformada por redes locales (Local Area Networks-LAN) pertenecientes y operadas por companias privadas, universidades, gobiernos, ONG y otras organizaciones. Estas redes locales se encuentran interconectadas, tambien, por redes metropolitanas (Metropolitan Area Networks-MAN), de origenes estatales, provinciales y regionales. Y estas, a su vez, estan interconectadas por redes nacionales e internacionales, como por ejemplo CA\*net en Canada, NORDUnet en los paises nordicos (Dinamarca, Finlandia, Suecia, Noruega, Islandia, etc.) y WIDE, TISN y JAIN en Japon.

Tanto en Japon como en Estados Unidos existen varias redes IP nacionales, pero a diferencia del Japon, en Estados Unidos existen agencias estatales intimamente

vinculadas al desarrollo de estas redes. La NSF, National Science Foundation (Fundacion Nacional para las Ciencias), es actualmente la agencia gubernamental lider en el desarrollo de INTERNET, aunque es, solamente, una de las tantas agencias gubernamentales involucradas. La mayor parte de las agencias involucradas en el desarrollo de INTERNET son de caracter no-gubernamental, y una gran parte de dichas agencias se encuentran fuera del area geografica de los Estados Unidos. Lo importante es saber que no existe una unica organizacion que controle INTERNET, por eso se la denomina, tambien, la Red de Redes.

Las redes nacionales y regionales que se encuentran interconectadas en INTERNET, estan agrupadas en un consorcio llamado CIX, "Commercial Internet Exchange". Los usuarios de cada red miembro tiene acceso IP a los usuarios de cualquiera de las otras redes miembro.

Algunas de estas redes miembros del consorcio comercial CIX son:

\* PSInet y AlterNet, proveedores de servicios IP en Estados Unidos, operados por Performance Systems International y por UUNET Technologies

\* CERFnet y BARRnet , redes regionales en California

\* US Sprint ,con su servicio Sprintlink y UniPalm Ltd., de Cambridge, Reino Unido, con su servicio PIPEX

\* NSFNET, ANSnet, y CO+RE, administradas por Advanced Network and Services, Inc., o sus subsidiarias. NSFNET es parcialmente financiada por la National Science Foundation. En esta agrupacion, por ejemplo, los usuarios de NSF deben acatar un acuerdo sobre "uso aceptable" de la red, que restringe el trafico comercial; otras redes regionales que se conectan a NSFNET tambien tienen restricciones comerciales para sus usuarios, en cambio ANSnet y CO+RE tienen restricciones diferentes.

Tambien existen redes IP comerciales intra-compania. Una red intra-compania es utilizada para las funciones de esa compania, y por razones de privacidad, es directamente accesible unicamente a los empleados de esa compania. En su mayoria, estas redes no se encuentran directamente interconectadas con INTERNET, sin embargo, se les permite el intercambio de correo electronico con la gran Red.

=====

## INFORMACION Y SERVICIOS EN INTERNET

=====

El tipo de informacion que Ud. podra encontrar en INTERNET corresponde a documentos gubernamentales, trabajos que han expirado sus derechos de autor, trabajos que son de dominio publico. Uno de los servicios que ofrece INTERNET mas importante en cuanto a su participacion y crecimiento es: USENET

USENET es una Red no real de "noticias", un conjunto de nodos que proveen un servicio que soporta un sistema de conferencias asincronicas, llamado "news" (noticias). El trafico USENET es llevado por varias redes, incluyendo UUCP, BITNET, FidoNet, etc. Cuando decimos una "red no real" significa que no es una red en si misma, sino que utiliza los protocolos existentes en redes independientes para hacer el transporte de las noticias. Actualmente existen en el mundo unos 55.000 nodos USENET.

En síntesis, USENET es un gran boletín en el cual gran cantidad de personas se conectan e intercambian información pública. Por ejemplo, las más usuales de las miles de conferencias existentes son:

alt.sex  
misc.jobs.offered  
rec.humor.funny  
rec.arts.erotica  
rec.music.misc  
talk.politics.misc  
alt.activism  
news.announce.conferences

Otros nuevos servicios de fuentes de información en las redes son los que se refieren a Gopher, WWW, WAIS, Archie, WHOIS, Netserv, Trickle, BitFTP, Usenet y Listservers.

Para recibir información actualizada enviar un mail a:

listserv@earncc.bitnet

con el siguiente mensaje:

send nettools memo

o via FTP a:

ds.internic.net:/pub/internet doc/EARN.nettoo

ls.txt

Dentro de INTERNET existen una serie de opciones para poder acceder a la información, y las más comunes son:

"TELNET": Es un programa que permite la conexión a otra computadora. Telnet necesita conocer el nombre de la computadora remota (host remoto) o en su defecto el número que corresponde a dicho host.

Por ejemplo: telnet steer.sdsu.edu

telnet 130.191.1.11

En este ejemplo nos estaríamos comunicando con la dirección steer.sdsu.edu o eventualmente con el número que corresponde a esa dirección.

"FTP ANONIMO": FTP permite copiar archivos desde una computadora remota a su computadora local (host local). Cientos de sitios proveen servicios de FTP ANONIMO. Para que Ud. pueda recuperar información, es necesario que su computadora use protocolos de comunicación como Xmodem, Zmodem, Kermit u otros. Dentro de FTP existen una serie de comandos para recuperar la información, los cuales son:

help: ayuda

open: nodo al cual conectarse

cd: cambiar directorio

get: traer un archivo

compress: comprimir el resultado

uuencode: convertir un archivo binario a ASCII

"FINGER": Es un programa que le permite a Ud. buscar información sobre usuarios registrados en la INTERNET. El comando funciona de la siguiente manera:

finger ron@hal.gnu.ai.mit.edu

En este ejemplo, estamos buscando al usuario RON con una cuenta de correo electrónico en el MIT.

## GLOSARIO

APC: Asociación para el Progreso de las Comunicaciones. Algunas de las redes que conforman dicha Asociación son: Alternex (Brasil), Fredsnaetet (Suecia), GreenNet (Inglaterra), Nicarao (Nicaragua), Peacenet/EcoNet/ConflicNet/HomeoNet (USA), Pegasus (Australia) y Web (Canada).

ASCII: (texto) Código estándar de intercambio de información.

Asíncronico: Forma y estilo de transmisión de datos en la cual el envío y la recepción no están sincronizadas.

Bell standards: Modem estándar de los Estados Unidos, desarrollado por Bell.

Binario: Archivo de computadora no textual.

Bulletin Board System (BBS): Sistema que trabaja en un computador brindando servicios, consulta a documentos organizados por temas (foros), acceso al sistema operativo del computador para poder dar comandos, etc. Aunque el acceso suele ser vía módem, también se puede operar el sistema localmente.

Casillero electrónico: (cuenta electrónica) Es un espacio en la memoria de su computadora, dentro de su cuenta personal, en el cual Ud. podrá realizar las actividades deseadas de las comunicaciones electrónicas.

Comunicación electrónica: Comunicación entre dos computadoras o más, a través de un módem, una línea de teléfono y un software de comunicación.

Conexión remota: Tipo de comunicación que permite acceder a un computador remoto como si se tratase de una terminal local del mismo.

Host: Computador central con el cual se comunican otros computadores de menor jerarquía y usualmente de menores recursos.

Intercambio de archivos: Transmisión de volúmenes de información en forma rápida y confiable.

Línea dedicada: Línea de teléfono dedicada exclusivamente para realizar las llamadas entre los módems.

Login: Nombre de identificación personal para acceder a los sistemas de comunicaciones electrónicas.

Mensajes interactivos: Intercambio de información breve transmitido en forma inmediata entre personas conectadas simultáneamente.

Módem: Es una placa o parte de un hardware de su computadora

que sirve para que las computadoras puedan comunicarse por línea telefónica con otros computadores. Su tarea fundamental es la permitir que los datos digitales viajen como señales analógicas.

**Modem externo:** Parte del hardware de su computadora que sirve para comunicarse con otra/s computadoras o redes y que se encuentra fuera de su CPU.

**Modem interno:** Es una placa, parte del hardware de su computadora que sirve para comunicarse con otra/s computadoras o redes y que se encuentra instalado dentro de su CPU.

**Modem local:** Es el modem que se encuentra en su computadora, desde el cual Ud. realiza la llamada al modem remoto.

**Modem remoto:** Es el modem que se encuentra en otra computadora, a la cual Ud. llamará para comunicarse desde su modem local.

**News (noticias):** Sistema de información permanente, por medio del envío de mensajes electrónicos organizados por temas de interés.

**Password:** Clave de acceso personal y privada para acceder a los sistemas de comunicación electrónica.

**Post:** Sistema para envío de mensajes electrónicos organizados por temas de interés.

**Prompt:** Posicionamiento del cursor en la pantalla de una computadora.

**Protocolo: (de comunicación)** Norma para el intercambio de información por vía electrónica.

**Servidores de archivos:** Distribución automática de archivos de interés general.

**Smarthost:** Computador "host" que, por estar mejor acondicionado, puede administrar la mensajería electrónica y "news" (noticias) de las computadoras que tiene "a su cargo".

**Software de comunicación:** Tipo de programa que controla la transferencia de datos dentro y fuera de la computadora. Es siempre necesario para la comunicación electrónica.

**Subject: (tema)** Indica la referencia o el contenido del mensaje.

**Teleconferencia:** Conferencia o discusión electrónica entre varios participantes utilizando como herramienta los mensajes interactivos.

**Terminal:** Computadora que permite el acceso al sistema central.

**Usuario:** Entidad autorizada a explotar los recursos de un sistema, para ello se le identifica con un "nombre de usuario" (user-id o login) y con una clave de acceso (password).

Usuario "Interactivo": Usuario que tiene la posibilidad de ejecutar una serie de comandos durante su conexión.  
Usuario "local": Usuario que accede a la consola del sistema o, eventualmente, a una de sus terminales.  
Usuario "remoto": Usuario que accede al sistema via modem.  
Usuario "tipo Unix": Usuario que se limita a enviar y recibir archivos durante la conexión y ejecuta, terminada esta, los comandos consignados en algunos de los archivos transferidos.  
UUCP: (Unix-to Unix Copy Program, o Unix-to Unix Communications Protocol). Es el conjunto de conceptos que permiten copiar archivos de un sistema Unix a otro en forma automática.

=====  
BIBLIOGRAFIA

=====  
Cisler, Steve, "At play in the field of the boards: report on ONE BBSCON", en Matrix News, vol. 2, nro. 7, de Quarterman, John S., 1992.  
Cisler, Steve, "What is the INTERNET?", en The whole? INTERNET, 1992.  
Contacto: Como diseñar Redes de Comunicación, ILET, Santiago, 1987.  
Dunayevich, Julian, Interconectividad en Redes, Facultad de Ciencias Exactas y Naturales, Centro de Comunicación Científica, Buenos Aires, 1994.  
Lane, Graham, Communications for progress: A guide to international email, Antenna, CIIR, EDRC, Interdoc, 1990.  
Manual del Usuario: version 1.0, Instituto Brasileiro de Analises Sociais e Economicas, (IBASE), Alternex, Rio de Janeiro, 1992.  
Matrix News, vol. 2, nros. 6, 8, 9, 10, 12, 1992.  
Matrix News, vol. 3 nros. 1, 3, 4, 5, 6, 7, 8, 1993.  
McClure, Charles y otros, The National Research and Education Network (NREN). Research and policy perspectives, New Jersey, 1991.  
Piscitelli, Alejandro y Navarro, Gustavo, CLACSO en Las Redes Electronicas y el Proyecto Vision, Buenos Aires, CLACSO, 1992.  
"Proyecto Red Nacional de Ciencia, Educación y Tecnología: La Red Científica Peruana, Herramienta de desarrollo", informes presentado en la Reunion sobre Redes Electronicas, CLACSO, Rio de Janeiro, 1992.  
Quarterman, John S., "Network Host Growth Rates", en Matrix News, vol. 3, nro. 7, 1993.  
Rittner, Don, The whole earth online almanac: Info from A to Z. New York, Brady Publishing, 1993.

Ruth, Stephen, Utreras, Florencio y Brescia, Ralph P. "The South American Scientific Network within one year: An attainable, low cost, high yield reality", en *Interciencia*, sept 1990, vol. 15, nro. 5.



## INTERNET

### **FUNDAMENTOS DEL EXPLORADOR.**

[¿En qué consiste un explorador Web?](#)

[¿Qué es una dirección URL?](#)

[Sugerencias acerca del explorador](#)

[La caché y la barra de herramientas](#)

[¿Qué es HTML?](#)

[¿Por qué hace falta tanto tiempo para descargar algunas páginas Web?](#)

[¿Qué es Active Desktop?](#)

[¿Cuál es la relación entre FTP y la descarga de archivos?](#)

### **¿En qué consiste un explorador Web?**

Un explorador Web contiene el software necesario para buscar, recuperar, ver y enviar información a través de Internet. Incluye software que le permite:

- **Enviar y recibir** mensajes de correo electrónico (e-mail) a todo el mundo de forma casi instantánea.
- **Leer** mensajes de grupos de noticias, foros acerca de miles de temas en los que los usuarios comparten información y opiniones.
- **Explorar** el World Wide Web (o Web), donde podrá encontrar gran variedad de textos, gráficos e información interactiva.

Los exploradores como Microsoft® Internet Explorer versión 4.0 incluyen software adicional relacionado con Internet. Por ejemplo, con Internet Explorer también obtendrá:

- Servidor NetShow™
- Software de conferencia NetMeeting™
- Controles ActiveX®
- Chat
- La interfaz de programación de aplicaciones DirectShow™ (anteriormente ActiveMovie™)
- Difusión por Web de Active Channel™
- Suscripciones
- HTML dinámico

### **Servidor NetShow**

Esta aplicación le permite ver y oír transmisiones grabadas y en directo, como conciertos o las últimas noticias, con sonido, gráficos, vídeo, direcciones URL y secuencias de comandos sincronizadas. La tecnología de secuencias le permite ver y oír la información según llega, en lugar de tener que esperar a que se descargue todo el archivo.

### **Software de conferencia NetMeeting**

Con una tarjeta de sonido, unos altavoces y un micrófono puede hablar con otras personas en cualquier parte del mundo, desde la familia hasta los compañeros,

gracias a NetMeeting. Agregue una tarjeta de captura de vídeo o cámara compatible con Windows para poder verles también. Intercambie imágenes y dibuje diagramas en una pizarra electrónica, comuníquese mediante conversaciones de texto, transfiera archivos y comparta aplicaciones.

### **Controles ActiveX**

La tecnología ActiveX permite que los autores desarrollen sitios Web innovadores y plenamente interactivos. Los controles ActiveX son los componentes de software que actúan de manera invisible en Internet Explorer para hacer que estos sitios cobren vida.

### **Chat**

Este programa le permite mantener conversaciones electrónicas en tiempo real con una o más personas. Puede decidir que el mensaje aparezca sólo como texto o como texto y gráficos. Puede enviar y recibir sonidos, archivos y vínculos activos para direcciones de correo electrónico, páginas Web y grupos de noticias. Incluso puede "susurrar" algo a otra persona del grupo con el que habla. Utilícelo en su próxima reunión familiar electrónica.

### **Interfaz de programación de aplicaciones ActiveMovie**

ActiveMovie le permite utilizar vídeo con calidad de televisión y sonido con calidad de CD, al tiempo que reduce al mínimo el tamaño del archivo y el tiempo de descarga si se compara con otros formatos de sonido y vídeo. Si utiliza la "descarga progresiva", ActiveMovie le permitirá comenzar a reproducir un clip de audio o de vídeo mientras aún se está descargando.

### **Active Channel**

La difusión por Web de Active Channel permite enviar información dinámica de forma regular a su sistema. Transmite automáticamente el contenido que se adapte a sus intereses, desde los productos de Disney hasta la cotización de la bolsa. Utilice la barra de canales para seleccionar sus temas favoritos e Internet Explorer 4.0 obtendrá la información para que pueda leerla cuando desee, incluso si trabaja sin conexión.

### **Suscripciones**

Esta característica envía la información seleccionada directamente a su escritorio, cuando la desee, y tal y como la desee de forma gratuita. Para suscribirse a un sitio Web, selecciónelo e indique cuándo desea que se actualice la información y cómo desea ser avisado, por ejemplo, mediante un mensaje de correo electrónico. Internet Explorer se encarga del resto. Después podrá ver el contenido sin estar conectado.

### **HTML dinámico**

Internet Explorer 4.0 permite utilizar este lenguaje de programación que hace posible crear páginas Web seductoras, únicas, divertidas y de rápida descarga. Las páginas se descargan rápidamente porque están creadas en código HTML ligero, en lugar de hacer un uso intensivo de gráficos. La frecuencia de acceso al servidor se ve reducida, por lo que aumenta el rendimiento del explorador en su equipo.

### **Descargue Internet Explorer 4.0**

Microsoft Internet Explorer 4.0 aprovecha todas las características que ya conoce y le gustan del sistema operativo Windows 95 e Internet Explorer 3.0. Además, entrega la información que desea directamente a su equipo. Si esto se combina

con la nueva herramienta inteligente para delimitar las búsquedas de Web y obtendrá inmediatamente información acerca de sus temas favoritos. Pero lo mejor es que Internet Explorer 4.0 incluye un conjunto único de aplicaciones, como correo electrónico, lectores de grupos de noticias y Chat, que hacen que sea fácil compartir sus ideas con su familia y sus amigos.

Visite el [sitio Web de Microsoft Internet Explorer 4.0](#) para obtener más información acerca de las novedades en Internet Explorer 4.0. Si no dispone de Internet Explorer, puede descargarlo desde aquí.

Si no dispone de un explorador Web, póngase en contacto con su ISP para saber cómo debe conectar para descargar uno.

¿Está harto del "correo lento"? Obtenga más información acerca de [Microsoft Internet Mail and News](#). Así podrá redactar sus mensajes una vez que [haya descargado](#) Internet Mail and News.

Obtenga más información acerca del software de conferencia [NetMeeting](#) y podrá comprobar sus ventajas si [descarga](#) hoy mismo el programa.

Para obtener más información acerca de Microsoft Chat, visite la página Web de [Microsoft Chat](#). Puede probar una [conversación de ejemplo](#) y si quiere estar en contacto con aquellos que más le importan, [descargue](#) Microsoft Chat.

Visite la página Web de [NetShow](#) para obtener más información acerca de esta característica de Internet Explorer, después, [descargue](#) el servidor NetShow para ver cómo puede llevar a su escritorio todo el potencial de la transmisión de información.

Visite la [página Web de ActiveMovie](#) para obtener más información acerca de esta característica de Internet Explorer.

## ¿Qué es una dirección URL?

Una dirección URL, o localizador de recursos uniforme, es la dirección de un archivo en Internet. Normalmente consta de cuatro partes: protocolo, servidor (o dominio), ruta y nombre del archivo. A veces falta la ruta o el nombre del archivo. Aquí hay un ejemplo:

<http://www.microsoft.com/magazine/tips/default.htm>

- *http* es el protocolo.
- *www.microsoft.com* es el servidor, o dominio.
- *magazine/tips* es la ruta.



## Sugerencias acerca del explorador

Aproveche todas las características interesantes de Microsoft® Internet Explorer versión 4.0. Estas sugerencias le ayudarán.

## Identifique los vínculos

Puede saber si un elemento de una página es un vínculo con sólo colocar el puntero del mouse sobre él. Si la forma del puntero cambia a una mano, el elemento es un vínculo, ya sea una fotografía, una imagen en tres dimensiones o texto en color (normalmente subrayado). Haga clic en cualquier vínculo de una página Web para ir a otra página, en el mismo o en otro sitio.

## Mostrar las páginas Web más rápidamente

Para mostrar las páginas Web más rápidamente:

1. En el menú Ver del explorador, haga clic en **Opciones de Internet**.
2. Seleccione la ficha **Opciones avanzadas**.
3. En el área Multimedia, desactive una o varias de las opciones siguientes: **Mostrar imágenes**, **Activar animaciones**, **Mostrar vídeos** o **Reproducir sonidos**.
4. Si desactiva la opción Mostrar imágenes o Mostrar vídeos, aún podrá ver cualquier imagen de la página Web si hace clic en su icono con el botón secundario del mouse y selecciona la opción Mostrar imagen.
5. Si las imágenes de la página actual siguen visibles después de desactivar la opción Mostrar imágenes, puede ocultarlas si hace clic en el menú Ver y selecciona Actualizar.

## Mostrar más rápidamente las páginas ya visitadas

Para mostrar más rápidamente las páginas ya visitadas :

1. En el menú Ver del explorador, haga clic en **Opciones de Internet**.
2. En la ficha General, seleccione **Configuración**.
3. Para crear más espacio para almacenar las páginas temporalmente, mueva a la derecha la barra de desplazamiento.
4. Para evitar que Internet Explorer actualice las páginas de la carpeta Temporary Internet Files, haga clic en **Nunca**.

## Cambiar la presentación de los colores de la página

Para cambiar la presentación de los colores de la página:

1. En el menú Ver del explorador, haga clic en **Opciones de Internet**.
2. Cambie la configuración según sea necesario.

## Utilizar una fuente diferente para mostrar el texto

Para utilizar una fuente diferente para mostrar el texto:

1. En el menú Ver del explorador, haga clic en **Opciones de Internet**.
2. En la ficha General, seleccione **Fuentes**.
3. Seleccione las fuentes que desee utilizar en las listas de Fuente Proporcional y Fuente de ancho fijo.

## Especificar la fuente y el color que se utilizará siempre

Para especificar la fuente y el color que se utilizará siempre:



1. En el menú Ver del explorador, haga clic en **Opciones de Internet**.
2. En la ficha General, seleccione **Accesibilidad**.
3. Cambie la configuración según sea necesario.

## **Aumentar o reducir el tamaño del texto**

Para aumentar o reducir el tamaño del texto:

- En el menú Ver, vaya a Fuentes y seleccione el tamaño que desee utilizar.

## **Ver páginas Web en otro idioma**

Algunos sitios Web ofrecen su contenido en varios idiomas. Puede agregar idiomas a la lista de idiomas en Internet Explorer, de forma que pueda ver estas páginas en su idioma preferido.

Para ver las páginas Web escritas en un idioma diferente:

1. En el menú Ver del explorador, haga clic en **Opciones de Internet**.
2. En la ficha General, haga clic en **Idiomas**.
3. Haga clic en **Agregar**.
4. Seleccione el idioma que desea agregar.
  - Si habla varios idiomas puede organizarlos según su prioridad. Si un sitio Web ofrece varios idiomas mostrará el contenido en el idioma que tenga la prioridad más alta.
  - Agregar idiomas no garantiza que disponga de una fuente que permita mostrar las páginas Web en el idioma preferido. Puede que necesite descargar un paquete de soporte multilinguaje para poder mostrar estas páginas.

## **Agregar una página a Favoritos**

Para agregar una página a su colección de páginas favoritas:

1. Vaya a la página que desee agregar a su colección de páginas favoritas.
2. En el menú Favoritos, haga clic en **Agregar a Favoritos**.
3. Si lo desea, escriba un nuevo nombre para la página.
  - Para abrir una de sus páginas favoritas, haga clic en el botón Favoritos de la barra de herramientas y seleccione la página que desea abrir.
  - Para no perder sus páginas favoritas, puede organizarlas en carpetas. Haga clic en el botón Crear en del cuadro de diálogo Agregar a Favoritos.

## **Agregar Informática personal a Favoritos**

Para asegurarse de que siempre tiene acceso a las últimas noticias de Internet, actualizaciones de software y trucos y sugerencias para utilizar Internet Explorer, ¿por qué no agrega el sitio de Informática personal a la lista de Favoritos?

Siga estos pasos para agregar el sitio de Informática personal a la lista de Favoritos:

1. En el menú Archivo de la barra de herramientas de Internet Explorer vaya a **Nuevo** y haga clic en **Ventana**, así no perderá esta guía de vista.
2. En el menú Archivo de la nueva ventana, haga clic en **Abrir** y escriba **<http://www.microsoft.com/magazine/>** en el cuadro de direcciones.
3. Haga clic en **Aceptar**.



4. Cuando termine la descarga de la página de Informática personal, en el menú Favoritos, haga clic en **Agregar a Favoritos** y después en **Aceptar**.
5. Cierre la nueva ventana.

### **Organizar los Favoritos en carpetas**

Para organizar en carpetas sus páginas favoritas:

1. En el menú Favoritos, haga clic en **Organizar Favoritos**.
2. Haga clic en **Crear nueva carpeta**, escriba un nombre para ella y presione **ENTRAR**.
3. Arrastre los accesos directos de la lista a las carpetas adecuadas.
  - Puede que desee organizar las páginas por temas. Por ejemplo, puede crear una carpeta llamada Arte para guardar toda la información sobre las exposiciones de arte.
  - Si el número de accesos directos o carpetas hace que arrastrar los elementos resulte incómodo, puede utilizar el botón Mover.

### **Cambiar la página principal**

Para cambiar la página principal:

1. Vaya a la página que desea que aparezca al iniciar Internet Explorer.
2. En el menú Ver, seleccione **Opciones de Internet**.
3. Haga clic en la ficha **General**.
4. En el área Página principal, haga clic en **Usar actual**.
  - Para restaurar su página principal original, haga clic en **Predeterminada**.

### **Guardar texto y gráficos del Web**

Cuando vea texto o gráficos en una página Web y le gusten o desee utilizarlos como referencia más adelante, puede guardarlos en el disco duro. Después puede abrir el archivo y verlo sin estar conectado.

Para guardar un archivo de origen o de texto:

1. En la barra de herramientas, haga clic en **Archivo** y después en **Guardar como**.
2. Haga clic en **Guardar** para guardar el archivo.

### **Para guardar una imagen**

1. Haga clic en la imagen con el botón secundario del mouse.
2. En el menú contextual que aparece, haga clic en **Guardar imagen como**.
3. Vaya a la carpeta en la que desea guardar el archivo.
4. Haga clic en **Guardar** para guardar el archivo.

Para abrir un archivo guardado anteriormente, haga doble clic en él desde la carpeta en la que lo guardó. Microsoft Internet Explorer se ejecutará automáticamente y aparecerá el archivo en la ventana del explorador.

### **Agregar una página a la barra de vínculos**

Para agregar una página a la barra de vínculos<sup>0</sup>:

1. Arrastre el icono de la página desde la barra de direcciones hasta la barra de

vínculos.

2. Arrastre un vínculo desde una página Web hasta la barra de vínculos.
  3. Arrastre un vínculo a la carpeta Vínculos de la lista de Favoritos: también puede arrastrarlo directamente al menú Favoritos y después a la carpeta Vínculos o puede arrastrarlo a la carpeta Vínculos cuando se muestren los Favoritos en la barra del explorador.
- También puede organizar sus vínculos si los arrastra a un lugar diferente de la barra de vínculos.

### **Crear un acceso directo de escritorio para la página actual**

- Haga clic en la página con el botón secundario del mouse y seleccione **Crear acceso directo**.
- Si la ventana de Internet Explorer no está maximizada, también puede crear un acceso directo si arrastra un vínculo desde esta ventana hasta la ubicación en la que desee crearlo, como por ejemplo una carpeta o el escritorio.

### **Volver a una página Web que ya ha visitado**

Existen varios modos de volver a una página Web en la que ya haya estado:

- Para volver a la última página visitada, puede hacer clic en el botón Atrás de la barra de herramientas o presionar la tecla RETROCESO.
- Para ver una lista de las últimas páginas visitadas, haga clic en la pequeña flecha situada a la derecha del botón Atrás o Adelante. Después, seleccione la página que desee.
- Si desea ver una de las últimas cinco páginas visitadas en esta sesión, haga clic en el menú Archivo y, en la lista de la parte inferior, seleccione la página a la que desea ir. Esta lista se reinicia cada vez que inicia Internet Explorer.
- Para ver más páginas, incluidas las visitadas en sesiones anteriores, haga clic en el botón Historial de la barra de herramientas y seleccione la carpeta correspondiente.

### **Cambiar la apariencia de la barra de herramientas**

Para cambiar la apariencia de la barra de herramientas:

- Puede mover o cambiar el tamaño de la barra de direcciones o de la barra de vínculos si las arrastra hacia arriba, abajo, izquierda o derecha. Incluso puede colocarla en la barra de menús.
- Para dejar más espacio libre en su pantalla puede ocultar las etiquetas de los botones de la barra de herramientas. Sólo tiene que hacer clic en ésta con el botón secundario del mouse y desactivar la marca situada junto a Etiquetas.
- Puede ocultar la barra de direcciones o la sección Vínculos de la barra de herramientas si hace clic en ella con el botón secundario del mouse y desactiva la marca situada junto a cada uno de los elementos que desee ocultar.
- Puede agregar elementos a la barra de vínculos si arrastra el icono desde la barra de direcciones o arrastra un vínculo desde una página.
- Puede reorganizar los elementos de la barra de vínculos si los arrastra a una nueva ubicación en la barra.
- Puede utilizar botones más pequeños para la barra de herramientas, parecidos

a los de Microsoft Office. En el menú Ver del explorador, haga clic en Opciones de Internet y seleccione la ficha Opciones avanzadas. En el área de la barra de herramientas, seleccione la casilla de verificación Iconos pequeños.

### La caché y la barra de herramientas

Si ha llegado hasta esta página, ya debe saber algo acerca de cómo utilizar un explorador Web. Por si aún no ha utilizado alguna de las características de Microsoft Internet Explorer, a continuación se muestra una guía.



### Caché

Al explorar la Web, su explorador registra las páginas que ha visitado y las guarda en el disco duro para que se carguen más rápidamente cuando vuelva a ellas. Así ahorra tiempo y dinero, ya que puede ver estas páginas sin necesidad de estar conectado a Internet. Los archivos guardados, sus "archivos temporales de Internet", se guardan en la caché de disco.



### Para vaciar la caché de disco de Internet Explorer

Mientras explora la Web, la caché de disco puede llenarse con archivos que no va a necesitar. A continuación se indica cómo puede vaciar la caché de disco de Internet Explorer.

Para Internet Explorer versión 3.0:

1. En el menú Ver de la barra de herramientas de Internet Explorer, haga clic en **Opciones**.
2. Seleccione la ficha **Avanzado**.
3. En el área Archivos temporales de Internet, haga clic en **Configuración**.
4. Después, haga clic en el botón **Vaciar carpeta**.

Para Internet Explorer versión 4.0:

1. En el menú Ver de la barra de herramientas de Internet Explorer, haga clic en **Opciones de Internet**.
2. Haga clic en la ficha **General**.
3. En el área Temporary Internet files, haga clic en el botón **Eliminar archivos** y después en **Aceptar**.
4. Haga clic en **Aceptar** para cerrar **Opciones de Internet**.



### Para cambiar el tamaño de la caché de disco de Internet Explorer

Puede cambiar la cantidad de espacio reservado para la caché de disco. Cuanto más grande sea más rápidamente se podrán cargar las páginas ya visitadas, pero disminuirá la cantidad de espacio disponible para otros archivos. A continuación se indica cómo puede configurar el tamaño de la caché de disco.

Para Internet Explorer 3.0:

1. En el menú Ver de la barra de herramientas de Internet Explorer, haga clic en **Opciones**.



2. Seleccione la ficha **Avanzado**.
3. En el área Archivos temporales de Internet, haga clic en **Configuración**.
4. Mueva la flecha del control deslizante **Espacio en disco a usar** hasta alcanzar el porcentaje de espacio en disco que desea asignar a la caché.
5. Haga clic en **Aceptar** para guardar la configuración.

Para Internet Explorer 4.0:

1. En el menú ver de la barra de herramientas de Internet Explorer, haga clic en **Opciones de Internet**.
2. Seleccione la ficha **General**.
3. En el área Temporary Internet Files, haga clic en **Configuración**.
4. Mueva la flecha del control deslizante **Ocupar el siguiente espacio en disco** hasta alcanzar el porcentaje de espacio en disco que desea asignar a la caché y después haga clic en **Aceptar**.
5. Haga clic en **Aceptar** para cerrar Opciones de Internet.

## **Barra de herramientas**

La barra de herramientas de Internet Explorer está formada por botones que son accesos directos para comandos del menú y que hacen que explorar sea más rápido y fácil.

**Atrás.** Le permite volver a páginas que ya ha visitado, a partir de la más reciente. En Microsoft Internet Explorer versión 4.0, haga clic con el botón secundario del mouse en el botón Atrás y seleccione la página en la lista de los sitios visitados recientemente.

**Adelante.** Le permite moverse hacia adelante para pasar a páginas que ha visto mediante el botón Atrás. En Microsoft Internet Explorer versión 4.0, haga clic con el botón secundario del mouse en el botón Adelante y seleccione la página en la lista de los sitios visitados recientemente.

**Detener.** Detiene el proceso de descarga de una página Web. Haga clic en él si por algún motivo desea detener la descarga de una página -- por ejemplo, si tiene problemas para descargarla o si no desea esperar a que termine. Después podrá intentar descargarla de nuevo o ir a algún otro sitio.

**Actualizar.** Actualiza cualquier página Web almacenada en la caché con su contenido más reciente. (Cuando vuelve a una página ya visitada el explorador muestra el archivo guardado en la caché, no la página actual en el World Wide Web. Así se ahorra tiempo en la descarga.)

**Inicio.** Vuelve a la página de inicio. Puede seleccionar cualquier página Web como página de inicio.

**Búsqueda.** Muestra una selección de motores de búsqueda populares en Internet en el panel izquierdo del explorador. Los resultados también aparecen en ese panel. Al hacer clic en un vínculo, la página aparece en el panel derecho, por lo que no pierde de vista los resultados de la búsqueda.

**Favoritos.** Muestra una lista de los sitios (y, con Internet Explorer 4.0, las carpetas, archivos y servidores) que ha guardado como Favoritos. Haga clic en cualquier elemento de la lista para saltar a él.

**Imprimir.** Imprime la página que está viendo. Es uno de los métodos para guardar información de Internet de forma que no necesite volver a conectar para

consultarla de nuevo. Incluso puede imprimir la dirección URL asociada con cada hipervínculo, lo que hará que sea más fácil ir a ese sitio más adelante.

**Fuente.** Le permite usar una fuente más grande o más pequeña para mostrar el texto en Internet Explorer 4.0.

**Correo.** Conecta con el cliente de mensajería y colaboración Microsoft Outlook™ Express para poder leer los mensajes de correo electrónico y de los grupos de noticias.

**Edición.** Abre un archivo en el procesador de textos Microsoft Word en el que se muestra el código HTML de la página que está viendo, de forma que puede verlo e incluso modificarlo.

### ¿Qué es HTML?

HTML, o lenguaje de marcado de hipertexto, es un lenguaje de programación que se utiliza para crear sitios Web. Contiene códigos estándar, o etiquetas, que determinan la apariencia de una página Web y cómo la mostrará el explorador. Las etiquetas HTML también permiten utilizar los hipervínculos que conectan la información del World Wide Web.

Consulte la sección "[Crear un sitio Web](#)".

### ¿Por qué hace falta tanto tiempo para descargar algunas páginas Web?

Existen muchos factores que afectan a la velocidad a la que se mueve una página Web desde su sitio Web hasta su equipo. Las páginas Web pueden cargarse lentamente por estos motivos:

- Su módem transmite los datos lentamente.
- Hay mucho tráfico en Internet.
- El archivo de la página es grande. Contiene, por ejemplo, archivos de gráficos, sonido o vídeo además del texto.
- El servidor y las líneas de conexión utilizadas por su servicio electrónico, su proveedor de acceso a Internet o el sitio Web transmiten los datos lentamente.

Para reducir el tiempo de descarga puede probar a:

- Obtener la última versión de su explorador.
- Comprar un módem más rápido que transmita por lo menos a 28.800 bits por segundo.
- Conectar fuera de las horas de mayor acceso.
- Desactivar el sonido y las imágenes en su explorador. Las páginas en las que sólo hay texto se cargan más rápidamente.
- Si una página se carga tan despacio que parece estar bloqueada, haga clic en el botón Detener de la barra de herramientas de su explorador e intente cargarla de nuevo pasados unos minutos.

### ¿Qué es Active Desktop?

La interfaz de Active Desktop™ es una característica de Microsoft® Internet Explorer versión 4.0 que permite utilizar el escritorio como un explorador. A continuación se indican algunas de las cosas que se pueden hacer con él:

- Obtener acceso sencillo e instantáneo a las aplicaciones que más utiliza, incluidas herramientas tales como directorios de la empresa y motores de búsqueda.
- Mostrar varias páginas HTML e imágenes del World Wide Web, de su red local o del disco duro.
- Crear ventanas en su escritorio para los objetos que le interesan, de forma que reciba directamente la información que necesite.
- Colocar componentes ActiveX® y Java junto a los iconos existentes en el escritorio, lo que le permite incluir en él su contenido Web favorito y los avisos de correo electrónico.

### **¿Qué es FTP y cuál es su relación con la descarga de archivos?**

FTP es la abreviatura de "Protocolo de transferencia de archivos." Es un sistema de reglas que permite la comunicación a través de Internet, así como transferir archivos y descargarlos de otros sistemas. Un explorador como Microsoft® Internet Explorer contiene las herramientas necesarias para controlar las transferencias de FTP. Por tanto, con Internet Explorer puede descargar cualquier archivo disponible en Internet.

---

### ***¿CÓMO BUSCO ALGO?***

Una vez que esté conectado a Internet, necesitará saber cómo encontrar los sitios Web y los grupos de noticias que le pueden interesar. Puede que le parezca divertido, como a mucha gente, pasear (o explorar) de un sitio a otro, yendo a donde quiera que le lleven los vínculos. Pero si desea encontrar algo más concreto y lo antes posible, necesitará utilizar un motor de búsqueda. He aquí los fundamentos para realizar una búsqueda:

[Motores de búsqueda: cómo encontrar una aguja en un pajar](#)  
[Deja News: cómo buscar en los grupos de noticias](#)  
[Sugerencias para las búsquedas](#)

### **Motores de búsqueda: cómo encontrar una aguja en un pajar**

Un motor de búsqueda es un servicio que crea índices, organiza y a menudo valora y revisa los sitios Web. Le ayuda a encontrar la aguja - ese sitio Web que tiene que ver - en el pajar de Internet. Cada motor de búsqueda funciona de una forma diferente.

- Algunos dependen de personas para mantener un catálogo de páginas o sitios Web.
- Algunos utilizan programas para identificar la información importante en los sitios de Internet.
- Otros combinan ambos servicios.

Por tanto, cuando realice búsquedas en sus "dominios", obtendrá diferentes resultados.



### ¿Cuál debería utilizar?

Ningún motor de búsqueda mantiene un registro con todo el contenido de Internet. Hasta los mayores motores de búsqueda, como Excite, Infoseek, Lycos, y Yahoo!, serían incapaces de proporcionar toda la información. (Algunos estudios indican que hasta estos motores encuentran menos de la mitad de la información que existe en realidad.)

La disponibilidad de los motores de búsqueda varía de un país a otro y es probable que no todos estén disponibles en su país. Aún así, he aquí una rápida introducción a algunos de los principales motores de búsqueda que podría utilizar:

- **Excite.** Un motor de búsqueda híbrido, cumple la definición tradicional del motor de búsqueda (continuamente está visitando sitios Web y los cataloga mediante un programa), pero también tiene un directorio, una subsección de los sitios que tiene catalogados, que puede ser revisada por los usuarios.
- **Infoseek.** El motor de búsqueda de Infoseek incluye un amplio catálogo de sitios. También cuenta con un directorio que cataloga los sitios por temas mediante software. Algunos sitios han sido revisados y recomendados.
- **Lycos.** Al igual que Excite e Infoseek, Lycos es un motor de búsqueda con un directorio, ordenado por temas. También puede buscar en el servicio "Top 5%" de Lycos, en el que se incluyen los sitios que sus revisores han considerado que pertenecen al cinco por ciento de los sitios principales de Internet.
- **Yahoo!** En realidad es un directorio más que un motor de búsqueda, lo que significa que son personas los que reúnen y catalogan los sitios que busca. Por eso ofrece menos resultados, pero Yahoo! hace que sea más fácil buscar los sitios según su categoría y continuar la búsqueda mediante un motor de búsqueda, si así lo desea.
- **Metacrawler.** En lugar de mantener su propia base de datos de sitios Web, Metacrawler ejecuta su búsqueda de forma simultánea en varios motores de búsqueda, como Excite, Infoseek, Lycos y Yahoo! Es algo más lento que utilizar un único motor de búsqueda, pero la probabilidad de conseguir un conjunto de resultados más completo es mayor<sup>o</sup>.

Los principales sitios de búsqueda suelen proporcionar más de un motor para encontrar sitios Web. También permiten buscar información como las últimas historias aparecidas en las noticias, mensajes de grupos de noticias, material de referencia (como entradas de diccionarios y mapas), direcciones postales y de correo electrónico y números de teléfono personales y de empresas.

He aquí un par de consejos para elegir un motor de búsqueda:

- Si busca un tema amplio y habitual, como la forma de adquirir un equipo personal, utilice un motor de búsqueda que proporcione menos resultados pero de mayor calidad (como suelen ser aquellos que dependen de las personas para catalogar los sitios), como Yahoo!

- Si busca un tema extraño, como los autores húngaros del siglo XIX, utilice un sitio que proporcione más resultados, aunque sean menos precisos (como suelen ser los compilados mediante software), como Excite.

## **Deja News: cómo buscar en los grupos de noticias**

Los grupos de noticias de Usenet son grupos de discusión de Internet que tratan prácticamente cualquier tema que pueda imaginar. (Existen más de 50.000 grupos de noticias y se agregan otros constantemente.) ¿Sufre migrañas? Visite el grupo de noticias alt.support.headaches.migraine para encontrar ayuda. ¿Es propietario de perros de caza? Visite rec.hunting.dogs. He aquí algunos ejemplos más de grupos de noticias que le permitirán hacerse una idea de la variedad existente:

- sci.physics.fusion
- soc.genealogy.hispanic
- biz.jobs.offered
- rec.arts.animation
- alt.autos.antique

Y, por supuesto, existe un buen número de grupos relacionados con la informática. El motor de búsqueda Deja News permite realizar búsquedas en millones de mensajes de los grupos de noticias para encontrar un tema concreto que le pueda interesar. También permite publicar un mensaje como respuesta a los leídos. El sitio de Deja News también es un buen lugar para aprender más cosas acerca de los grupos de noticias en general.

## **Sugerencias para las búsquedas**

Independientemente del motor de búsqueda que utilice, estas sugerencias le ayudarán a encontrar lo que busca.

### **Utilice más de un motor de búsqueda.**

Ningún motor de búsqueda mantiene un índice de todos los sitios y páginas Web. Por eso, si su primera búsqueda no produce los resultados deseados, pruebe al menos con otro motor de búsqueda.

### **Lea la página "Acerca de".**

Muchos motores de búsqueda disponen de un vínculo que lleva a información detallada acerca de la forma en la que compilan y buscan la información y cómo obtener los mejores resultados de ella. Si lee esta página puede ahorrar mucho tiempo y dolores de cabeza. Visite también los vínculos de la página principal del motor de búsqueda que tengan nombres parecidos a "How To," "Search Help" y "Advanced Search" para obtener sugerencias.

### **Obtendrá los resultados buscados si es específico.**

Cuanto más específica sea la palabra de búsqueda, más precisos serán los

resultados de la búsqueda. Si busca las palabras "perro Labrador", por ejemplo, en lugar de sólo "perro", el resultado incluirá menos sitios, pero tratarán acerca del tipo de perro que le interesa.

### **Obtendrá más resultados si es genérico.**

Si busca una palabra demasiado específica puede obtener pocos resultados, o quizá ninguno. Para obtener más resultados busque una palabra relacionada que sea más genérica.

### **Defina la búsqueda mediante operadores booleanos.**

Los principales motores de búsqueda permiten elegir si se quiere buscar exactamente la misma frase que se ha escrito, todas las palabras que aparecen en ella, pero no necesariamente juntas, o cualquier palabra de la frase, entre otras opciones. Unos pocos motores de búsqueda no proporcionan esas opciones. En este caso, puede definir la búsqueda agregando una o más palabras o símbolos al criterio de búsqueda.

- **AND.** Para buscar dos (o más) términos en la misma página: escriba la palabra AND entre los términos. (Por ejemplo: mar AND kayak.) O incluya un signo más justo antes del segundo término. (Por ejemplo: mar +kayac.)
- **OR.** Para buscar alguno de los dos (o más) términos en la misma página: escriba la palabra OR entre los términos. (Por ejemplo: kayak OR canoa.)
- **AND NOT.** Para buscar páginas que incluyan el primer término, pero no el segundo: Escriba las palabras AND NOT entre los términos. (Por ejemplo: kayak AND NOT rápidos.) O coloque un signo menos justo antes del segundo término. (Por ejemplo: kayak -rápidos.)
- **"".** Para buscar una frase exacta: encierre la frase entre comillas. (Por ejemplo: "alquiler de kayacs para el mar".)
- **()**. Para agrupar partes de la búsqueda: enciérrelas entre paréntesis. (Por ejemplo: kayak AND (equipo OR equipamiento). Así obtendrá las páginas en las que aparezcan las palabras kayak y equipo o en las que aparezcan kayak y equipamiento.)
- **\***. Para buscar las diferentes formas de una palabra: agregue un asterisco al final de la palabra. (Por ejemplo: viaj\*. Así obtendrá las páginas que incluyan las palabras viaje, viajar, viajero, viajante, etcétera.)

### **Buscar con Microsoft® Internet Explorer versión 4.0**

#### **Buscar en Internet con un motor de búsqueda e Internet Explorer 4.0**

6. En la barra de herramientas, haga clic en el botón **Búsqueda**.
7. En la barra de búsqueda que aparece en el panel de la izquierda de la ventana de Internet Explorer, seleccione el motor de búsqueda que prefiera en la lista desplegable llamada **Seleccionar proveedor**.
8. Según el motor de búsqueda que haya seleccionado tendrá diferentes opciones para configurar la búsqueda. Haga sus elecciones o acepte las que aparecen predeterminadas.

9. En el cuadro de texto correspondiente, escriba la palabra o frase que desea buscar y haga clic en **Search**, **Submit** o **Go Get It**, según el motor de búsqueda.
10. Los resultados aparecerán en la lista de la barra de búsqueda. Haga clic en cualquier vínculo para abrir la página Web asociada en el panel de la derecha de la ventana de Internet Explorer.
11. Cuando quiera examinar otro de los vínculos que aparecen en los resultados de la búsqueda, haga clic en él. La página Web se abrirá en el panel de la derecha.
12. Para ocultar la barra de búsqueda, haga clic de nuevo en el botón **Búsqueda** de la barra de herramientas.

#### **Buscar en Internet desde la barra de direcciones de Internet Explorer 4.0**

5. En la barra de direcciones, escriba **go**, **find** o **?** seguidos por un espacio y la palabra o frase que desea encontrar. Por ejemplo, escriba *Go perro Labrador*.
6. En la lista de resultados de la búsqueda, haga clic en un vínculo para mostrar la página Web.

Nota: la característica AutoSearch sólo utiliza un servicio de búsqueda para buscar la información.

#### **Buscar texto en la página Web actual con Internet Explorer 4.0**

3. En el menú Edición, haga clic en **Buscar** en esta página.
4. Escriba el texto que desea encontrar.
5. Cambie las opciones necesarias.
6. Haga clic en **Buscar siguiente**.

#### **Buscar un sitio Web cuando una dirección URL no funciona mediante Internet Explorer 4.0**

Si la dirección URL, o dirección Web, que escribe en la barra de direcciones no funciona, pruebe este método para corregirla:

4. En el menú Ver, haga clic en **Opciones de Internet**.
5. Haga clic en la ficha **Opciones avanzadas**.
6. Vaya a Búsqueda. Cuando la dirección URL falle, en la opción Buscar:
  - Si desea que Internet Explorer le pregunte si desea buscar una dirección similar, haga clic en **Preguntar siempre**.
  - Si desea que Internet Explorer realice la búsqueda sin preguntar, haga clic en **Buscar siempre**.
  - Si no desea que Internet Explorer busque una dirección similar, haga clic en **No buscar nunca**.
7. Si desea que Internet Explorer busque la dirección con un dominio diferente, active la casilla de verificación Explorar automáticamente los dominios principales comunes, en la sección Búsqueda. (Un dominio viene determinado por una extensión de tres letras, como .org, .edu, .com y .gov, en la dirección, que indica el tipo de un sitio. Si selecciona la opción

**Explorar automáticamente los dominios principales comunes**, Internet Explorer buscará la raíz de la dirección URL con otras extensiones de dominio.)

8. Haga clic en **Aceptar**.

## **CONECTARSE**

Seleccionar un proveedor de servicios Internet

WebTV

Cibercafés

Acceso en el aeropuerto

En camino con dispositivos PC de bolsillo y PC de mano

En el automóvil con Auto PC

### **Seleccionar un proveedor de servicios Internet**

Un proveedor de servicios Internet (ISP) proporciona acceso telefónico y el software necesario para conectar con Internet, así como algo de asistencia técnica. Muchos ISP incluyen una cuenta de correo electrónico, alojan las páginas Web de sus clientes y ofrecen servicios a empresas que realizan negocios en



Internet.

### **Tenga en cuenta estos factores**

He aquí algunas cosas que hay que tener en cuenta al seleccionar un ISP:

**Marcado.** ¿El ISP o el proveedor de servicios electrónicos ofrecen sus servicios mediante un número telefónico de su misma área? ¿Ofrece un número de teléfono local o uno gratuito en las áreas a las que espera viajar?

**Acceso.** ¿Con qué frecuencia consigue conectar con su ISP, en lugar de obtener la señal de línea ocupada? (No confíe en un representante del servicio de atención al cliente para obtener esta información. Pruebe a llamar varias veces al número de conexión en diferentes horas del día y vea cuál es el resultado.)

**Soporte técnico y atención al cliente.** Hay disponible personal de soporte técnico y atención al cliente con conocimientos y de buen trato siempre que los necesita? ¿El número de teléfono es gratuito? ¿Cuánto tiempo hay que estar a la espera? ¿Puede ponerse en contacto electrónicamente con el servicio técnico y el departamento de atención al cliente?

**Correo electrónico.** ¿El servicio ofrece una cuenta de correo electrónico y se tiene acceso a ella a través de otro ISP si no se encuentra en el área local de su ISP?



**Páginas Web.** ¿Puede publicar un sitio Web en el servidor del ISP? ¿De cuánto espacio dispone y a qué precio? ¿Le cobrarán según el tráfico generado por su página? ¿El servidor permite utilizar CGI (necesario si incluye un formulario)?

**Rendimiento.** ¿Con qué rapidez entrega el servicio su correo electrónico? ¿Cuánto tiempo es necesario para descargar las páginas Web? ¿Con qué frecuencia se producen errores y cuánto duran?

**Actualizaciones.** ¿Puede actualizar su módem por uno más rápido?

**Precio.** Muchos servicios ofrecen diferentes tarifas mensuales, desde acceso ilimitado por una tarifa mensual hasta horas de acceso limitadas por una tarifa mensual con las horas adicionales necesarias con un pago adicional. Cuando compare los precios de los ISP y los proveedores de servicios electrónicos, tenga en cuenta los factores anteriores y las características especiales y el contenido que ofrecen los proveedores de servicios electrónicos.



## **Asistente para la conexión a Internet**

El sistema operativo Microsoft® Windows® 98 incluye un nuevo Asistente para la conexión a Internet, que facilita la configuración de la primera conexión con su ISP o proveedor de servicios electrónicos. Incluye una lista de ISP y proveedores de servicios electrónicos por países para que pueda elegir. El asistente configura el sistema automáticamente para conectarse a Internet con el ISP que haya seleccionado.

## **WebTV**

Al conectar el sistema WebTV® Plus a su televisor normal puede explorar Internet, ver los programas de televisión normales, hacer ambas cosas al mismo tiempo o ver programas mejorados en su propio televisor. (Los programas de televisión mejorados le permiten obtener más información acerca del programa que está viendo.)

Al unir Internet y la programación de televisión, WebTV abre paso a nuevos mundos de entretenimiento y aprendizaje. Sólo hay que colocar el receptor encima del televisor y conectar ambos. El receptor incluye un motor de gráficos 3D, un sintonizador estéreo 3 en 1, un disco duro interno de 1,1 gigabytes y la posibilidad de imprimir el contenido que encuentre en Internet. Incluso puede guardar la información y verla más adelante. Además, es fácil pasar de un sitio Web a la televisión y viceversa.

## **Cibercafés**

Si no tiene acceso habitual a un equipo en casa o si quiere ser sociable, visite un cibercafé, o café Internet. Estos establecimientos, que proporcionan (por un pago) un equipo y acceso a Internet a sus clientes, así como bebida y comida, son cada vez más habituales en las principales ciudades de todo el mundo.

## **Acceso en el aeropuerto**

¿El vuelo sale con retraso? ¿Por qué no comprueba si tiene correo electrónico? Algunas líneas aéreas y aeropuertos proporcionan quioscos de Internet (estaciones informáticas desde las que se puede tener acceso a Internet) o lugares en los que puede conectar su equipo portátil para el acceso a Internet. Consulte con las líneas aéreas y aeropuertos para averiguar si ofrecen estos servicios y qué hace falta para utilizarlos.

### **En camino con PC de bolsillo y PC de mano**

Los dispositivos PC de bolsillo, diseñados para caber en el bolsillo de una chaqueta o en un bolso y los dispositivos PC de mano, del tamaño de la palma de la mano, le permiten recoger el correo electrónico donde quiera que vaya. El sistema operativo Microsoft Windows CE alimenta ambos tipos de equipos portátiles. Con su PC de bolsillo, puede marcar y conectar con un servidor cuando no esté en su equipo de escritorio, de forma que pueda recibir el correo electrónico y recorrer el World Wide Web. Con su PC de mano puede descargar el correo electrónico y el contenido de Internet desde su escritorio para leerlo cuando le sea más cómodo.

### **En el automóvil con Auto PC**

Con Auto PC, alimentado por el sistema operativo Microsoft® Windows® CE, tiene acceso a Internet incluso cuando está en la autopista. Mediante sencillos comandos de voz puede escuchar sus mensajes de correo electrónico y recibir sin cables el contenido de Internet para escucharlo mientras conduce.

---

## ***EL CORREO ELECTRÓNICO***

El correo electrónico (e-mail) proporciona un método rápido y barato para enviar información a cualquier lugar del mundo y para recibir información en su propio buzón de correo electrónico, o "Inbox". Puede sustituir al teléfono, fax, mensajeros y al correo tradicional. Utilícelo para mantener el contacto con sus amigos, familiares y contactos comerciales. Más de 80 millones de personas ya lo hacen en todo el mundo. Esto es lo que tiene que hacer.

[Fundamentos del envío y la recepción](#)

[¿Qué sucede con los datos adjuntos?](#)

[Aprovechar los grupos de noticias](#)

## **Fundamentos del envío y la recepción**

Para enviar y recibir mensajes de correo electrónico, o e-mail, a través de Internet, y para organizar sus mensajes, necesita una cuenta de correo electrónico. Puede obtenerla mediante un proveedor de servicios Internet o un proveedor de servicios electrónicos. También necesita un cliente de correo electrónico, el software de correo electrónico para su PC. Su cliente de correo electrónico crea un buzón en su equipo, en el que se descargan los nuevos mensajes.

## Escoja un cliente de correo electrónico

He aquí algunos ejemplos:

**Microsoft® Outlook™ 98:** un cliente de mensajería y colaboración, proporciona correo electrónico avanzado, calendario y una agenda de contactos, así como una plataforma para compartir información en grupos de trabajo cuando se utiliza con Microsoft Exchange Server. Puede descargar gratuitamente Outlook 98 desde el sitio Web de Microsoft Office.

**Hotmail:** el principal proveedor mundial de correo electrónico gratuito (está financiado mediante publicidad). Como está basado en Web, todo lo que necesita para utilizarlo se halla en Internet, en lugar de estar en su sistema. Así, puede utilizar Hotmail en cualquier PC que tenga acceso a Internet. Si viaja, se muda o cambia de proveedor de acceso, su dirección de correo electrónico de Hotmail y el servicio que le presta serán siempre los mismos.

**Microsoft Outlook Express:** diseñado para usuarios que necesitan la funcionalidad, rapidez y confiabilidad del correo electrónico y de los grupos de noticias, sin llegar a requerir las demás capacidades de Outlook. Outlook Express se incluye en el sistema operativo Microsoft Windows 98 y resulta especialmente fácil de utilizar si cuenta con la versión 4.0 del explorador Web Microsoft Internet Explorer.

**Microsoft Exchange:** incluido automáticamente en el sistema operativo Microsoft Windows 95

## Un poco de terminología

Existen muchos acrónimos asociados con el correo electrónico. A continuación se muestran algunos de los que puede encontrar:

**SMTP:** protocolo simple de transferencia de correo, las reglas estándar que utilizan muchos clientes de correo electrónico para controlar los mensajes salientes.

**POP3:** versión 3 del protocolo de oficina de correo, las reglas estándar que utilizan muchos clientes de correo electrónico para controlar los mensajes entrantes.

Los clientes de correo electrónico basados en estándares como SMTP y POP3 pueden compartir información entre ellos, por lo que no es necesario que todo el mundo utilice el mismo cliente. Para configurar su cliente de correo electrónico puede necesitar las direcciones de sus servidores SMTP y POP3. Si es así, su proveedor de servicio Internet (ISP) puede proporcionarle esta información. Algunos clientes de correo electrónico utilizan programas llamados Asistentes para facilitar la configuración.

**MIME:** extensiones multipropósito de correo Internet, un formato para convertir datos adjuntos de correo electrónico, por ejemplo un archivo de Microsoft Word, en texto ASCII para poder enviarlo de una cuenta de correo electrónico a otra.

## Anatomía de una dirección de correo electrónico

Para enviar un mensaje a alguien debe escribir la dirección de correo electrónico en el campo "Para" del mensaje. Generalmente, una dirección de correo electrónico consta de dos partes: su identidad de inicio de sesión y la identidad del ISP. Las dos partes están separadas por el símbolo @. Una dirección de correo

electrónico para alguien que, por ejemplo, utilice Hotmail tiene esta apariencia: sunombre@hotmail.com

La extensión .com indica que MSN es un establecimiento comercial. Otras extensiones habituales son .gov (para usuarios del gobierno) y .edu (para usuarios de instituciones educativas). Las direcciones de correo electrónico de otros países pueden incluir letras que indiquen el país al que pertenece el ISP.

### ¿Qué sucede con los datos adjuntos?

En los mensajes de correo electrónico se puede enviar algo más que texto. También se puede adjuntar a un mensaje de correo electrónico documentos informáticos, como archivos de procesadores de textos, hojas de cálculo, imágenes y vídeo. Por tanto, el correo electrónico es un método útil para compartir archivos que deseen o necesiten dos o más personas.

Con Microsoft® Outlook™ 98, el correo electrónico se hace aún más sofisticado. Outlook 98 permite enviar mensajes con "diseño de fondo" HTML, que contiene imágenes de fondo, fuentes especiales y colores preseleccionados para mantener la uniformidad.



### Notas especiales

- La persona que recibe un mensaje de correo electrónico con datos adjuntos podrá abrirlo si dispone del programa con el que se creó esos datos adjuntos.
- Si el destinatario no dispone de tal programa, quizás pueda obtener el visor apropiado, como el visor de Microsoft® Word o el de Microsoft PowerPoint®. Un visor contiene los componentes que necesita un programa para mostrar un archivo creado con la versión completa del programa.
- Para enviar datos adjuntos el cliente de correo electrónico convierte su código digital en texto ASCII, para lo que utiliza un formato como uuencode o MIME (extensiones multipropósito de correo Internet). Para que el destinatario pueda leer los datos adjuntos, su cliente de correo electrónico debe ser capaz de descifrar ese formato o debe disponer de otro programa que pueda hacerlo.

### Aprovechar los grupos de noticias

Si su cliente de correo electrónico es también un lector de noticias puede utilizarlo para realizar una suscripción a, leer mensajes de y publicar mensajes en grupos de noticias, discusiones electrónicas acerca de prácticamente cualquier tema. La versión 4.0 de Microsoft® Internet Explorer también permite un fácil acceso a los grupos de noticias. En el menú Ir a, seleccione Noticias. Internet Explorer ejecutará automáticamente Outlook Express y abrirá la carpeta de grupos de noticias para que pueda empezar a trabajar.

He aquí algunos fundamentos acerca de los grupos de noticias:

- Cada grupo de noticias trata de un tema concreto, al que normalmente hace referencia su nombre.
- Un grupo de noticias es en realidad una cadena de discusión formada por una serie de mensajes relacionados. Cada mensaje responde a otro anterior o trata de algún modo el tema general del grupo de noticias. Puede publicar sus propios mensajes como respuesta a aquellos que más le interesen.
- Juntos, los grupos de noticias conforman Usenet, que puede considerarse como parte de Internet.

Antes de realizar una suscripción:

- Lea el archivo FAQ, o preguntas más frecuentes (P+F), de un grupo de noticias para obtener más información acerca del mismo. (La mayoría de los grupos de noticias cuentan con un archivo de este tipo.)
- Compruebe si el grupo de noticias tiene un archivo en el que se describan su carácter y sus reglas. Si es así, léalo antes de exponer sus mensajes.
- Lea buena parte de una cadena de discusión antes de unirse a ella. Los demás participantes agradecerán que haya dedicado algo de tiempo a conocer el tema de la conversación.

## **BUSCAR EN INTERNET**

Internet es una red mundial de sistemas informáticos que permite enviar y recibir información de forma casi instantánea. Se sorprenderá por lo que puede encontrar en Internet con sólo explorar un poco. Busque lo que busque, es probable que alguien haya pensado en proporcionar eso mismo a los demás. He aquí una guía del contenido de Internet.

[¿Qué hay en el Web?](#)

[¿Internet y el Web son lo mismo?](#)

[¿Qué son los grupos de noticias?](#)

[¿Qué es un FAQ \(P+F\)?](#)

[¿Qué son Usenet, FTP y gopher](#)

### **¿Qué hay en el Web?**

El World Wide Web alberga millones de sitios Web, creados por empresas, agencias, instituciones, personas, etcétera. Piense en algún tema y probablemente estará en alguna parte de la Web. ¿Alguna vez ha oído hablar de un narval? Es una ballena que tiene un gran cuerno en su frente y vive en el Océano Ártico y, sí, es el tema de múltiples sitios en el Web. He aquí una guía para encontrar información un poco más común en el Web.



### **Compras**

Salga y compre instantáneamente en el Web en los sitios de empresas o en los creados como centros comerciales, que conectan cientos de "tiendas". Estos sitios

permiten recorrer los productos y servicios e incluso pedirlos o pagarlos electrónicamente.



### **Banca e inversiones**

Internet está llena de información y servicios relacionados con la banca y las inversiones, desde consejos generales para planear la jubilación hasta las cotizaciones de la bolsa actualizadas minuto a minuto. Además, cada vez más bancos ofrecen a sus clientes acceso electrónico a sus cuentas. También puede manejar electrónicamente su propia cartera de valores.

- En el sitio Web de [Microsoft® Investor](#), podrá leer artículos de fondo acerca de inversiones, así como investigar, seguir, encontrar y comerciar con acciones y fondos de inversión y obtener un informe de mercado actualizado tres veces al día.
- Visite el sitio Web de [Microsoft Money Insider™](#) para obtener consejos acerca de casi todos los aspectos de la administración del dinero, como ahorros, impuestos, adquisición de viviendas y planes de jubilación. El sitio ofrece artículos de información, herramientas prácticas, recursos útiles y mucho más.
- Conozca el software de administración personal Microsoft Money 98 Financial Suite y el sitio Web de [Money 98](#). Money 98 Financial Suite incluye herramientas para crear planes y realizar seguimientos, consejos personalizados e información electrónica integrados en un único producto.

### **Noticias**

No es necesario esperar al periódico de la mañana para saber lo que sucede en el mundo. Muchos sitios Web dedicados a las noticias se actualizan varias veces a lo largo del día, así que puede obtener los últimos sucesos del mundo, deportes, información meteorológica, etcétera siempre que lo desee.

### **Referencia**

Internet permite tener acceso a muchos materiales de referencia, como las colecciones de muchas bibliotecas, centros de investigación, museos y otras instituciones semejantes. Puede encontrar información tan sencilla como la definición de una palabra o tan compleja como los últimos avances en investigación científica.

- [Microsoft Encarta® Online](#) es una enciclopedia interactiva repleta de información acerca de las culturas, sucesos, descubrimientos y muchos más aspectos de todo el mundo. Desde la página principal de Encarta puede buscar cualquier tema.

### **Varios**

Parte del interés y el valor del Web es la facilidad, no sólo para grandes empresas y organizaciones, si no también para entidades más pequeñas o individuos, para mantener sus propios sitios Web. Por eso todo el mundo encuentra algo interesante en el Web.

Compras, banca e inversión, noticias y materiales de referencia son sólo la punta del iceberg. También existen muchos recursos relacionados con el entretenimiento y la diversión, temas poco comunes e intereses minoritarios. Para hacerse una idea acerca de lo que hay ahí fuera, utilice uno o más motores de búsqueda para explorar el Web. Escriba un tema, vea lo que aparece en pantalla y vaya de un sitio a otro.

### **¿Internet y la Web son lo mismo?**

Internet y el World Wide Web (que a veces se abrevia como W3) están muy relacionados, pero no son lo mismo.

- Internet es una red global de equipos descentralizada.
- El Web es una colección de documentos, o sitios Web, a los que se puede tener acceso a través de Internet y de un explorador Web. El Web contiene la inmensa mayoría del contenido disponible en Internet (pero no todo).

### **¿Cómo funciona?**

Los sitios Web residen en sistemas denominados servidores de Internet. Cuando conecta con Internet, su explorador Web puede comunicarse con estos servidores y pedirles que envíen a su PC una copia de la página Web que desea ver. La URL que escribe o el hipervínculo en el que hace clic indica a su equipo el servidor con el que tiene que ponerse en contacto y la página que tiene que solicitar.

### **¿Qué es TCP/IP?**

Los sistemas interconectados de Internet son de diferentes tipos y utilizan diferentes sistemas operativos. Para poder comunicarse entre ellos a través de Internet utilizan un conjunto común de reglas llamado Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP) o simplemente Protocolo de Internet (IP). El software TCP/IP también permite conectar su equipo con Internet.



### **¿Qué hay en Internet aparte del Web?**

A través de Internet se puede tener acceso a muchos tipos de información. He aquí algunas de las cosas, además de visitar los sitios Web, que se pueden hacer mediante Internet:

- Enviar y recibir mensajes de correo electrónico. (Consulte el capítulo [El correo electrónico](#) de la Guía de Internet.)
- Realizar suscripción a un grupo de Usenet.
- Transferir documentos y programas entre su PC y un sitio FTP, o Protocolo de Transferencia de Archivos.
- Utilizar un "gopher" para buscar información acerca de otros sitios.

### **¿Qué son los grupos de noticias?**

Los grupos de noticias de Usenet son grupos de discusión de Internet que tratan prácticamente de cualquier tema que pueda imaginar. (Existen más de 50,000 grupos de noticias y se agregan más constantemente.) Cada grupo de noticias trata de un tema concreto, al que a menudo hace referencia con su nombre.

Un grupo de noticias crea una cadena de discusión, o una serie de mensajes relacionados. Cada mensaje responde a uno anterior o trata de algún modo el tema general del grupo. Puede enviar sus propios mensajes como respuesta a los mensajes que más le interesen.



### **Miles de temas**

¿Sufre migrañas? Visite el grupo de noticias alt.support.headaches.migraine para encontrar ayuda. ¿Es propietario de perros de caza? Visite rec.hunting.dogs. He aquí algunos ejemplos más de grupos de noticias que le permitirán hacerse una idea de la variedad existente:

- sci.physics.fusion
- soc.genealogy.hispanic
- biz.jobs.offered
- rec.arts.animation
- alt.autos.antique

Y, por supuesto, existe gran cantidad de grupos relacionados con la informática.



### **Cómo hacerlo**

Si su cliente de correo electrónico también puede leer grupos de noticias, también puede utilizarlo para suscribirse a, leer mensajes y enviarlos a los grupos de noticias. [Microsoft® Internet Explorer versión 4.0](#) facilita el acceso a los grupos de noticias. En el menú Ir a, seleccione Noticias. Internet Explorer ejecutará automáticamente Outlook Express™ y abrirá su carpeta de grupos de noticias para que pueda empezar a trabajar.



### **Empiece aquí**

Antes de unirse a la discusión de un grupo de noticias:

- Lea el archivo FAQ, o preguntas más frecuentes (P+F), de un grupo de noticias para saber más acerca del mismo. (Casi todos los grupos cuentan con un archivo de este tipo.)
- Compruebe si el grupo de noticias tiene un archivo en el que se describan su carácter y sus reglas. Si es así, léalo antes de enviar sus mensajes.
- Lea buena parte de una cadena de discusión antes de unirse a ella. Los demás participantes agradecerán que haya dedicado algo de tiempo a conocer el tema de la conversación.

### **¿Qué es un FAQ (P+F)?**

FAQ significa "preguntas más frecuentes" (P+F). Muchos sitios de la World Wide Web incluyen una página FAQ, que responde a las preguntas más frecuentes acerca del mismo, como su tema, cómo encontrar información en él y quién la ha creado. Puede considerar la Guía de Internet de Informática personal de Microsoft como una página FAQ, o algo parecido, de Internet. Normalmente la página FAQ será un documento de texto que podrá descargar, y no un sitio Web. Cuando visite



un grupo de noticias por primera vez, consulte su FAQ. Probablemente responderá todas las preguntas que le surjan acerca del grupo y le ayudará a aprovecharlo al máximo.

### **¿Qué son Usenet, FTP y gopher?**

Usenet, FTP y gopher representan otras formas de acceso a la información de Internet diferentes a los sitios Web.

- **Usenet:** colección de varios miles de grupo de noticias que tratan prácticamente cualquier tema. Con un lector de noticias puede realizar una suscripción a un grupo, leer los mensajes enviados por otros suscriptores y enviar sus propios mensajes como respuesta. Los grupos de noticias proporcionan un método para que los usuarios compartan información acerca de los temas que les interesan.
- **FTP:** significa "Protocolo de transferencia de archivos", un sistema de reglas y un programa que permiten tener acceso a otro equipo y transferir información de uno a otro. La mayoría de los exploradores incluyen algún tipo de capacidad FTP. Puede encontrar muchos sitios FTP con un motor de búsqueda de Internet o utilizar el motor de búsqueda Archie, que contiene un índice de sitios FTP.
- **Gopher:** sencillo sistema de menús que puede utilizar para conectarse con archivos de Internet. Este sistema fácil de utilizar fue la puerta de acceso a Internet para muchos usuarios antes de que se hiciera popular el World Wide Web. Mediante el gopher de la Universidad de Minnesota, que fue el primero, puede conectar con los restantes servicios gopher de todo el mundo. Puede buscar servidores gopher con un programa llamado Veronica.

### **SEGURIDAD EN INTERNET**

Internet puede parecer un lugar por el que los usuarios vagan de forma anónima y privada, especialmente si tienen acceso desde su propia casa, Sin embargo, cada vez que alguien se adentra en Internet, su PC intercambia información de forma activa con los demás sistemas. Por tanto, ¿qué grado de seguridad y privacidad tienen esas comunicaciones? Eso dependerá principalmente de los sitios que visite y de las características de seguridad de su explorador Web.

[Seguridad con Microsoft® Internet Explorer 4.0](#)

[Compras](#)

[Banca e inversiones](#)

[Virus](#)

[Cookies](#)

[Privacidad](#)

[ActiveX®, Java y certificados](#)

**Seguridad con Microsoft Internet Explorer 4.0**

[Microsoft® Internet Explorer versión 4.0](#) permite aprovechar al máximo la eficaz tecnología de los sitios Web de confianza, al tiempo que reduce el riesgo de sufrir daños en los sitios de los que no se tiene mucha información.

### **Zonas de seguridad**

Puede crearlas si asigna sitios Web a una de las cuatro zonas de seguridad (zona intranet local, zona de sitios en los que se confía, zona Internet o zona de sitios restringidos) y si determina el nivel de seguridad que desea para cada zona. Seleccione seguridad alta, media o baja o personalice docenas de aspectos relacionados con la seguridad de cada zona.



### **Más características**

Las características de seguridad de Internet Explorer 4.0 también permiten:

- Bloquear los sitios Web que contengan lenguaje, desnudez, sexo o violencia que considere inadecuados para los usuarios de su PC.
- Utilizar la tecnología Authenticode™ para comprobar si el software es seguro antes de descargarlo.
- Habilitar los subprogramas Java (miniaplicaciones que pueden mejorar la experiencia en el Web) para los sitios en los que confíe y deshabilitarlas para el resto.

### **Compras**

¿Con qué frecuencia entrega su tarjeta de crédito al camarero de un restaurante o dice su número de cuenta por teléfono al solicitar un producto? Este tipo de acciones probablemente suponen un riesgo mayor que adquirir productos en línea,



al menos en los sitios Web en los que confía.

### **Certificados**

Si duda si debe proporcionar el número de su tarjeta de crédito u otra información personal al sitio Web de una compañía, compruebe el certificado del sitio. (Con Microsoft® Internet Explorer versión 4.0, en el menú Archivo, haga clic en Propiedades y después en Certificados.) El certificado de un sitio Web es un documento electrónico que certifica su identidad para que pueda estar seguro de que la información va a donde se supone que debe ir.

### **Sopesar el riesgo**

Existe la posibilidad de que un ladrón pueda interceptar el número de su tarjeta de crédito mientras viaja de su sistema al servidor del sitio Web, pero es muy pequeña. De hecho, es mucho más difícil en el mundo electrónico que en el mundo real, en el que el número de su tarjeta de crédito aparece impreso en recibos y estados de cuenta que son enviados por correo, entregados o tirados a una papelería. Algunos sitios pueden colaborar con su explorador para codificar la información de su transacción, de forma que no pueda ser leída si es interceptada.

## Banca e inversiones

Los bancos electrónicos y los servicios de inversión utilizan la codificación para proteger la información de sus transacciones. Antes de que la información deje el servidor del sitio Web para ir a su equipo, y viceversa, es convertida en código. Cuando llega a su destino es descodificada. Aunque la información viaja por Internet -- donde podría ser vulnerable y ser interceptada por alguien con malas intenciones -- ésta es esencialmente incomprensible. Por tanto, si alguien interceptara su información mientras se transfiere entre el banco y su sistema, sería incapaz de leerla. Mientras mantenga en secreto su contraseña de banca electrónica, nadie más tendrá acceso a su información financiera. Solicite a su banco información acerca de las medidas que toma para mantener la seguridad de las transacciones.



## Dos niveles de codificación

Los servidores y los exploradores Web utilizan codificación de 40 o 128 bits. Con la codificación de 40 bits, existen muchos miles de millones de claves posibles para desbloquear el código para cada transmisión única y sólo funciona una de ellas. Con la codificación de 128 bits existen trescientos mil millones de claves posibles más que con la codificación de 40 bits.



## Cómo comprobar la codificación

Microsoft® Internet Explorer versión 4.0 indica si se utiliza la codificación mediante un candado que aparece en la parte inferior de la ventana de Internet Explorer. Así puede conocer el nivel de codificación de una página Web con Internet Explorer 4.0:

13. Haga clic con el botón secundario del mouse en la página, seleccione **Propiedades** y después **Certificados**.
14. En el cuadro Campos, seleccione **Tipo de codificación**. En el cuadro Detalles aparecerá información acerca de la codificación de la página.
15. Haga clic en **Cerrar** y después en **Aceptar**.



Puede aumentar al máximo la seguridad de sus transacciones con su explorador Web. Con Internet Explorer 4.0, por ejemplo, puede:

- Buscar el icono de un candado en la barra que aparece en la parte inferior del explorador. Aparece mientras está conectado con un sitio seguro (un sitio que utilice codificación).

- Configure al máximo el nivel de seguridad de su explorador. En el menú Ver, haga clic en Opciones de Internet. En la ficha Seguridad, seleccione Alta y haga clic en Aceptar.
- Seleccione cada una de las opciones individuales del explorador que aumentan la seguridad. Por ejemplo, en el menú Ver, haga clic en Opciones de Internet. Seleccione la ficha Opciones avanzadas, vaya a la sección Seguridad, active "Avisar del cambio entre modalidad segura y no segura" y haga clic en Aceptar.

Para obtener más detalles acerca de las opciones de seguridad de su explorador, consulte sus archivos de Ayuda.

## **Virus**

Para protegerse frente a los virus en Internet, no descargue archivos de lugares que no sepa que son seguros. Los virus suelen estar ocultos en los programas y se activan al ejecutarlos. También pueden aparecer como archivos adjuntos de otros tipos de archivos ejecutables, como archivos Web que realicen acciones especiales y como archivos de vídeo.



### **Haga caso de la advertencia**

Normalmente, cuando está a punto de descargar un tipo de archivo que podría contener un virus, el explorador muestra una advertencia y pregunta si desea abrir el archivo o guardarlo en disco. Si está seguro de que los archivos provienen de un lugar de confianza, puede abrirlo sin preocupaciones. Si no está seguro, lo mejor sería cancelar la descarga.



### **Compruebe el certificado**

Un método para comprobar la identidad de un sitio es mirar si dispone de un certificado Web en el que aparezca el nombre del creador del sitio. Con Microsoft® Internet Explorer versión 4.0, puede comprobar el certificado de una página Web si hace clic en **Propiedades** y después en **Certificados**.



### **Utilice un programa antivirus**

Existen muchos programas antivirus que pueden buscar virus en su sistema, protegerlo contra los virus conocidos y puede que incluso reparar los daños causados por uno de ellos. Para aprovechar las ventajas de estos programas, asegúrese de que los ejecuta tal y como se recomienda (podría programarlo para que compruebe su sistema automáticamente a intervalos regulares) y obtiene las actualizaciones del programa, en las que se incluye información acerca de los nuevos virus.

### **Cookies**

Un cookie es una pequeña cantidad de información que un sitio Web almacena en su sistema, información que devuelve su explorador cada vez que visita el sitio de nuevo. Normalmente, el cookie está diseñado para recordar al sitio la información

del usuario, como su contraseña o el color de fondo personalizado, por lo que la exploración se hace más sencilla.



### **No suelen ser perjudiciales**

Los Cookies son habituales y normalmente no son perjudiciales. No pueden utilizarse para recopilar información de su sistema sin que la proporcione el usuario. Pero ciertos servicios pueden utilizarlas para crear un perfil de sus intereses según los sitios que visite. Así se puede personalizar la publicidad para cada usuario en los sitios que utilicen este sistema.



### **Controlar los cookies**

Con Microsoft® Internet Explorer versión 4.0 puede decidir si desea que se le avise cada vez que un servidor intente entregarle un cookie:

16. En el menú Ver, haga clic en **Opciones de Internet** y seleccione la ficha **Opciones avanzadas**.

17. Vaya a Seguridad.

18. En Cookies, seleccione **Preguntar antes de aceptar cookies** y haga clic en **Aceptar**.

19. Cuando reciba una advertencia podrá rechazar el cookie.

Puede decidir que desea prohibir todos los cookies:

7. En el menú Ver, haga clic en **Opciones de Internet** y seleccione la ficha **Opciones avanzadas**.

8. Vaya a Seguridad.

9. En Cookies, seleccione **Desactivar el uso de cualquier cookie** y haga clic en **Aceptar**.

Para eliminar los cookies de su sistema, mediante el Explorador de Windows®, vaya a C:\Windows\Cookies y elimine todos los archivos que halle.

### **Privacidad**

Cualquier sitio Web que visite podrá decirle quién es y dónde se encuentra su proveedor de servicios Internet, el último sitio que ha visitado, el explorador que utiliza y qué hace mientras visita ese sitio. Mediante una petición de registro, un sitio puede recoger información adicional, como su nombre, dirección de correo electrónico, dirección postal, nivel de ingresos e intereses. En este caso es usted



el que decide si desea proporcionar esta información.

### **¿Está en la lista?**

Si su número de teléfono aparece en un directorio telefónico, es probable que su nombre, dirección y número estén en las bases de datos del World Wide Web, a disposición de los usuarios que deseen buscarlo.

### **Tenga cuidado con el correo electrónico**

Teóricamente, los encargados del correo y los administradores de sistemas que retransmiten los mensajes de correo electrónico podrían leer su correo electrónico si lo desearan. Pero se envían tantos mensajes diariamente que no es probable que lean uno en concreto. Aún así, debería saber que los empresarios tienen el derecho de leer el correo que es enviado con sus equipos, las autoridades pueden controlar su correo electrónico bajo determinadas circunstancias y los juzgados pueden exigir la entrega del correo electrónico relacionado con un caso. Es una buena idea no decir nada en el correo electrónico que no se pueda decir en público.



### **Los grupos de noticias no son privados**

Los mensajes que son enviados a los grupos de noticias de Usenet están a disposición de cualquier persona en Internet y están archivados y se pueden realizar búsquedas en ellos, por lo que no deben considerarse como privados. Además, las personas que envían correo basura a menudo obtienen las direcciones de correo electrónico de los grupos de noticias.

### **ActiveX, Java y certificados**

Incluso si no descarga intencionadamente software de un sitio Web, los elementos de un sitio pueden descargarse, ejecutarse en su sistema y presentar un riesgo de seguridad potencial, como liberar un virus en el sistema.

#### **ActiveX**

Las tecnologías ActiveX® permiten distribuir software a través de Internet. ActiveX puede presentarse en forma de controles ActiveX, normalmente elementos gráficos como marquesinas con desplazamiento, en los sitios Web. Considérelos como pequeños programas del sitio Web que se ejecutan en su sistema. Un control ActiveX puede incluir una firma digital de su creador. Por tanto, una entidad emisora de certificados como VeriSign puede certificar la firma.

Un certificado es su garantía de que el control era seguro cuando se diseñó y de que no ha sido modificado desde entonces. Puede configurar su explorador Web para habilitar, deshabilitar o le pida que decida lo que desea hacer con los controles ActiveX si están etiquetados como seguros.

Con Microsoft® Internet Explorer versión 4.0, puede desactivar ActiveX por completo de forma automática:

20. En el menú Ver, haga clic en **Opciones de internet** y seleccione la ficha **Seguridad**.

21. En la lista desplegable Zona, seleccione **Zona de Internet**.

22. En el cuadro Zona de Internet, seleccione **Alta** y haga clic en **Aceptar**.

Para que el explorador le pregunte antes de descargar y ejecutar cualquier control ActiveX, seleccione Media en el cuadro Zona de Internet.

Para configurar cada aspecto de la seguridad de los controles ActiveX:

10. En el menú Ver, haga clic en **Opciones de Internet** y seleccione la ficha **Seguridad**.

11. En la lista desplegable Zona, seleccione **Zona de Internet**.
12. En el cuadro Zona de Internet, seleccione **Personalizar** y haga clic en **Configuración**.
13. En Controles y complementos para ActiveX, seleccione la configuración que desea y haga clic en **Aceptar** y después de nuevo en **Aceptar**.

## Java

Java es un lenguaje informático. Las miniaplicaciones basadas en Java, o subprogramas, pueden ser descargadas desde sitios Web y ejecutadas por los exploradores. Normalmente, estos subprogramas tienen sus funciones limitadas. Sin embargo, existen algunos problemas relacionados con Java. Para disponer de una mayor seguridad, desactive la posibilidad de utilizar Java en su explorador, excepto para los sitios en los que confíe.

Para deshabilitar Java con Internet Explorer 4.0:

7. En el menú Ver, haga clic en **Opciones de Internet** y seleccione la ficha **Seguridad**.
8. En la lista desplegable Zona, seleccione **Zona de Internet**.
9. En el cuadro Zona de Internet, seleccione **Personalizar** y haga clic en **Configuración**.
10. Vaya a la sección Java, seleccione **Desactivar Java**, haga clic en **Aceptar** y después de nuevo en **Aceptar**.

## Certificados

Los certificados digitales, concedidos por entidades emisoras de certificados, indican que un sitio Web o un elemento de un sitio Web ha sido firmado digitalmente por su creador. Un certificado permite saber quién es el responsable del sitio o el elemento y comprueba que no contiene componentes malignos (como virus) y que no ha sido modificado desde que se creó el certificado.

Cuando el explorador recibe un certificado, comprueba su lista de entidades emisoras de certificados. Si encuentra una que coincida permite que continúe la actividad. Si el explorador indica que en el certificado falta algo, lo más seguro es cancelar la acción.

---

## **CREAR UN SITIO WEB**

Gracias a las herramientas actuales de creación para el Web y a todos los recursos que sobre este tema abundan en Internet, puede crear un sitio Web con una magnífica apariencia sin saber nada del código del lenguaje de marcado de hipertexto (HTML). Por supuesto, si desea aprender los aspectos técnicos de HTML, existe un buen número de sitios en los que puede encontrar información. Este capítulo le ayudará a empezar.

[Obtener un dominio o presencia en el Web](#)

[Es sencillo con FrontPage® 98, Office 97 y Publisher 98](#)

[Liquid Motion™](#)

[Definiciones para el principiante](#)

[Editor DHTML de Microsoft®](#)

[Cómo ver el código fuente](#)

[Cómo guardar los documentos de Microsoft Office como archivos HTM](#)

### **Obtener un dominio o presencia en el Web**

Antes de poder publicar un sitio Web en el World Wide Web, aquél necesita disponer de una dirección única. Esta dirección es el nombre de dominio. El nombre de dominio de Microsoft, por ejemplo, es "microsoft.com" y el localizador de recursos uniforme (URL) de la página principal de Microsoft es <http://www.microsoft.com>

Algunos proveedores de servicio Internet reservan partes de sus discos duros para las páginas Web de los suscriptores que no necesitan registrar un dominio. Consulte con su proveedor de servicios para ver si existe esa posibilidad, puede estar incluida en su cuota de acceso mensual.



### **¿Qué es InterNIC?**

Para asegurarse de que cada sitio Web tiene una dirección que no es utilizada por nadie más, los nombres de dominio deben registrarse con el administrador del dominio. En la mayor parte de los casos se trata de InterNIC. InterNIC administra los nombres de dominio que terminan en .com (para organizaciones comerciales), .org (para organizaciones sin ánimo de lucro), .net (para redes), .edu (para instituciones educativas), .gov (para organizaciones gubernamentales) y .mil (para servicios del ejército).



### **Cómo registrarse**

Puede buscar el nombre de dominio que desea utilizar para comprobar si ya está siendo utilizado y puede registrar el nombre de un dominio a través de InterNIC. La sección [Registration Services](#) del sitio de InterNIC proporciona información detallada acerca del modo de registrarse y los requisitos previos para poder hacerlo. La página [FAQ](#) de este sitio resultará especialmente útil.

### **Es fácil con FrontPage 98, Office 97 y Publisher 98**

Algunos productos de Microsoft® hacen que sea más fácil crear su propio sitio Web sin tener conocimientos del código del lenguaje de marcado de hipertexto (HTML).



### **Microsoft FrontPage 98**

Con [Microsoft FrontPage® 98](#), la herramienta de creación y administración de sitios Web, puede crear y administrar con facilidad sitios Web de apariencia profesional, al utilizar contenido (palabras, imágenes y otros elementos) que ya tiene en otras aplicaciones.

**Cree fácilmente sitios Web con una magnífica apariencia**



- Características fáciles de utilizar de última tecnología que le permiten crear sitios Web profesionales sin programar. No hay necesidad de escribir código.
- Cree marcos y dibuje tablas. Agregue gráficos dinámicos a sus sitios mediante Microsoft Image Composer, Microsoft GIF Animator y más de 2.000 imágenes de ejemplo. Incluya fácilmente eficaces funciones Web mediante los subprogramas basados en Java, controles ActiveX® y los complementos del explorador.
- La compatibilidad con bases de datos y páginas Active Server permite incluir fácilmente el contenido de bases de datos en las páginas Web y que los usuarios realicen consultas dinámicas a bases de datos de su sitio Web, proporcionándoles acceso directo a la información que necesitan.

#### **Administre de forma efectiva el contenido Web y la estructura del sitio**

- Las completas herramientas de administración le permiten crear rápidamente y mantener sitios Web bien organizados. Vea la estructura de exploración de su sitio Web, los directorios de información, los hipervínculos, su estado o todos los archivos al mismo tiempo.
- Con el mantenimiento automático de hipervínculos puede realizar cambios libremente sin preocuparse por los vínculos rotos. Gracias a los estilos temáticos para todo el sitio puede actualizar fácilmente la apariencia de su sitio con sólo hacer clic en un botón.
- Las flexibles características de colaboración tales como las listas de tareas y la creación remota y/o local le permiten trabajar con otras personas para crear y administrar su sitio Web.

#### **Integra sin problemas el contenido existente y las aplicaciones conocidas**

- La integración sin problemas con el contenido existente y con las aplicaciones de las que ya dispone aumenta la productividad desde el primer momento. Asimismo, la gran integración con el explorador hace que sea fácil personalizar y ver el contenido de su sitio Web.
- Por ejemplo, puede administrar sin esfuerzos los hipervínculos de Microsoft Office 97 y utilizar herramientas conocidas como el corrector ortográfico y el diccionario de sinónimos para asegurar que el contenido de su sitio Web es preciso y convincente.
- La eficacia de FrontPage ya está disponible en todos los servidores Web populares que se ejecutan en diferentes sistemas operativos.

#### **Microsoft Office 97**

Si está acostumbrado a trabajar con [Microsoft Office 97](#), conocerá sus sofisticadas herramientas para procesar textos y administrar bases de datos y hojas de cálculo, crear presentaciones con diapositivas y muchas otras tareas. Con Office 97, dispone de la versatilidad de un conjunto de aplicaciones integradas para crear sitios Web eficaces.

#### **Qué puede hacer**

Office 97 es totalmente compatible con el [Asistente para la publicación en Web de Microsoft](#), que puede descargar\* y utilizar para publicar su sitio en un servidor Web. A continuación hay algunas de las características adicionales que hacen de Office 97 una herramienta útil para crear contenido Web:

- El procesador de textos Microsoft Word 97 le permite crear y ver documentos tal y como aparecerán en Internet. Word incluye plantillas personalizadas para diseñar páginas Web y guardar los documentos en formato HTML con un solo clic del mouse.
- Las hojas de cálculo de Microsoft Excel 97 le proporcionan un método sencillo para vincular una hoja de cálculo a una página Web, de forma que los usuarios puedan ver los datos en su complejo formato tabular.
- El sistema de administración de bases de datos de Microsoft Access 97 convertirá en HTML sus datos o informes, lo que le permitirá publicar en el Web páginas dinámicas e interactivas con información de sus bases de datos.
- El programa de presentaciones gráficas Microsoft PowerPoint® 97 hace que sea posible publicar presentaciones en Internet en forma de páginas Web, y que incluya animaciones y narración.

### **Microsoft Publisher 98**

El programa de edición [Microsoft Publisher 98](#) hace que se fácil crear sitios Web, boletines, prospectos, folletos, tarjetas de presentación y muchos otros elementos con calidad profesional. Publisher le permite seleccionar entre varias opciones para crear sitios Web:

- Puede empezar a diseñar su sitio Web desde cero.
- Puede empezar con una de las más de 1.600 plantillas personalizables con diseño profesional.
- Puede dejar que AutoConvert convierta automáticamente otro de sus proyectos de Publisher, como un folleto o un boletín, en un sitio Web. Después puede realizar en él los cambios que desee.

#### **Crear un sitio único**

Publisher tiene más de 10.000 imágenes prediseñadas de alta calidad, 1.500 fotografías, 1.000 gráficos para el Web, 175 fuentes, 340 archivos GIF animados, 60 esquemas de color y muchos otros elementos de diseño para crear publicaciones excepcionales. Puede aplicar una misma apariencia a diferentes tipos de publicaciones, por lo que es fácil diseñar un conjunto coordinado de material de mercadotecnia, para imprimirlo y para publicarlo en el Web.

### **Liquid Motion**

La herramienta para aplicaciones Web multimedia [Microsoft® Liquid Motion](#) hace que resulte igual de sencillo para novatos y expertos crear y publicar animaciones en el World Wide Web. Las animaciones pueden dar nueva vida a su sitio Web de cara a los usuarios, atraer su atención, captar su interés y comunicar el mensaje deseado.

Con Liquid Motion puede crear animaciones impactantes sin necesidad de saber programar. Como es fácil de aprender y utilizar, obtendrá resultados rápidamente. Además, Liquid Motion funciona con cualquier explorador que permita utilizar Java y en cualquier plataforma, como [Microsoft Internet Explorer 4.0](#) y [3.0](#), por lo que puede estar seguro de que los visitantes de sus páginas podrán ver sus animaciones.

### **Definiciones para el principiante**

Estos términos habituales relacionados con el Web y sus definiciones le ayudarán a comprender los fundamentos de la creación de contenido Web.

**HTML:** lenguaje de marcado de hipertexto, es un lenguaje de programación que se utiliza para crear sitios Web. Contiene códigos estándar, o etiquetas, que determinan la apariencia de una página Web cuando la muestra el explorador. Por ejemplo, las etiquetas se utilizan para crear encabezados, párrafos y listas. Las etiquetas HTML también permiten crear los hipervínculos que conectan la información en la World Wide Web. Con la ayuda de programas como la herramienta de creación y administración de sitios Web Microsoft® FrontPage® 98. puede crear sitios Web sin necesidad de aprender HTML. Sin embargo, puede resultar útil conocer los fundamentos.

**DHTML:** HTML dinámico, una extensión del lenguaje HTML. DHTML proporciona mayor control de la apariencia y la posición de cada elemento de su página Web. Se considera dinámico porque proporciona un medio para incluir elementos en su página Web que se descargan en el explorador del usuario junto con la página, pero no son visibles hasta que el usuario interactúa con ellos. Esto significa que cuando el usuario interactúa con los elementos, estos se activan sin solicitar nada al servidor del sitio Web.

**XML:** lenguaje de marcado extensible, un formato de datos que hace que sea más fácil definir tipos de documentos, crear y administrar documentos y compartirlos en el Web. Es una versión abreviada del lenguaje de marcado generalizado estándar(SGML), el lenguaje del que parten todos los lenguajes utilizados para describir documentos, por lo que es más fácil de utilizar, comprender y también es más fácil escribir programas para él.

**ActiveX®:** una tecnología de Microsoft que facilita la interoperabilidad, o compartir datos, entre aplicaciones. Los controles ActiveX (o aplicaciones componente) permiten incrustar objetos inteligentes en el sitio Web. Una vez que el explorador descarga un control ActiveX desde un sitio, éste permanece en el PC, de forma que si el usuario visita otro sitio (u otra página del mismo sitio) que utiliza el mismo control, ya no es necesario descargarlo de nuevo.

**Java:** lenguaje de programación orientado a objetos desarrollado por Sun Microsystems para crear subprogramas, o programas que pueden ser distribuidos como datos adjuntos de los documentos Web. Se puede incluir un subprograma en una página HTML, de igual forma que una imagen. Cuando se utiliza un explorador compatible con Java para ver una página que incluye un subprograma de Java, el código se transfiere a su sistema y el explorador lo ejecuta.

**Secuencias de comandos:** método rápido de programación que proporciona a los usuarios sin formación técnica un método para crear contenido más rico en sus sistemas y a los programadores un método rápido para crear aplicaciones

sencillas. Las secuencias de comandos permiten establecer y almacenar variables y trabajar con datos desde el código HTML. En la actualidad muchos sitios Web utilizan las secuencias de comandos para comprobar qué explorador utiliza el usuario, validar los datos escritos por los usuarios, trabajar con subprogramas o controles y comunicarse con el usuario.

### **Nuevo editor DHTML de Microsoft**

El componente de edición DHTML de Microsoft® permite a los autores de Web y a los desarrolladores de aplicaciones agregar a sus sitios Web y aplicaciones las capacidades de edición del lenguaje de marcado de hipertexto (HTML) . El componente de edición utiliza la tecnología del modelo de objetos componentes (COM) de Microsoft para poner a disposición inmediata del desarrollador servicios de edición, como el formato HTML básico, tablas, deshacer/rehacer, buscar y la ubicación absoluta.

Las aplicaciones que utilizan o generan HTML (por ejemplo, las aplicaciones de creación para el Web, correo electrónico o aplicaciones de Ayuda, o formularios HTML) pueden utilizar el Componente de edición DHTML para proporcionar amplias posibilidades de edición. Los autores de Web y los desarrolladores pueden utilizar cualquier lenguaje para obtener acceso a los servicios de edición y proporcionar una interfaz de usuario para estas características de edición. También pueden tener acceso al modelo de objetos documentos para agregar características personalizadas de edición más sofisticadas a sus aplicaciones.

### **Cómo ver el código fuente**

Uno de los métodos para aprender el lenguaje de marcado de hipertexto (HTML) y cómo crear el sitio Web que desea es ver el código fuente de un sitio Web que contenga alguna característica parecida a la que desea crear. La versión 4.0 de Microsoft Internet Explorer facilita este proceso. En el menú Ver, haga clic en **Código fuente**. El código fuente de la página que ve en ese momento aparecerá en una ventana del Bloc de notas

### **Cómo guardar los documentos de Microsoft Office como archivos HTM**

Para guardar un documento de Microsoft® Office, por ejemplo un archivo de Microsoft Excel o Word, como un archivo HTM:

1. En el menú Archivo, haga clic en **Guardar como**.
2. En la lista desplegable "Guardar como tipo", seleccione **Archivo HTML** y haga clic en **Aceptar**.

---

## ***BREVE HISTORIA DE INTERNET***

El Departamento de Defensa de los EE.UU. sentó las bases de Internet hace apenas 30 años con una red denominada ARPANET. Pero el público en general no comenzó a utilizar Internet hasta después del desarrollo del World Wide Web a principios de los años 90. En junio de 1993, sólo había 130 sitios Web. Ahora hay millones. A continuación veremos brevemente cómo se produjo el fenómeno.

[El principio: ARPANET](#)

[Tim Berners-Lee: padre del Web](#)

[¿Quién manda aquí?](#)

[¿Qué son los dominios?](#)

### **El principio: ARPANET**

En 1957, El Gobierno de los Estados Unidos formó la Agencia de Proyectos de Investigación Avanzados (ARPA), una división del Departamento de Defensa encargada de asegurar el liderazgo de los EE.UU. en la ciencia y la tecnología con aplicaciones militares. Algo más de una década después, en 1969, ARPA estableció ARPANET, la predecesora de Internet.

### **Investigación y educación**

ARPANET era una red que conectaba los principales sistemas informáticos de la Universidad de California en Los Angeles, la Universidad de California en Santa Barbara, el Instituto de Investigación de Stanford y la Universidad de Utah. En un par de años se unieron a la red unas cuantas instituciones educativas y de investigación más.

Como respuesta a la amenaza de un posible ataque nuclear, ARPANET fue diseñada para permitir que la comunicación se mantuviera si eran destruidos uno o más sitios. A diferencia de hoy en día, en que millones de personas tienen acceso a Internet desde su casa, su trabajo o desde la biblioteca pública, ARPANET sólo estaba al alcance de los profesionales de la informática,



ingenieros y científicos que sabían moverse por sus complejos recovecos.

### **Evolución**

Durante la década de los 70, los desarrolladores crearon los protocolos que se utilizan para transferir información a través de Internet. A principios de los años 80 ya habían nacido los grupos de Usenet y el correo electrónico. La mayor parte de los usuarios pertenecían a universidades, aunque las bibliotecas también comenzaban a conectar sus catálogos con Internet. A finales de los años 80 los desarrolladores crearon índices, como Archie y el Servidor de Información de Área Extensa (WAIS) para mantener un registro de la información presente en Internet. Para proporcionar a los usuarios una interfaz amigable y fácil de utilizar, la Universidad de Minnesota creó el Gopher, un sencillo sistema de menús para el acceso a los archivos, en 1991.

### **Tim Berners-Lee: padre del Web**

El World Wide Web nació en 1991, gracias al desarrollador Tim Berners-Lee y otros miembros del Laboratorio Europeo de Física de Partículas, también conocido como Conseil Européen pour la Recherche Nucléaire (CERN). El equipo del CERN creó el protocolo, basado en hipertexto, que permite conectar el contenido del Web mediante hipervínculos. Berners-Lee dirige ahora el World Wide Web Consortium (W3C), un grupo de representantes de la industria y de las universidades que supervisa los estándares de la tecnología Web.

Inicialmente, Internet estaba limitada a usos no comerciales debido a que su financiación provenía del Gobierno de los Estados Unidos y su red troncal procedía en gran parte de tres agencias del Gobierno: la Fundación Nacional de la Ciencia, la Administración Nacional Espacial y de Aviación y el Departamento de Energía de los EE.UU. Pero según comenzaron a aparecer más redes independientes, los usuarios tuvieron acceso a sitios Web comerciales sin utilizar la red financiada por el Gobierno. A finales de 1992, el primer proveedor de servicios electrónicos comercial, Delphi, ofreció acceso completo a Internet a sus suscriptores, después le siguieron otros proveedores.

En junio de 1993, el Web sólo abarcaba 130 sitios. Un año después, el número había crecido a casi 3.000. Hacia abril de 1998, había más de 2,2 millones de sitios en el Web.

### **¿Quién manda aquí?**

No existe ninguna autoridad que controle el World Wide Web. Las herramientas de creación de sitios Web actuales permiten que prácticamente cualquier usuario con acceso a un equipo personal y a Internet publique un sitio Web y contribuya a la definición de lo que es este medio y lo que puede hacer. Sin embargo, el World Wide Web Consortium (W3C) supervisa el desarrollo de la tecnología Web.

### **Usted da forma a la Web**

Según el creador del World Wide Web, Tim Berners-Lee, "[El sueño detrás del Web](#) es el de un espacio de información común en el que la comunicación se realiza al compartir información. Su universalidad es esencial: el hecho de que un vínculo de hipertexto pueda llevar a cualquier sitio, ya sea personal, local o global, ya sea burdo o muy elegante."

Con el desarrollo de herramientas que permiten crear sitios Web sin tener conocimientos del lenguaje de marcado de hipertexto (HTML), este sueño se está convirtiendo en realidad. Si lee el capítulo "Crear un sitio Web", podrá convertirse



en una de las fuerzas que da forma a este "espacio de información común."

### **World Wide Web Consortium**

Para supervisar los estándares de la tecnología Web está el W3C, establecido por Berners-Lee en 1994. Este grupo internacional de representantes de la industria y de las universidades promueve la Web al desarrollar protocolos comunes para transmitir información a través de Internet. El consorcio proporciona información, código de referencia, prototipos y aplicaciones de ejemplo a los desarrolladores y usuarios. Sus sedes están en el Laboratorio de Informática del Instituto Tecnológico de Massachusetts en los Estados Unidos, el Institut National de Recherche en Informatique et en Automatique en Europa y el campus de la Universidad Shonan Fujisawa de Keio en Japón.

### **¿Qué son los dominios?**

Los dominios dividen los sitios del World Wide Web en categorías según la naturaleza de su propietario y forman parte de la dirección de un sitio, o localizador uniforme de recursos (URL). Los dominios de primer nivel habituales son:

- .com-para empresas comerciales.
- .org-para organizaciones sin ánimo de lucro.
- .net-para redes.
- .edu-para instituciones educativas.
- .gov-para organizaciones gubernamentales.
- .mil-para servicios militares.
- .int-para organizaciones establecidas por tratados internacionales.

Se han propuesto más dominios de primer nivel con tres y cuatro letras, y es probable que se implementen algunos de ellos. Cada país conectado a Internet tiene un dominio de primer nivel de dos letras.

# COREL CHART

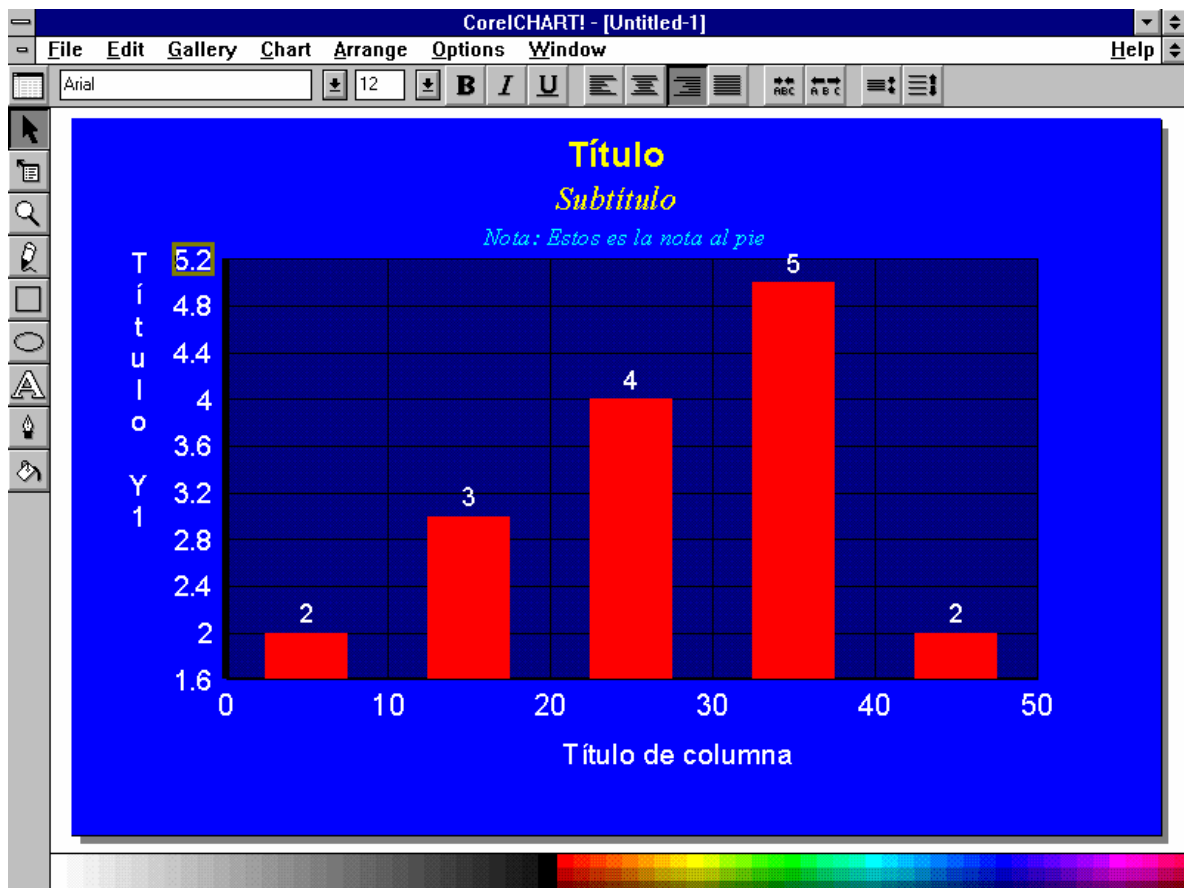


FIG. LA PANTALLA DE COREL CHART

## Creación de gráficos

1. Seleccione el comando Nuevo... del menú Archivo para abrir la ventana de diálogo Nuevo.
2. Seleccione un tipo de gráfico (de barras, tarta, superficies en 3D, etc.) en el recuadro Galería.
3. Seleccione un gráfico en los ejemplos que aparecen en el recuadro de simulación Tipos de gráfico.
4. Asegúrese de que la opción Usar datos de muestra está activada.
5. Efectúe una doble pulsación sobre la simulación del gráfico seleccionado o pulse el ratón sobre Aceptar. El gráfico aparecerá en Visión de gráficos.



6. Pulse el ratón sobre el botón Gestor de datos de la esquina superior izquierda de la barra de herramientas para abrir la ventana del Gestor de datos correspondiente al gráfico actual.

7. Lleve el cursor a varias celdas de datos de la matriz para sustituir los datos existentes por otros números y texto introducidos por el usuario.

8. Pulse el ratón sobre Exploración automática. Esto se hace para el Gestor de datos pueda actualizar los cambios introducidos en el rango de datos y las cabeceras de texto.

9. Pulse el ratón sobre el botón de la esquina superior izquierda de la barra de herramientas con el fin de ver los cambios introducidos en el gráfico.

10. Guarde el gráfico bajo otro nombre utilizando el comando Guardar como... del menú Archivo.

### **Introducción de información en el Gestor de datos**

El Gestor de datos es como una hoja de cálculo. Es el lugar en el que se introducen o importan texto o números y se asignan atributos, o rótulos, a las celdas de texto y números para determinar la parte del gráfico en la que se convertirán. Hay una ventana para el Gestor de datos de cada gráfico. (Puede haber varios gráficos y ventanas de Gestor de datos abiertos al mismo tiempo).

Asignación manual de atributos a las celdas:

1. Si todavía no ha introducido texto, pulse el ratón sobre una celda para seleccionarla e introducir un número o texto. Una vez finalizada la introducción de datos, pulse el ratón sobre el botón Introducir que figura al lado del recuadro de contenido o pulse la tecla Intro. Emplee las teclas de flecha para salir de la celda o el ratón para activar otra celda.

Para seleccionar celdas no contiguas, mantenga pulsada la tecla Ctrl y vaya pulsando el ratón sobre cada una de las celdas que desee activar.

2. Una vez seleccionadas las celdas, vaya a la lista de elementos desplegados del menú, pulse el ratón sobre la flecha de desplazamiento y después sobre el nombre del atributo que desee utilizar.

3. Para ver las celdas que disponen de atributos, utilice las teclas del cursor para pasar de unas a otras. Cuando llegue a una celda, su atributo aparecerá en la lista de elementos del gráfico.

Para importar datos en el gestor de datos:

1. Pulse el ratón sobre para abrir la ventana del Gestor de datos correspondiente al gráfico actual. Seleccione Importar datos... en el menú Archivo. En la ventana de diálogo Importar seleccione el formato de archivo que vaya a importar en el recuadro de lista que aparece en la esquina inferior izquierda y busque en los discos y directorios el archivo que desea.

## Utilización de pictogramas en los gráficos

Esta opción permite utilizar un archivo de CorelDRAW, o archivo CDR, como relleno para un gráfico de barras o histograma. Se puede acceder a esta función pulsando el ratón sobre la Herramienta Relleno y el icono Pictograma del menú lateral.

1. Seleccione una barra.
2. Seleccione Mostrar como pictograma en el menú Gráfico.
2. Pulse el ratón sobre la herramienta Relleno del menú lateral y sobre el botón pictograma. Esto abrirá la persiana Pictograma desde la que podrá acceder a los directorios que contienen archivos CDR.
3. Seleccione un archivo .CDR. El objeto quedará relleno por versiones de menor tamaño del pictograma (por ejemplo, una barra mostrará un pictograma para todas las divisiones de la escala numérica).

## Intercambio de información con otras aplicaciones

### Importación y exportación

CorelCHART incluye una serie de filtros de formato de archivo que permiten intercambiar gráficos entre CorelCHART y otras aplicaciones, o exportar un gráfico para su utilización en otra aplicación.

La importación permite acceder a gráficos creados en otros programas de ilustración y paquetes de presentación, además de clip art e imágenes escaneadas.

La exportación guarda los archivos de CorelCHART en formatos utilizados por otros programas. De esta manera, se pueden crear gráficos para muchos programas de autoedición y proceso de texto de utilización muy extendida.

Dado que cada formato gestiona la información de los archivos gráficos de una manera distinta, no siempre resulta posible traducir con precisión el contenido de un formato a otro. La intensidad de la diferencia depende del formato gráfico utilizado para importarlo o exportarlo.

### Vinculación e incorporación de objetos (OLE)

La capacidad OLE de Windows proporciona dos métodos para crear un gráfico u otro objeto en una aplicación de Windows y utilizarlo en otro. La vinculación de objetos implica la utilización de un objeto existente en un archivo creado en una aplicación servidor y su vinculación con otro archivo de otra aplicación, conocida como aplicación cliente. La incorporación de objetos implica el inicio del proceso en la aplicación cliente, la ejecución de la aplicación servidor con la aplicación cliente, la creación del objeto y su inclusión en el documento de la aplicación

cliente. Los objetos incorporados, a diferencia de los objetos vinculados, sólo existen en el documento cliente (también llamado de destino).

La ventaja clave del OLE sobre la importación y la exportación estriba en que es posible modificar un objeto después de vincularlo o incorporarlo y actualizarlo en el archivo de destino con facilidad.

CorelCHART es a la vez una aplicación cliente y servidor, lo mismo que CorelDRAW. Esto quiere decir que pueden ser tanto el destino como el origen de los objetos vinculados o incorporados.

#### Utilización del Portapapeles sin OLE

El Portapapeles es una zona de almacenamiento temporal utilizada para transferir texto, gráficos y otro tipo de datos entre aplicaciones de Windows. OLE utiliza el portapapeles de Windows como dispositivo de comunicación; también puede servir para mover objetos de un archivo de gráficos a otro, para pegar un gráfico en otra aplicación de Windows o para pegar un gráfico no editable en un archivo de gráficos. No se puede actualizar ni editar un objeto después de colocarlo en el Portapapeles.

Transfiera la información al portapapeles utilizando los comandos Cortar, Copiar y Pegar del menú Edición.

#### ***Exportación de gráficos para su utilización en otros programas***

1. -Abra el archivo de CorelCHART que desee exportar.
2. Seleccione Exportar en el menú Archivo.
3. Seleccione el formato de exportación que desee utilizar en el recuadro Mostrar archivos de tipo.
4. Escoja una de las siguientes alternativas:
  - Acepte el nombre que aparece el recuadro Nombre de archivo para el gráfico que se está exportando.
  - Escriba un nuevo nombre en el recuadro Nombre de archivo.
  - Seleccione un nombre en la lista de Nombre de archivo.

CorelCHART añade automáticamente la extensión correspondiente al formato de exportación seleccionado.

Para guardar el archivo en otro directorio, seleccione la unidad en el recuadro Unidad y el directorio en el recuadro Directorios.

5. Pulse el ratón sobre Aceptar.

Dependiendo del formato seleccionado, existe la posibilidad de que aparezca otra ventana de diálogo. Seleccione las opciones que desee en esta ventana de diálogo y pulse el ratón sobre Aceptar.

### ***Importación de gráficos de otros programas***

Para importar objetos gráficos al gráfico:

1. Seleccione Importar en el menú Archivo.
2. Seleccione el formato de importación que desee utilizar en el recuadro Mostrar archivos de texto.

El recuadro Nombre de archivo muestra los archivos del directorio actual que tienen la extensión correspondiente al formato seleccionado. Si el archivo que desea está en otro directorio, seleccione el archivo en el recuadro Unidades y el Directorio en el recuadro Directorio.

4. Escriba o seleccione los datos que desee importar en el recuadro Nombre de archivo.
5. Pulse el ratón sobre Aceptar.

### ***Copiado de gráficos e imágenes en el Portapapeles de Windows***

El portapapeles es una zona de almacenamiento temporal utilizada para transferir texto y gráficos entre aplicaciones de Windows. También se puede utilizar el Portapapeles para mover anotaciones entre archivos de CorelCHART.

Para situar una copia de un gráfico en el Portapapeles:

1. Para copiar un gráfico, seleccione Copiar gráfico del menú Edición. Para un objeto gráfico, utilice la herramienta Selección para seleccionarlo y escoja el comando Copiar del menú Edición.
2. Abra el archivo de CorelCHART o la otra aplicación en la que desee copiar la animación.
3. Seleccione Pegar en el menú Edición.

Para cortar un objeto de un gráfico y pegarlo en otro archivo:

1. Seleccione el objeto con la herramienta Selección.
2. Seleccione Cortar en el menú Edición.
3. Abra el archivo en el que desee situar la anotación.
4. Seleccione el comando Pegar del menú Edición.

## ***Vinculación de gráficos con archivos de CorelDRAW***

1. Abra el archivo del gráfico que desee vincular con CorelDRAW.
2. Seleccione el comando Copiar gráfico del menú Edición. Esto copia todo el gráfico en el Portapapeles de Windows.
3. Seleccione el comando Salir del menú Archivo para abandonar CorelCHART.
4. Arranque CorelDRAW efectuando una doble pulsación sobre su icono del Administrador de programas.
5. Abra el archivo de CorelDRAW con el que desee vincular el gráfico.
6. Seleccione el comando Pegar especial del menú Edición.
7. Seleccione Gráfico Corel como tipo de archivo en la ventana de diálogo Pegar especial, y pulse sobre Pegar vínculo. El gráfico aparecerá ahora en CorelDRAW como objeto vinculado y tendrá tiradores.

## ***Establecimiento de vínculos de intercambio dinámico con hojas de cálculo***

Para establecer un vínculo de intercambio dinámico con una hoja de cálculo de Windows que admita esta relación:

1. Ejecute CorelCHART con la ventana correspondiente al Gestor de datos del gráfico abierto.
2. Ejecute la hoja de cálculo efectuando una doble pulsación sobre su icono en el Administrador de programas.
3. Abra el correspondiente archivo de datos.
4. Seleccione el rango de celdas de la hoja de cálculo que desee vincular.
5. Seleccione Copiar en el menú Edición.
6. Pulse el ratón sobre el icono de CorelCHART o la parte visible de su ventana o utilice el comando Pasar a del Administrador de tareas para hacer que pase a ser la ventana activa.
7. Ponga el cursor sobre la celda del Gestor de datos en la que desee que comiencen los datos vinculados.
8. Seleccione Pegar vínculo en el menú Edición.

Los datos del gráfico quedarán ahora vinculados con el archivo de origen. Los datos del archivo del gráfico se actualizarán automáticamente siempre que se modifique la hoja de cálculo.

## ***Herramientas de CorelCHART***

## ***El botón Gestor de datos***

Pulse sobre este icono para acceder al Gestor de datos con el fin de introducir datos o texto para el gráfico.

Para volver a la visión de gráficos, pulse sobre el icono visión de Gráficos del Gestor de datos.

## ***Herramienta Selección***

Selecciona, mueve, o altera los objetos de un gráfico

Utilice la Herramienta Selección para mover, estirar y escalar interactivamente los objetos, así como para seleccionar texto para su modificación. Después de seleccionar un objeto, puede utilizar los comandos de los menús o la caja de herramientas para cambiarlos.

Para seleccionar un objeto

1. Para seleccionar un objeto, pulse sobre el objeto o su contorno. Al seleccionarlo, aparecerá un recuadro resaltado con ocho tiradores (en algunos casos sólo aparece un recuadro resaltado).
2. Para estirar un objeto, pulse y arrastre un tirador situado en la esquina del recuadro. Si mantiene pulsada la tecla Ctrl cuando mueva un objeto, éste se desplazará a lo largo de sus ejes en vertical u horizontal.
3. Para moverlo, pulse el ratón en cualquier lugar del recuadro donde no haya un tirador, y arrastre el objeto.

Métodos abreviados

- Si pulsa la barra espaciadora mientras utiliza las herramientas de dibujo, se activará la herramienta Selección.
- Si pulsa la barra espaciadora cuando la herramienta Selección está seleccionada, se activará la última herramienta de dibujo utilizada.

## ***Herramienta de acceso rápido***

Esta herramienta permite el acceso directo a los menús correspondientes a un determinado elemento del gráfico.

Por ejemplo, si pulsa sobre el recuadro Subtítulo con la herramienta de acceso rápido, aparecerá la siguiente selección de menú.

NOTA: También se puede activar esta herramienta seleccionando un objeto y pulsando el botón derecho del ratón.

## ***Herramienta Zoom***

Cambia el tamaño de la ventana

Pulse el icono de la herramienta y mueva el cursor a lo largo del menú lateral para seleccionar la visión deseada; a continuación, suelte el botón del ratón. Entre las opciones del menú lateral se incluyen:

- ajustar al tamaño de la ventana
- 25% del tamaño real
- 50% del tamaño real
- 100% del tamaño real
- 200% del tamaño real
- 400% del tamaño real

## ***Herramienta Lápiz***

Dibuja rectas, polígonos, curvas y flechas.

Con el ratón pulse sobre el icono de la herramienta para ver un menú desplegable que permite elegir la opción de dibujo.

Para dibujar una recta, pulse esta opción, muévase al lugar donde quiere que comience y pulse el botón del ratón. Manténgalo pulsado y arrastre el ratón hasta de final de la línea. Suelte el botón.

Para dibujar un polígono, pulse esta opción, muévase al lugar donde quiera que comience el polígono y pulse el botón del ratón. Suéltelo, vaya a otra posición y vuelva a pulsarlo. Repita este proceso hasta conseguir la figura deseada y entonces pulse dos veces para cerrar el polígono.

Empleado para dibujar a mano alzada. Pulse esta opción, vaya a la posición donde quiera que comience la línea y pulse el botón del ratón. Manténgalo pulsado mientras dibuja.

Para dibujar una flecha, pulse esta opción, lleve el cursor al lugar donde quiera que comience la flecha, y pulse el botón del ratón. Manténgalo pulsado, y arrástrelo a la posición donde vaya a situar la punta de la flecha, y suelte el botón. Utilice la herramienta de acceso rápido para seleccionar el Estilo de flecha

NOTA: A las rectas, curvas y flechas dibujadas con la herramienta Lápiz se les asignan automáticamente los atributos por defecto de color y grosor del filete en vigor. Utilice la herramienta Filete para modificar estos atributos.

## ***Herramienta Rectángulo***

Dibuja rectángulos y cuadrados. A los objetos dibujados con la herramienta Rectángulo se les asignan los atributos por defecto de relleno y filete.

Para dibujar con la herramienta Rectángulo:

1. Pulse sobre la herramienta Rectángulo y lleve el cursor al punto en el que desee que comience el rectángulo.
2. Pulse y arrastre el ratón hasta dibujar el rectángulo deseado para, a continuación, soltarlo.

Para redondear las esquinas o los rectángulos o cuadrados:

1. Seleccione la herramienta y pulse el ratón sobre el rectángulo o cuadrado.
2. Seleccione Redondez de esquinas en el menú desplegable
3. Seleccione uno de los grados de redondez en el menú lateral.

Método abreviado

Para pasar de la herramienta Rectángulo a la herramienta Selección, pulse el ratón sobre la barra espaciadora.

### ***Herramienta Elipse***

Dibuja elipses y círculos. A los objetos dibujados con la herramienta Elipse se les asignan automáticamente los atributos por defecto de relleno y filete.

Para dibujar con la herramienta Elipse:

1. Pulse el ratón sobre la herramienta Elipse y ponga el cursor sobre el punto en el que desee que comience la elipse.
2. Pulse y mantenga pulsado el ratón hasta conseguir la elipse que desee para, posteriormente, soltarlo.

Si mantiene pulsada la tecla Ctrl cuando arrastre el ratón, se dibujará un círculo.

Método abreviado

Para pasar de la herramienta Elipse a la herramienta Selección sucesivamente, pulse la barra espaciadora.

### ***Herramienta Texto***

Añade texto para las anotaciones.

Para añadir texto:

1. Pulse la herramienta Texto. El cursor se convertirá en una cruz.
2. Muévelo al lugar donde quiere que comience el texto, y pulse y arrastre para dibujar un rectángulo en el que introducir el texto.
3. Suelte el ratón.
4. Cuando el cursor adquiera la forma de una barra | parpadeante (denominada punto de inserción), escriba el texto.



Para editar y formatear texto:

1. Seleccione la herramienta Texto y lleve el cursor al texto que desee editar.
2. Una vez aparecido el cursor en forma de barra, escoja una de las siguientes opciones
3. Edite el texto con las siguientes técnicas.

Para Haga esto

Mover el punto de inserción Pulse el cursor sobre su nueva posición o emplee las siguientes teclas o combinaciones de teclas:

←, →, ↑, ↓, Ⓜ	mueve en la dirección de la flecha
INICIO	pasa al comienzo del texto
FIN	pasa al final de la línea
Seleccione texto	Arrastre el ratón alrededor del texto o mantenga pulsada la tecla MAYUS y pulse las teclas ←, Ⓜ, INICIO o FIN.
Borrar texto	Pulse RETROCESO o SUPR para borrar un carácter cada vez o seleccione los caracteres y pulse la tecla SUPR.
Añadir texto	Mueva el punto de inserción al lugar en el que desee que comience el texto y comience a escribir.
Sustituir texto	Seleccione el texto y comience a escribir.
Copiar en el Portapapeles	Seleccione le texto y pulse CTRL+C o seleccione Copiar en el menú Edición.

Cortar y almacenar en el Portapapeles Seleccione texto y pulse CTRL+X o seleccione Cortar en el menú Edición.

Pegar Pulse CTRL+V o seleccione Pegar en el menú Edición.

Para formatear el texto o cambiar su color o filete:

· Resáltelo y seleccione a continuación los códigos de formato que desee en la banda de texto. Escoja los atributos de filete y relleno en los menús de las herramientas Relleno y Filete o en la Paleta de colores en pantalla.

### **Herramienta Filete**

Abre un menú desplegable en el que se pueden seleccionar el grosor y el color del filete. Los colores también se pueden elegir en la Paleta de colores en pantalla.

Iconos del menú

Pluma del filete

Abre la ventana de diálogo Pluma del filete en la que se puede introducir un valor para el grosor del filete.

Ninguno

Suprime los filetes de los objetos seleccionados.

Anchuras de línea

Selecciona las anchuras de línea de la línea que tienen entre 1/4 (línea fina), 1/2, 1, 2, 4, 8, 12, 16, 20 y 24 puntos.

Ventana de diálogo Color del filete

Abre la ventana de diálogo Color del filete para seleccionar o crear colores para el relleno del filete.

Blanco, negro, gris

Selecciona el blanco, negro y sucesivos tonos de gris para rellenar el filete.

### ***La herramienta Relleno***

Muestra un menú que consta de una serie de iconos para especificar el relleno de los objetos.

Iconos del menú

Color de relleno uniforme

Muestra la Ventana de diálogo Relleno uniforme para seleccionar o crear un relleno de color uniforme.

Persiana relleno

Muestra una persiana para acceder con rapidez a varios rellenos. Véase Persiana relleno

.

Ninguno

Hace que el objeto sea transparente, con lo que se podrán ver los objetos situados por detrás.

Relleno degradado

Muestra la ventana de diálogo Relleno degradado para especificar rellenos degradados.

## Patrón bicolor

Muestra la Ventana de diálogo Patrón bicolor desde la que se puede seleccionar un patrón bicolor.

## Patrón a todo color

Muestra la ventana de diálogo Patrón a todo color desde la que se puede seleccionar un patrón de relleno a todo color.

## Texturas Bitmap

Muestra la Ventana de diálogo Texturas bitmap desde la que se puede seleccionar un relleno de textura bitmap.

## Persiana pictograma

Muestra la Persiana pictograma desde la que se puede seleccionar una imagen de CorelDRAW (.CDR) como relleno para un objeto.

## Patrones y sólido

Selecciona rellenos sólidos en blanco o negro y distintos patrones.

## ***Persiana Relleno***

La persiana Relleno permite acceder con velocidad a una serie de rellenos.

### Controles de la persiana Icono plegar ventana

Ocultar los controles, dejando únicamente visible la barra de títulos.

### Relleno degradado

Muestra controles para crear rellenos degradados.

Tipo Si pulsa el botón izquierdo del ratón, se seleccionará un degradado lineal; con el botón central, se seleccionará en cambio un degradado radial y con el derecho un degradado cónico.

Es posible cambiar el ángulo de los degradados lineales y el centro de un degradado cónico o radial arrastrando el control del recuadro de simulación.

Si mantiene pulsada la tecla CTRL cuando arrastre el ratón, el ángulo de los degradados lineales quedará limitado a múltiplos de 15 grados y la cantidad de desplazamiento de los degradados radiales o cónicos a incrementos del 10 por ciento.

**Colores** Muestra una paleta desde la que se pueden seleccionar los colores inicial y final. Pulse el botón izquierdo para especificar el color inicial y el derecho para el final.

Si pulsa el ratón sobre el botón de relleno degradado sin tener ningún objeto seleccionado, se mostrará una ventana de diálogo desde la que se puede establecer el degradado actual como relleno por defecto para un determinado tipo de objeto nuevo.

#### Rellenos de patrones bicolors

Muestra controles para seleccionar y colorear rellenos de patrones bicolors.

**Colores** Muestra una paleta desde la que se pueden seleccionar colores para el fondo y primer plano. Pulse el botón izquierdo para especificar el color del primer plano o el derecho para el color del fondo.

**Lista de patrones** Si pulsa el ratón sobre la flecha de la esquina inferior izquierda del recuadro de simulación del recuadro de simulación, aparecerá una lista de patrones. Pulse el ratón sobre el patrón que desee y seleccione Aceptar. Si selecciona Cancelar, se ignorará la selección y se cerrará la lista.

Archivo muestra un menú para guardar, borrar e importar patrones

**Guardar patrón actual:** Guarda el relleno de patrón bicolor visualizado en el recuadro de simulación.

**Borrar elemento:** Borra de la lista el patrón seleccionado.

**Importar patrón:** Muestra otra ventana de diálogo que permite crear un patrón a partir de un gráfico importado.

**NOTA:** Los mejores resultados se consiguen limitando el uso de los colores a no más de dos por gráfico importado.

#### Rellenos de patrón a todo color

Muestra controles para seleccionar patrones bicolors.

**Lista de patrones** Pulse el ratón sobre la flecha de la esquina inferior derecha de la ventana de simulación para activar una lista de patrones. Pulse el ratón sobre el patrón que desee emplear y seleccione Aceptar. Si selecciona Cancelar, se ignorará la selección y se cerrará la lista.

Con Archivo se visualizará un menú para guardar, borrar e importar patrones.

**Guardar relleno actual:** Guarda el patrón de relleno a todo color visualizado en la ventana de simulación.

**Borrar elemento:** Borra de la lista el patrón seleccionado.

**Importar patrón:** Muestra otra ventana de diálogo que permite crear un patrón basado en otro gráfico importado.

## Texturas bitmap

Muestra controles para seleccionar y editar rellenos de texturas bitmap.

**Lista de texturas** Si pulsa el ratón sobre la flecha de la esquina inferior derecha de la ventana de simulación, aparecerá una lista de texturas. Pulse el ratón sobre la textura que desee utilizar y seleccione Aceptar. El comando Cancelar ignora la selección del usuario y cierra la lista.

También se puede seleccionar una textura pulsando el ratón sobre su nombre en la lista situada por encima del botón Actualizar de.

Archivo muestra un menú para elegir Bibliotecas y borrar texturas.

**Cargar biblioteca de texturas:** Abre una ventana de diálogo en la que se puede seleccionar la biblioteca que contiene la textura en cuestión. También se pueden seleccionar bibliotecas en la lista situada por debajo de la ventana de simulación.

**Borrar elemento:** Borra de la lista la textura seleccionada.

Seleccione la Biblioteca que contenga la textura que desee utilizar.

Pulse el ratón sobre el nombre de la textura que desee utilizar.

## Editar

Activa otra ventana de diálogo dependiente del tipo de relleno. Véase Relleno degradado, Patrón de relleno bicolor, Patrón de relleno a todo color o Textura bitmap.

## Aplicar

Aplica las opciones escogidas al objeto seleccionado.

## **\*Banda de texto**

La Banda de texto aparece en la parte superior de la ventana principal y le permite acceder con velocidad a varias opciones de formateo.

Recuadro/botón      Acción

Escriba o seleccione el tipo de letra que desee aplicar

Escriba o seleccione el tamaño en puntos que desee aplicar.

Selecciona y deselecciona la negrita.

Selecciona y deselecciona la cursiva.

Ninguno.

Alinea el texto con el lado izquierdo de su silueta.

Centra el texto en el interior de su silueta.

Alinea el texto con el lado derecho de su silueta.

Aumenta o reduce el espaciado entre caracteres para alinear el texto con los lados derecho e izquierdo de su silueta.

Aumenta el espaciado entre caracteres.  
Reduce el espaciado interlineal.  
Aumenta el espaciado interlineal.

### ***Persiana de Herramienta de visión en 3D***

La persiana 3D se abre seleccionando Persiana 3D en el menú Gráfico en el caso de que el gráfico de barras o de dispersión en 3D. La Persiana 3D sirve para:

- modificar el tamaño, escala y perspectiva
- modificar la longitud de los ejes y el grosor de las paredes
- rotar el gráfico.

Pulse el ratón sobre una flecha roja y mantenga pulsado su botón hasta alcanzar el efecto perseguido. Cuanto más tiempo mantenga pulsado el botón del ratón, más intenso será el efecto.

Botón/

Recuadro de comprobación	Acción
--------------------------	--------

Mueve el gráfico en vertical o diagonal o hacia adelante o atrás.

Amplía o reduce el gráfico sin modificar su posición en la página.

Panorámica en 2D inserta el gráfico en vertical o diagonal en cualquier posición de la página sin modificar el tamaño del gráfico.

Perspectiva en 3D controla la perspectiva del gráfico haciendo que una parte de éste se acerque y aumente de tamaño y alejando otras y reduciendo su tamaño. Cuanto menos paralelas estén las líneas del gráfico, más distorsionado aparecerá éste. El gráfico puede llegar a estar muy distorsionado (como visto a través de un objetivo fotográfico ojo de pez) o poco distorsionado (como visto a través de la lente de un teleobjetivo). Si se aumenta la distorsión, se aumentará el efecto del gráfico y se enfatizarán sus datos acercándolos al espectador.

Cambia la longitud de los ejes.

Inclina el gráfico en tres dimensiones.

Mostrar gráfico	Si se desactiva el recuadro, quedará una visión de líneas del gráfico que reflejará los cambios del original.
-----------------	---

Deshacer	Deshace los cambios introducidos y regresa a la ventana del gráfico.
Renovar	Renueva el gráfico para reflejar los cambios.

### ***Ventana de diálogo Patrón a todo color***

Permite especificar el tamaño de los mosaicos que componen el patrón. Asimismo, permite mover todo el patrón en el interior del objeto relleno y escalonar los mosaicos de patrón .

Opciones de la ventana de diálogo  
Recuadro de simulación

Muestra un mosaico del patrón seleccionado.

Lista de patrones

Si pulsa el ratón sobre la flecha de la esquina superior derecha de la ventana de simulación, aparecerá una lista de patrones. Pulse sobre el patrón que desee utilizar y seleccione Aceptar. Si escoge el comando Cancelar, se ignorará la selección efectuada y se cerrará la lista.

Archivo muestra un menú para borrar e importar patrones.

Guardar relleno actual      Guarda el patrón de relleno actualmente visualizado en la ventana de simulación. Utilice este comando para añadir un patrón creado a partir de un gráfico importado a la lista de patrones.

Borrar elemento      Borra de la lista el patrón seleccionado.

Importar patrón      Muestra otra ventana de diálogo que permite crear un patrón a partir de un gráfico importado.

Cargar

Abre una ventana de diálogo en la que se pueden seleccionar los patrones por su nombre. Véase la Ventana de diálogo Cargar patrón.

Importar

Muestra otra ventana de diálogo que permite crear un patrón a partir de un gráfico importado.

Pequeño, Medio, Grande

Selecciona un tamaño de mosaico de patrón de 0.25x0.2, 50.50x0.50 o 1.00x1.00 pulgada.

Mosaico

Muestra controles adicionales para modificar el tamaño y la posición de los mosaicos que componen el patrón.

Anchura/Altura      Especifica un tamaño de mosaico de patrón de hasta 3x3 pulgadas.

Si desea utilizar otra unidad de medida, abra la lista de unidades y seleccione la unidad que desee.

Desplazamiento del primer mosaico      Especifica la posición del primer mosaico en relación con la esquina superior izquierda de la silueta del objeto.

Desplazamiento de fila/columna      Mueve la cantidad especificada filas o columnas.



# STATISTIC<sup>2</sup>

## Introduction

1. [Definitions](#)
2. [Observations](#)
3. [Data](#)
4. [Statistics](#)
5. [Variables](#)
6. [Variation](#)
7. [Sampling](#)
8. [Models](#)
9. [Inference](#)

### Basic Definitions

We start with a number of definitions, which will be explained in more detail in subsequent sections.

Statistics studies statistical techniques

Statistical techniques map data into statistics

Data are coded observations

Let us first discuss the concept of observation

## What are observations

We shall assume we can all agree on what we mean by *observations*. Some discussion points are definitely worth mentioning, however. They are sort of philosophical, so you don't have to agree with them.

### Observations need not be scientific

Indeed, everybody observes, and not everybody is engaged in science.

### Observations need not be empirical

In fact, in quite a few disciplines people observe by using *introspection* or *intuition* or *empathy*. I am not sure what these are, but they do not seem to involve the senses, and thus they are not empirical. Nevertheless there seems to be no obvious reason why we should exclude them from the domain of observations.

### Observations are not sense impressions

The point I am trying to make is that observations are *interpreted* and, at least in principle, *public*. They can be communicated to others, although often they are not. Thus observations are already *coded* by our brains, and filtered through language and cognition.

### Observations need not be quantitative

Indeed, it can be argued that observations are *never* quantitative. We always have to perform some sort of operation on our observations to make them quantitative.

Example:

We observe the qualitative fact that a particular object is between two particular marks on our measuring device.

Example:

We observe the qualitative fact that student A gave response "a" to multiple choice item 1, response "b" to multiple choice item 2, ... , response "d" to multiple choice item n.

---

<sup>2</sup> Page author: [Jan de Leeuw \[deleeuw@stat.ucla.edu\]](mailto:Jan.de.Leeuw@stat.ucla.edu)  
<http://www.stat.ucla.edu/textbook/introduction/>

# What are Data ?

## Data are coded observations

We have already seen that observations are coded, at least implicitly. Data are coded explicitly. Thus the person collecting the data makes a *design*, or *protocol*, or *codebook*. This tells her how to transform the observations to data.

We create a codebook for a survey. The codebook tells us how to code the possible observations we collect. Thus we may decide to code income, for instance, in four categories. If we ask the respondent for her religion, we need some rules on how to enter the responses into our data base. We should not only have categories for the major religions, but we should also take into account that somebody might answer "republican" or "visitor from the second house".

We measure height of fathers and sons.

Dose-response.

Statistical techniques are used to analyze data. More formally, let us suppose we know what we mean by *data*. Then a statistical technique is a function defined on the

## Data are coded observations

## Statistics studies statistical techniques

# Variables

10. [What is a variable? \(Statistical Variables\)](#).
11. [Random Variables](#).
12. [Modeling Samples With Random Variables](#).

## What is a variable? (Statistical Variables)

A variable is a quantity which varies over different instances. In Mathematics, variables are often denoted by letters such as  $x$  and  $y$  in the equation  $y = x + 3$ . Here,  $x$  and  $y$  stand in for a family of pairs of values which satisfy the equation. Some members of the family are  $(x,y) = (0,3), (1,4), (2,5)$ , etc.  $x$  and  $y$  vary in the sense that any statement made about the pair is true for every pair in the family, so the statement applies to various instances of  $x$  and  $y$  simultaneously.

A statistical variable varies in another sense.  $X$  might be the weight of a particular person. Repeated weighings of that person yield slightly different values from weighing to weighing. Maybe this is due to error in the measurement device, or maybe the person's weight actually fluctuates; the reason isn't important. What is important is that on any given weighing there is uncertainty about what the measurement will be.

If we make a statement like  $y = x + 3$  about statistical variables we mean that  $x$  is subject to chance variation as in the previous paragraph, and whatever value  $x$  happens to assume, we add 3 to it to obtain  $y$ .  $Y$  is therefore also subject to the same chance variation, but it is still related to  $x$ .

In Statistics, we call statistical variables *random variables*, and refer to the actual value taken on by a random variable as its *realization*. Random variables will be denoted by letters which are underlined. For example, the statement  $P(\underline{X} = 2)$  refers to the probability of the event that random variable  $\underline{X}$  takes on the realization 2. We can be more general and speak of  $P(\underline{X} = x)$ , where " $x$ " stands in for a family of possible values of the realization.  $x$  is then a variable in the sense of variables in Mathematics, while  $\underline{X}$  is a statistical variable.

# Random Variables

A random variable, also called a statistical variable, is a function that maps the outcome of an experiment to a real number. The roll of a die has six possible outcomes corresponding to the six faces. If the face with one dot appears, we say the outcome is 1. If the face with two dots appears, we say the outcome is 2, and so on. By associating each element of the sample space (the set of possible outcomes) with a real number, we form a random variable.

Since there are probabilities associated with the members of the sample space, there are also probabilities associated with potential realizations of any random variable defined on that sample space. In the die example, let  $\underline{X}$  be a random variable showing the outcome. The possible realizations of  $\underline{X}$  are  $\{1, 2, 3, 4, 5, 6\}$  corresponding to the six faces. Each face has probability  $1/6$ , and so the probability distribution of  $\underline{X}$  is

$x$	$P(\underline{X}=x)$
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Now consider  $\underline{Y}$ , another random variable defined on the same sample space. Let  $\underline{Y} = 1$  if the outcome is the face with two, four, or six dots, and let  $\underline{Y} = 0$  if the outcome is the face with one, three, or five dots. ( $\underline{Y}$  could also have been defined in terms of  $\underline{X}$ :  $\underline{Y} = 1$  if  $\underline{X}$  is even, and  $\underline{Y} = 0$  if  $\underline{X}$  is odd.) The probability distribution of  $\underline{Y}$  is

$y$	$P(\underline{Y}=y)$
0	1/2
1	1/2

$P(\underline{Y}=1)$  is the sum of the probabilities of all the outcomes for which  $\underline{Y}$  would be one. There are three of them, each with probability  $1/6$ , so the total is  $3/6$ .

The probability distribution of a random variable depends on the probabilities associated with the outcomes in the underlying sample space, and on the way the random variable maps those outcomes to realizations. If the probabilities associated with the sample space of the die change so that  $P(\underline{Y}=1) = P(\underline{Y}=3) = P(\underline{Y}=5) = 1/4$  and  $P(\underline{Y}=2) = P(\underline{Y}=4) = P(\underline{Y}=6) = 1/12$ , the distribution of  $\underline{Y}$  changes to:

$y$	$P(\underline{Y}=y)$
0	3/4
1	1/4

On the other hand, if the probabilities of the six faces are once again all  $1/6$ , but  $\underline{Y}$  is redefined so that  $\underline{Y} = 1$  if the face showing is two or four, and  $\underline{Y} = 0$  otherwise, the distribution of  $\underline{Y}$  is:

$y$	$P(\underline{Y}=y)$
0	2/3
1	1/3

In practice, one usually models the potential outcomes of an experiment as a random variable without bothering to first define the sample space, then define the random variable, and finally calculate its distribution. This is very natural since we are often interested in numerical codings of the outcomes rather than the outcomes themselves. With the die, we care about the *number* of dots on the face showing; not the actual pattern.

## Modeling Samples With Random Variables

Members of a sample can be modeled as [random variables](#). Before sampling is done, each draw from the population is regarded as a random variable with an associated probability distribution. The distribution is determined by probabilities associated with the underlying sample space, and by the [sampling design](#). After sampling, we observe the sampled values. These are the realizations of the random variables representing the sample.

For example, suppose we draw a sample of size two with replacement from a population of three chips numbered 1, 2, and 3. We let  $X_1$  and  $X_2$  be random variables representing the numbers of the first and second chips drawn, respectively. The probability distribution of  $X_1$  is  $P(X_1 = 1) = P(X_1 = 2) = P(X_1 = 3) = 1/3$ , since each of the three chips is equally likely to be drawn first. The probability distribution of  $X_2$  is the same, since the sampling is with replacement. After the experiment, we note that  $X_1$  was realized as, say,  $x_1 = 2$ , and  $x_2 = 3$ .

### Conventional

- [Variance](#)
- [Mean Absolute Deviation](#)
- [Interquartile Range](#)

### Robustness

## The Variance

### Discussion

A measure of [location](#) is usually not sufficient to describe the distribution of a [random variable](#). We therefore introduce the variance of a random variable as a measure of spread of the distribution. In several widely used [probability models](#), the [mean](#) along with the variance of a random variable are adequate to describe its distribution.

### Definition

The variance of a [random variable](#) is defined as

where we define: When we have a sample  $x_1, \dots, x_n$  of observations, we define the sample variance as

$$s^2(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}$$

## Sampling

13. [Populations and Samples](#).
  - [Finite Populations](#).

- [Infinite Populations.](#)
- 14. [Parameters and Statistics.](#)
- 15. [Sampling from Infinite Populations.](#)
- 16. [Sampling from Finite Populations.](#)
- 17. [Sampling Design](#)
- 18. Sampling Designs for Finite Populations:
  - [Sampling With and Without Replacement.](#)
  - [Simple Random Sampling.](#)
  - [Simple Random Sampling With and Without Replacement.](#)
- 19. Sampling Designs for Infinite Populations.
- 20. [Resampling.](#)

## Populations and Samples

A statistical population is a set of items about which we have a question. The items might be people or things, real or hypothetical. For example, we might be interested in the birth weights of babies born at the UCLA Medical Center during the past 10 years, ending today. The characteristic of interest is weight, and if we went to the records department at the hospital, we could actually tabulate this list of birth weights. Each weight is associated with a baby, that person actually exists today, and there are a finite number of such persons.

As another example, consider the population of all babies that will be born at the UCLA Medical Center in the *next* 10 years. This is a hypothetical population; we can't go down to the records department of the hospital and collect a list of birth weights for these babies even if we wanted to. Nonetheless, these babies also constitute a population even though we don't even know how many of them there are.

In either case we may study some aspect of baby birth weights by collecting a sample from the population. A sample is a subset of the population, usually a small subset. Inferences are then made about the population based on the sample. In the first case, a sample could be obtained by recording birth weights only for babies born in the past week. In the second case, we could plan to collect birth weights over the next week. Each is a sample from the relevant population.

The scheme by which items are chosen for the sample is called the [sample design](#), and it is of crucial importance. The sample must be representative of the population with respect to the characteristic(s) of interest if inferences about the population are to be valid.

## Finite Populations

A finite population is a population containing only a finite number of items. The population of babies born today at the UCLA Medical Center is an example of a finite population. Population parameters like mean weight are directly calculable from the items themselves, at least in principle.

## Infinite Populations

An infinite population is a population with an infinite number of elements. This may sound obvious, but it actually implies something more: an infinite population is the set of all items that could be described by a particular probabilistic mechanism. The population of all babies that will ever be born at the UCLA Medical Center is infinite in the sense that we think of a continuing stream of babies whose weights may be described by particular probability distribution.

Consider the following quote from Harry Roberts' 1965 paper, *Probabilistic Prediction*:

*Statisticians have long distinguished 'finite populations' from 'infinite populations' or 'processes'. The question can arise as to whether parametric inference should be directed to a finite population, from which a sample has been randomly chosen, or to an underlying process or infinite population, physical or conceptual, that is assumed to have given rise to the finite population itself...*

# Parameters and Statistics

Any quantity computed using all population values is called a *population parameter* (or just *parameter* for short). Any quantity computed using the members of a sample is called a *statistic*. The value of a population parameter never changes; it is a fixed quantity. However, the value of a statistic will change depending on which items from the population are chosen for the sample.

The average birth weight of all babies born at the UCLA Medical Center in the past 10 years is the same whether we compute it today or tomorrow or next week is

$$\frac{1}{N} \sum_{i=1}^N x_i$$

where  $N$  is the number of items in the population, 10,000 in this case. The computation always uses the same 10,000 values.

Suppose a sample of 100 babies is drawn from the complete set of 10,000. The *sample mean* is the arithmetic average of the 100 ( $n = 100$ ) sample values:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

A new sample drawn tomorrow will yield a new sample mean. In general, the sample mean will vary from sample to sample. If the sample is chosen randomly, then the sample mean is also random in the sense that it varies with the sample.

Often, we make inferences about population parameters based on sample statistics. The stability of our inference depends directly on how stable our statistic is across the different samples from which it may be computed. A statistic which always yields a value in a narrow range despite the fact that it is derived from different samples, is said to be *stable*. In addition, if that range is centered about the true value of some population parameter, the statistic is said to be *unbiased* for that parameter. Stability and unbiasedness can be assessed by looking at the [sampling distribution](#) of the statistic, which describes its probabilistic behavior.

## Sampling from Infinite Populations

When sampling from an [infinite population](#), every draw has the same probabilistic structure. More formally, let random variables  $X_1, X_2, \dots, X_n$  represent the sample. We say that the  $X_i$ 's are *identically distributed* if they all have the same probability distribution. Moreover, if the realization of  $X_i$  does not affect probabilities associated with realizations of any other  $X_j$ 's, the random variables are said to be *independent*. This contrasts with the situation for a [sampling from a finite population](#), where outcomes of early draws affect later ones.

## Sampling from Finite Populations

There are two ways in which samples obtained from [finite populations](#) differ from those obtained from [infinite populations](#). First, when sampling successively and without replacement from a finite population, the probabilistic structure of items drawn later depend on the outcomes of earlier draws. For example, suppose 30 babies are born today, and we select a sample of size 10. On the first draw, every baby has probability  $1/30$  of being selected. On the second draw, the remaining 29 babies have probability  $1/29$  of being selected, and the baby selected on the first draw can't be chosen again. Therefore what may happen on the second draw is affected by what happens on the first draw. The draws are not independent, and this will affect our estimates of population parameters.

The second difference is most easily understood in terms of random variables. Let  $X_1$  represent the weight of the baby drawn first,  $X_2$  represent the weight of the baby drawn second, and so on. *It is important to understand that in the finite population context, we regard the weights of the 30 babies in the population as fixed quantities.* When  $X_1$  is realized by drawing the first baby, the randomness that makes  $X_1$  a random variable is due to the random selection of one baby from among 30. The randomness is not due to any indeterminacy concerning that particular baby's weight.

For example, consider Baby Fred, one of the 30 babies. Suppose Baby Fred weighs eight pounds, and is the only baby that weighs eight pounds. The realization of  $X_1$  as eight pounds would be due to the selection of Fred as baby number one, not to the realization of Fred's weight as eight pounds.

There are circumstances under which we would not regard the 30 baby weights as fixed. Then, there would be two sources of randomness in this sampling procedure, and both would influence the probability distribution of the  $X_i$ 's. That is not the case here because our population is considered to be already determined by the time we arrive. However, this is just an assumption often made implicitly, and may not be appropriate in all situations.

## Sampling Design

The term *sampling design* refers to the scheme used to select elements of the population for the sample. In order to ensure that inferences about the population made using the sample are valid, the sample should be representative of the population with respect to the characteristics of interest. More precisely, the probabilistic structure of the sample should depend on the population parameter of interest in a known way. That way will be determined by (1) the probability model describing the population, and (2) the sampling design. Statistics computed from such a sample will have [sampling distributions](#) that depend on and provide information about the parameter. This information can be presented using techniques of estimation, in which we make statements about parameters using knowledge provided by samples and statistics.

## Sampling With and Without Replacement

When selecting a sample of size  $n$  from a population containing  $N$  items, we make  $n$  successive draws from the pool of  $N$  items. We may elect to set a drawn item aside after it is chosen, or to put it back into the original pool of items.

If we set the item aside, it can never be drawn more than once, and moreover, the probabilities with which remaining items are chosen on the next draw change. This is sampling *without replacement*, and is what we mean when we agree to select a sample of size  $n$  at random from a population of size  $N$ . (The latter statement makes it sound as though we are choosing  $n$  items simultaneously rather than sequentially. In fact, making  $n$  simultaneous draws is the same as making  $n$  sequential draws without replacement: no item can appear more than once in the sample.)

If we put the items back after each draw, they are available to be drawn again later, and the probabilities associated with selecting them remain the same on all  $n$  draws. This is sampling *with replacement*, and is the simplest of all sampling schemes.

## Simple Random Sampling

Under simple random sampling one starts with a population of  $N$  items, and samples  $n$  of them, either with or without replacement. The defining characteristic of this sampling scheme is that all potential samples have equal probability of being the sample actually chosen. For example, if Fred, Sally, and Bill are a "population" of three students, and we select a sample of size one, the possible samples are (Fred), (Sally) and (Bill). Each has a  $1/3$  chance of being the sample which is selected, so this is a simple random sampling scheme. If we were to select a sample of size two, the possible samples are (Fred and Sally), (Fred and Bill), and (Sally and Bill). Each of these samples also has probability  $1/3$  of being the sample chosen, so this is also an SRS scheme.

# Simple Random Sampling With and Without Replacement

Consider a hat containing three chips: one red, one blue, and one yellow. We wish to choose a sample of size two. On the first draw, the probability associated with drawing the red chip is  $P(\text{Red}) = 1/3$ . So are the probabilities associated with drawing the other chips:  $P(\text{Blue}) = 1/3$  and  $P(\text{Yellow}) = 1/3$ . There are three chips in the hat, and all three chips are equally likely. No matter which chip is actually selected on the first draw, the situation remains unchanged on the second draw because the chip chosen on the first draw is put back.

Now let's look at the samples we could get:

(RR)      (RB)      (RY)      (BB)      (BR)      (BY)      (YY)      (YR)      (YY)

The first member of each pair represents the color of the first chip drawn, and the second member represents the color of the second chip drawn. It is important to note here that we do consider the order of the colors to be important. In other words, the outcome (RB) is considered different from (BR). (If this were not the case, there would be only six pairs in the sample space, rather than nine.)

Each of the nine samples is equally likely to be the one we obtain, and so this selection procedure of drawing two chips in succession with replacement can be called simple random sampling with replacement.

Suppose now that we sampled without replacement: when a chip is drawn it is permanently removed from the pool. The sample space now is:

(RB)      (RY)      (BY)

There are now just three pairs which are candidates to be the sample. However, once again they all have an equal chance to be the pair that is actually selected for the sample, and so this procedure can also be called simple random sampling, but this is simple random sampling *without* replacement.

## Models

1. [Modeling Populations.](#)
2. [Modeling Samples.](#)

## Modeling Populations

A model for a population is an idealized description of some aspect of the population. Often, we use probability models to describe values associated with items in a population. For example, to say that SAT verbal scores are normally distributed with mean 475 and standard deviation 100 is to describe the SAT scores of the people in the population. The proportion of people scoring in a particular range is given by the area under the normal curve and over that part of the number line corresponding to the range.

## Modeling Samples

Members of a sample can be modeled as [random variables](#). For instance, let  $X$ ,  $Y$ , and  $Z$  be a sample of size three from some population. The probability distributions describing the behavior of  $X$ ,  $Y$ , and  $Z$  are determined by (1) the probability model describing the population, and (2) the [sampling design](#).

Some sampling designs are transparent: they do not affect the probability distributions of the random variables representing the sample. In this case the distributions of  $X$ ,  $Y$ , and  $Z$  are just given by the population model. One example of this is [sampling from infinite populations](#).

Some sampling designs are not transparent: they do affect the probability distributions of the random variables representing the sample. This occurs when [sampling from finite populations](#) because the outcomes of earlier draws affect probabilities governing later draws.



# Inference

3. [Inductive and Deductive Inference](#) .
4. [Inferential Stability](#).
5. [Estimation](#).
  - Sampling Distributions.
    - [What is a Sampling Distribution?](#).
    - [Sampling Distributions Provide Information About Population Parameters](#).
  - [Confidence Intervals](#).
  - [Hypothesis Testing](#).

## Inductive versus Deductive Inference

Inference is the process by which we draw conclusions about the state of some set of things based on knowledge about another set of things. Inference can be either *inductive* or *deductive*. It is inductive if we draw conclusions about a general case based on information from a specific case. For instance, say we are driving home on a Friday afternoon, and observe the the cars on a freeway overpass in gridlock. We conclude that the freeway is jammed, and elect to take a different way home. We have made an *inference* about the condition of the freeway based on some *sample data*, the condition of that poriton of the freeway which is visible on the overpass. This reasoning is inductive.

Deductive reasoning goes the opposite way: we draw conclusions about a specific case based on knowledge about a more general case. On our drive home suppose we hear a traffic report stating that the freeway is jammed. We conclude that, in particular, the portion of the freeway we must travel is jammed and elect to find an alternate route. We have made an inference about the condition of the portion of the freeway of concern to us based on information about the state of the freeway as a whole.

A more conventional example is that of throwing a die. You are told that the probabilities associated with each face are proportional to the number of dots on each face, and asked to predict which face will turn up. You predict the outcome will be a six, since it has the highest probability of occurance. This is your deductive inference.

On the other hand, suppose you are not told anything about the die. You are shown a list of outcomes of 50 throws, and asked to estimate the probabilities associated with each of the six possible outcomes. You observe that of the 50 throws ten resulted in a one showing, 20 resulted in a two, and 20 in a three. Your inductive inference is that the probabilities associated with the six faces are 0.2, 0.4, 0.4, 0, 0 and 0 in that order. This seems reasonable because these probabilities and the outcomes on the list are consistent with one another. (In fact, among all possbile choices for the six-tuple of probabilities above, (0.2, 0.4, 0.4, 0, 0 and 0) is the most consistent in the sense that it would be expected to produce exactly what was observed.)

Statistical reasoning is inductive, while problems in probability are generally deductive. However, statisticians use the laws of probability to understand relationships between what they observe and the conditions (as expressed by probabilities of various events) which lead to what they observe. Conclusions can then be drawn about *conditions* based on *observations*. But how reliable are those conclusions? After all, the same set of conditions (probabilitites) may well produce a different set of observations if the die throwing experiment is repeated. This variability relates directly to the notion of statistical [stability](#).

## Inferential Stability

Our best guess about the six probabilities governing the outcomes in the [die tossing](#) experiment was very particular to the 50 outcomes on our list. If the list had instead contained 50 sixes, we would've drawn a very different conclusion. It matters which list we use.

The good news is that the *true* probabilities *never* change. The only reason we might see different lists is that there is chance variation in the outcomes even though the true probabilities are fixed. For example, suppose the true probability of getting a one is 0.2. This does not guarentee that exactly 20 percent of the tosses will

come up one. It only means that in the long run the proportion of ones will be close to 0.2. (This is known as the *Law of Large Numbers*.) An experiment in which the die is thrown 50 times may result in only five ones; an identical experiment using the same die performed an hour later might result in 12 ones for no reason other than chance variation.

When we base our inference on one such experiment, we are not 100 percent sure our inference is correct because the experiment on which the inference is based is subject to chance variation. Assessing the reliability of our inference means measuring this chance variation and its potential effect on our conclusions. How can this be done?

We can quantify the variability of our estimate of the probability of tossing a one as follows: Conduct a large number, say 100, of identical experiments: each experiment involves tossing the die 50 times, and recording the proportion of ones observed. At the end we have 100 proportions. Calculate the sample variance of these 100 numbers. This measures the variability of the outcome of our experiment.

To see how stable our inference is, we might go one step further. For each experiment, state the inference we would draw from it, and then examine the variance of *these* results. In this case, our inference is that the true probability of rolling a one is equal to the observed proportion of ones, so this second variance will equal the first. A small variance means that in a *future* experiment, the result is not likely to deviate very much from the average one would get if the experiment were repeated many times. A large variance means that the result is hard to predict; it may deviate quite a bit.

## Estimation

The activity in which we make inductive inferences about population parameters using sample data is called *estimation*. Often, estimation is done by producing an estimate of the parameter of interest, and an accompanying measure of the estimate's stability. Let's look at a simple example involving the population mean.

Suppose we are interested in birth weights of all babies born at the UCLA Medical Center in the past ten years. These babies are the population, and we want to know what the mean weight is. If we had the resources, we could compute the true average of all the weights:

where  $N$  is the number of babies in the population. Our job is to estimate on the basis of what we learn from a sample of size  $n$  drawn from the population.

A [simple random sample](#) of size  $n$  is drawn, and we compute the sample mean,

$\bar{x}$  is a random quantity since it is computed from a random sample. Both the sample and any quantities computed from it are subject to chance variation. We could offer  $\bar{x}$  as our estimate of  $\mu$ , but knowing that a new sample taken under the same conditions could generate a different  $\bar{x}$ , should make us pause. It would be better to offer a range of values, and better yet to include the probability that stated range includes the true value of the population mean. This is precisely what a [confidence interval](#) does. This method of estimation provides the user with information about the parameter, and also with a measure of reliability.

Another related form of estimation is testing of statistical hypotheses. The question answered here is slightly different than the one above. Instead of estimating the true value of  $\mu$ , we would use a hypothesis test to answer a question of the form, "Is the true value of  $\mu$  equal to seven pounds as opposed to being larger?" In other words, we already have a suspicion about where  $\mu$  lies, and we want to know whether the data in our sample support or refute our theory. The result of a hypothesis test is also reported with an associated measure of reliability that is related to that of the confidence interval.

How are confidence intervals and hypothesis tests related? Both depend on the [sampling distribution](#) of the estimator. It turns out that for a specified level of reliability, the confidence interval derived from our sample contains exactly those values theorized for  $\mu$  which a hypothesis test based on our sample would accept.

## What is a Sampling Distribution?

The sampling distribution of a [statistic](#) is just its probability distribution. Since statistics are computed from randomly drawn samples, they vary with the sample actually obtained. We could derive the sampling distribution of any statistic as follows: list all the samples in the sample space. For each one, calculate the

value of the statistic computed from that sample, and the probability that that sample is the one actually obtained. Collapse the table so that each distinct value of the statistic appears only once, and the probability associated with that value is the sum of the probabilities associated with the samples which yielded that particular value.

A small example should make things clear. Suppose we roll a fair die twice, and the statistic we compute is the sum of the two outcomes. The sample space is:

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

The probability associated with each of the 36 outcomes in the sample space is  $1/36$ . The statistic associated with each outcome in the sample space is:

2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	10
6	7	8	9	10	11
7	8	9	10	11	12

and we assign a probability of  $1/36$  to each entry in this table. Finally, we construct a new table of statistic values and probabilities that has each value of the statistic listed just once:

Total	Probability
2	$1/36$
3	$2/36$
4	$3/36$
5	$4/36$
6	$5/36$
7	$6/36$
8	$5/36$
9	$4/36$
10	$3/36$
11	$2/36$
12	$1/36$

This is the sampling distribution of the total.

## Confidence Intervals

A confidence interval is a range of values which includes the parameter being estimated, with a specified (and stated) probability. This probability is called the *coverage probability*, and when we speak of a "95 percent

confidence interval" we mean: the probability of the event, "the interval contains the parameter being estimated", is 0.95.

The fact that there is a probability associated with this event suggests that the event is a random phenomenon. A confidence interval is computed from a random sample, and so the interval itself is subject to chance variation. A 95 percent confidence interval for a certain parameter computed from one sample may not be the same as a 95 percent confidence interval for the same parameter computed from a different sample. However, both have probability 0.95 of containing, or *covering*, the parameter.

Another quantity often mentioned in connection with confidence intervals is *significance level*. The significance level, denoted  $\alpha$ , is one minus the coverage probability. It is traditional to specify the coverage probability in terms of the significance level: a 95 percent confidence interval is a  $(1 - \alpha)$  percent confidence interval with  $\alpha = 0.05$ .

For a given coverage probability short intervals are better than long ones, since they tell us more precisely where the unknown parameter is likely to be. We can increase the coverage probability by widening the interval. Of course, if we widen the interval too much we diminish its usefulness. Reporting that a 100 percent confidence interval for the parameter runs from to doesn't help anybody.

Confidence intervals are constructed from random samples with the help of *point estimates*. A point estimate is a statistic around which the confidence interval sits. If the point estimate is always near the parameter being estimated, short intervals can be used. This is because the parameter can be covered by going only a short distance from the estimate. If the distance between the estimate and the parameter is large, or varies a lot, wider intervals are required to achieve a given coverage probability.

Information about variability of this distance is captured in the [sampling distribution](#) of the point estimator. For example, suppose we want to estimate the mean of a normally distributed population with known standard deviation, say  $\sigma$ . We take a sample of size  $n$ , and use the sample mean  $\bar{x}$ , as our point estimate of the population mean. The [sampling distribution of the mean](#) also follows a normal distribution, with mean  $\mu$ , but the standard deviation of the sampling distribution is  $\frac{\sigma}{\sqrt{n}}$ .

From what we know about the normal distribution, we can say that the probability falls within two standard deviations of  $\mu$  is 0.95. (This is the [68-95-99 rule](#).) So  $\bar{x}$  will be no more than a distance of  $2 \frac{\sigma}{\sqrt{n}}$  apart with

$$\left( \bar{x} - \frac{2\sigma}{\sqrt{n}}, \bar{x} + \frac{2\sigma}{\sqrt{n}} \right)$$

probability 0.95. This determines our 95 percent confidence interval as a range of values:

## Hypothesis Testing

A hypothesis test is a procedure for determining whether data in a sample are consistent with a particular hypothesis concerning an unknown parameter of the population from which the sample came. A simple example is testing whether a coin is fair. We toss the coin ten times, and note the proportion of heads. The sequence of heads and tails we obtain is the sample data, and the number of heads divided by the number of tosses (10) is our statistic. If the proportion of heads is around 0.5, we conclude that the evidence is in agreement with the hypothesis that the true probability of a head is 0.5.

But what do we mean by "around 0.5"? Suppose the proportion of observed heads is 0.6. That seems reasonably close to 0.5, but what about 0.7 or 0.8 or 0.9? How far away from what we expect should the observed proportion be before we conclude the coin is not fair? The answer lies with the [sampling distribution](#) of the our statistic.

We must choose a statistic with a sampling distribution that depends on the true value of the parameter. In this case the parameter is the probability of a head. Then the statistic carries information about the parameter because some values of the statistic will be more likely than others under the hypothesis that the coin is fair. For example, if the coin is fair, an observed proportion of 0.6 is more likely than an observed proportion of 0.1.

If the coin is fair, we expect the discrepancy between 0.5 and the observed proportion of heads to be small. One way to judge whether a given discrepancy is small or not is to look at the probability of observing a discrepancy at least as large as the one given. This is called the *p*-value. Suppose our experiment resulted in an observed proportion of 0.7. This is a distance of 0.2 from what we expect under the hypothesis that the true probability of a head is 0.5. The probability of observing a discrepancy of at least 0.2 is the probability of an observed proportion of 0.7, 0.8, 0.9, 1.0, 0.3, 0.2, 0.1, and 0.0. *The last four values must be included because*

they too represent discrepancies of at least 0.2! When these probabilities are computed under the assumption that the coin is fair, the sampling distribution of the observed proportion tells us something about whether the coin is fair. These probabilities are:

Proportion	Probability
0.0	0.000977
0.1	0.009766
0.2	0.043945
0.3	0.117188
0.7	0.117188
0.8	0.043945
0.9	0.009766
1.0	0.000977

The  $p$ -value is the sum: 0.3438.

What do we conclude? We can make the following statement: If the probability of a head is really 0.5, the probability that the observed proportion of heads is at least as extreme as our result is 0.34. If we feel that an event occurring with probability 0.34 is a rare event, then we should reject the hypothesis that the coin is fair. Now most people would not consider something that happens 34 percent of the time to be a rare event, so in this case we would not be able to say that the result of our experiment casts doubt on the hypothesis. Suppose, however, that the  $p$ -value had been 0.0001. Our statement would be: If the probability of a head is really 0.5, the probability that the observed proportion of heads is at least as extreme as our result, is 0.0001. This means that under the assumption the coin is fair, we would get a result like ours only one time in 10,000 experiments. This does cast doubt on the "if" clause above, and would cause us to reject the notion that the coin is fair.

Statisticians usually consider "rare" events to be those which occur with probability 0.05 or less, but anyone carrying out a hypothesis test is free to decide what this cut-off probability will be. The cut-off probability is called the *significance level*, and is denoted by  $\alpha$ . (This is the same that comes up in the context of confidence intervals.) In the example above, we would state our conclusion by saying we do not reject the hypothesis that the coin is fair at significance level 0.05.

In the jargon of hypothesis testing, our example would look like this:

Since  $0.3438 > 0.05$ , we do not reject at  $\alpha = 0.05$ .

# Basic Definitions

## Contents

[Statistical Inference](#)

[Experiment](#)

[Experimental \(or Sampling\) Unit](#)

[Population](#)

[Sample](#)

[Parameter](#)

[Statistic](#)

[Sampling Distribution](#)

[Estimate](#)

[Estimator](#)

[Estimation](#)

### **Statistical Inference**

Statistical Inference makes use of information from a sample to draw conclusions (inferences) about the population from which the sample was taken.

### **Experiment**

An experiment is any process or study which results in the collection of data, the outcome of which is unknown. In statistics, the term is usually restricted to situations in which the researcher has control over some of the conditions under which the experiment takes place.

#### **Example**

Before introducing a new drug treatment to reduce high blood pressure, the manufacturer carries out an experiment to compare the effectiveness of the new drug with that of one currently prescribed. Newly diagnosed subjects are recruited from a group of local general practices. Half of them are chosen at random to receive the new drug, the remainder receiving the present one. So, the researcher has control over the type of subject recruited and the way in which they are allocated to treatment.

### **Experimental (or Sampling) Unit**

A unit is a person, animal, plant or thing which is actually studied by a researcher; the basic objects upon which the study or experiment is carried out. For example, a person; a monkey; a sample of soil; a pot of seedlings; a postcode area; a doctor's practice.

### **Population**

A population is any entire collection of people, animals, plants or things from which we may collect data. It is the entire group we are interested in, which we wish to describe or draw conclusions about.

In order to make any generalisations about a population, a sample, that is meant to be representative of the population, is often studied. For each population there are many possible samples. A sample statistic gives information about a corresponding population parameter. For example, the sample mean for a set of data would give information about the overall population mean.

It is important that the investigator carefully and completely defines the population before collecting the sample, including a description of the members to be included.

#### **Example**

The population for a study of infant health might be all children born in the U.K. in the 1980's. The sample might be all babies born on 7th May in any of the years.

### **Sample**

A sample is a group of units selected from a larger group (the population). By studying the sample it is hoped to draw valid conclusions about the larger group.

A sample is generally selected for study because the population is too large to study in its entirety. The sample should be representative of the general population. This is often best achieved by random sampling. Also, before collecting the sample, it is important that the researcher carefully and completely defines the population, including a description of the members to be included.

#### **Example**

The population for a study of infant health might be all children born in the U.K. in the 1980's. The sample might be all babies born on 7th May in any of the years.

#### **Parameter**

A parameter is a value, usually unknown (and which therefore has to be estimated), used to represent a certain population characteristic. For example, the population mean is a parameter that is often used to indicate the average value of a quantity.

Within a population, a parameter is a fixed value which does not vary. Each sample drawn from the population has its own value of any statistic that is used to estimate this parameter. For example, the mean of the data in a sample is used to give information about the overall mean in the population from which that sample was drawn.

Parameters are often assigned Greek letters ( e.g.  $\mu$  ), whereas statistics are assigned Roman letters (e.g. s).

$\sigma$

#### **Statistic**

A statistic is a quantity that is calculated from a sample of data. It is used to give information about unknown values in the corresponding population. For example, the average of the data in a sample is used to give information about the overall average in the population from which that sample was drawn.

It is possible to draw more than one sample from the same population and the value of a statistic will in general vary from sample to sample. For example, the average value in a sample is a statistic. The average values in more than one sample, drawn from the same population, will not necessarily be equal.

Statistics are often assigned Roman letters (e.g. m and s), whereas the equivalent unknown values in the population (parameters ) are assigned Greek letters ( e.g.  $\mu$  and  $\sigma$  ).

$\sigma$

$\mu$

#### **Sampling Distribution**

The sampling distribution describes probabilities associated with a statistic when a random sample is drawn from a population.

The sampling distribution is the [probability distribution](#) or [probability density function](#) of the statistic.

Derivation of the sampling distribution is the first step in calculating a confidence interval or carrying out a hypothesis test for a parameter.

#### **Example**

Suppose that  $x_1, \dots, x_n$  are a simple random sample from a normally distributed population with expected value and known variance.

Then the sample mean is a statistic used to give information about the population parameter is normally distributed with expected value and variance

#### **Estimate**

An estimate is an indication of the value of an unknown quantity based on observed data.

More formally, an estimate is the particular value of an estimator that is obtained from a particular sample of data and used to indicate the value of a parameter.

#### **Example**

Suppose the manager of a shop wanted to know the mean expenditure of customers in her shop in the last year. She could calculate the average expenditure of the hundreds (or perhaps thousands) of customers who bought goods in her shop, that is, the population mean. Instead she could use an estimate of this population mean by calculating the mean of a representative sample of customers. If this value was found to be £25, then £25 would be her estimate.

**Estimator**

An estimator is any quantity calculated from the sample data which is used to give information about an unknown quantity in the population. For example, the sample mean is an estimator of the population mean. Estimators of population parameters are sometimes distinguished from the true value by using the symbol 'hat'. For example,

= true population standard deviation

= estimated (from a sample) population standard deviation

**Example**

The usual estimator of the population mean is

$$\hat{\mu} = \bar{X} = \frac{\sum X_i}{n}$$

where n is the size of the sample and X<sub>1</sub>, X<sub>2</sub>, X<sub>3</sub>, ..., X<sub>n</sub> are the values of the sample.

If the value of the estimator in a particular sample is found to be 5, then 5 is the estimate of the population mean  $\mu$ .

**Estimation**

Estimation is the process by which sample data are used to indicate the value of an unknown quantity in a population.

Results of estimation can be expressed as a single value, known as a point estimate; or a range of values, known as a confidence interval.