

Argentina \$8,90 (recargo al interior \$0,20) / México: \$45

**USERS**

**Microsoft®**

Curso teórico y práctico de programación

# Desarrollador .net

Con toda la potencia  
de **Visual Basic .NET** y **C#**

La mejor forma de aprender  
a programar desde cero



Basado en el programa  
Desarrollador Cinco Estrellas  
de Microsoft

**5**

## Bases de datos

Sistema relacional - Diagrama  
entidad/relación - Normalización

## Lenguaje SQL

Tablas - Agrupar y aplicar funciones  
Consultas de inserción - Actualización



ISBN 978-987-1347-43-8



9 789871 347438





# RedUSERS

COMUNIDAD DE TECNOLOGIA



## EL SITIO Nº1 DE TECNOLOGIA

Noticias al instante // Entrevistas y coberturas exclusivas //  
Análisis y opinión de los máximos referentes // Reviews de  
productos // Trucos para mejorar la productividad //  
Regístrate, participa, y comparte tus opiniones



## SUSCRIBITE

SIN CARGO A CUALQUIERA  
DE NUESTROS NEWSLETTERS  
Y RECIBÍ EN TU CORREO  
ELECTRÓNICO TODA LA  
INFORMACIÓN DEL UNIVERSO  
TECNOLÓGICO ACTUALIZADA  
AL INSTANTE



INGRESÁ A  
[redusers.com/suscribirse-al-newsletter](http://redusers.com/suscribirse-al-newsletter)  
¡Y REGÍSTRATE YA!

[www.reduserspremium.blogspot.com.ar](http://www.reduserspremium.blogspot.com.ar)



Foros



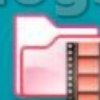
Encuestas



Tutoriales



Agenda de eventos



Videos



¡Y mucho más!



[redusers.com](http://redusers.com)

Seguinos en:



[www.facebook.com/redusers](http://www.facebook.com/redusers)



[www.twitter.com/redusers](http://www.twitter.com/redusers)



[www.youtube.com/redusersvideos](http://www.youtube.com/redusersvideos)





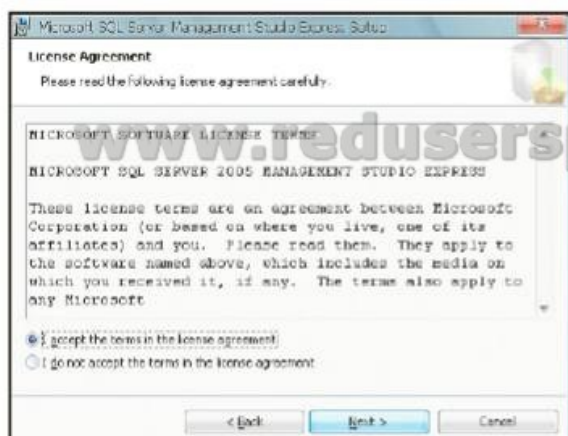
# SQL Server Management Studio Express

Esta herramienta gráfica nos servirá para realizar las distintas tareas de administración que un desarrollador debe llevar a cabo.

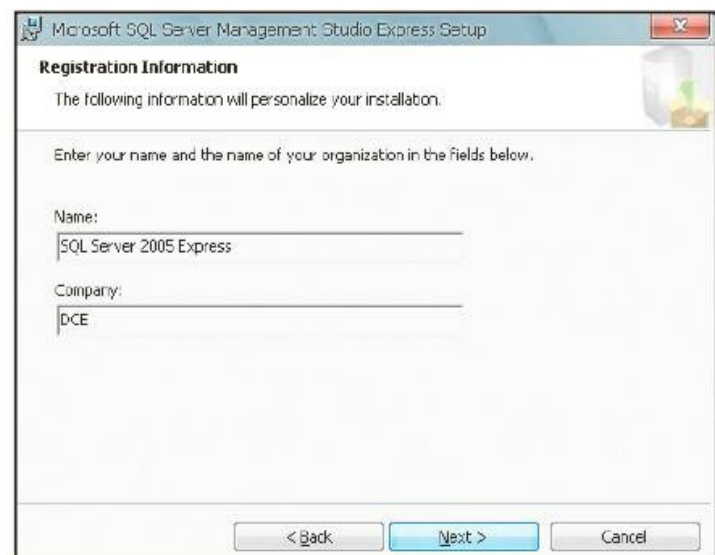
Es posible administrar el servidor SQL Server mediante distintas herramientas que vienen incluidas en el mismo producto. Algunas de ellas se manejan desde la línea de comando, como **SQLCMD**, en tanto que otras son visuales. Management Studio Express es una herramienta visual, característica que nos permitirá administrar nuestros servidores SQL de manera muy sencilla. No viene por defecto con SQL Server 2005 Express, por lo cual debemos descargarla desde el sitio de Microsoft, en la siguiente dirección: <http://msdn.microsoft.com/vstudio/express/sql/download> (el tamaño del archivo es de aproximadamente 40 MB).

## Instalación de Management Studio Express

Ejecutamos el archivo descargado desde la página Web de Microsoft para iniciar la instalación de esta herramienta. Como veremos a continuación, los pasos que debemos seguir son muy sencillos.



En la primera pantalla, se nos da la bienvenida y, a continuación, se muestra el contrato de licencia de usuario final. Lo aceptamos y presionamos el botón Siguiente.



Luego, el sistema muestra la información de registro, y nos pide nuestro nombre y el de la compañía en la que trabajamos. Completamos lo que corresponda y presionamos Siguiente. Acto seguido, el sistema muestra la aplicación que será instalada. Presionamos Siguiente, y se nos notifica en pantalla que está a

! Sólo en inglés

Al momento de escribir esta obra, Microsoft Management Studio Express está disponible, lamentablemente, sólo en el idioma inglés.

punto de iniciarse la instalación. Continuamos con un clic en Siguiente otra vez. Se inicia la instalación de Management Studio Express con todos sus componentes. Al finalizar el proceso, se nos notifica en pantalla, y se crea un icono llamado SQL Server Management Studio Express, para acceder a esta aplicación desde el grupo de programas Microsoft SQL Server 2005.

## Iniciando el programa

Para comenzar a trabajar, activamos la herramienta desde Inicio/Programas/Microsoft SQL Server 2005/Management Studio Express. Se abre en pantalla una ventana que nos pide información para conectarnos a un servidor.

- **Server Type** (Tipo de servidor): Indica a qué tipo de servicio vamos a conectarnos. En Management Studio Express sólo tene-

mos la opción de hacerlo a Database Engine o bien a SQL Server Compact Edition.

- **Server Name** (Nombre del servidor): Indica a qué servidor nos estamos conectando. Podremos identificarlo con su nombre o mediante su dirección IP.

Desplegando el combo de texto, aparecen los servidores que están disponibles. Si en el listado no figura el que precisamos, probamos, en primera instancia, a escribir su nombre o IP en el cuadro de texto.

- **Authentication** (Tipo de autenticación): En esta opción debemos seleccionar qué tipo de identificación usaremos con él cuando deseemos conectarnos. Las opciones son autenticación Windows o SQL. La primera utiliza las credenciales del usuario del sistema operativo, mientras que la segunda solicita el ingreso de un usuario y contraseña.

Luego de haber completado los datos anteriormente mencionados, y si no hubo ningún error (como: No tener acceso al servidor, El

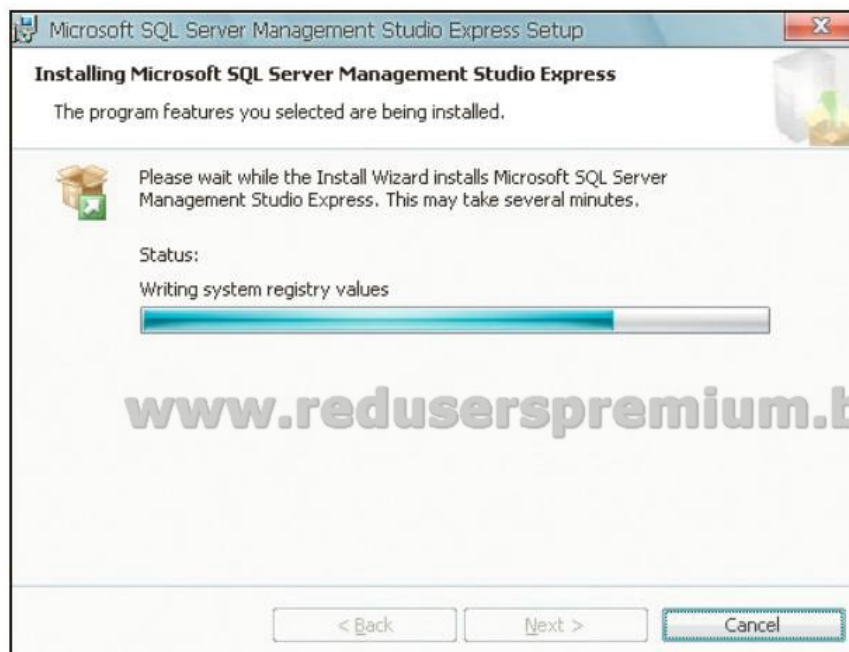


FIGURA 008 | El proceso de descarga e instalación de esta herramienta es muy sencillo y resultará imprescindible para trabajar con bases de datos.





usuario no dispone de permisos de conexión, El Servidor ingresado es incorrecto o no se encuentra), podremos acceder a la herramienta Management Studio para empezar a utilizarla.

## Conocer la interfaz

Management Studio posee una interfaz similar a la de Visual Studio 2005, lo cual nos será de gran ayuda en lo que respecta a su curva de aprendizaje. Si bien hay diferencias considerables con las interfaces de las versiones anteriores de SQL Server, será muy sencillo adaptarnos a esta última, porque es muy intuitiva.

Management Studio Express es una herramienta visual, que permite administrar nuestros servidores SQL de manera muy sencilla.

## Menús de SQL Management Studio

Además de los menús en la aplicación, disponemos de un panel izquierdo llamado



FIGURA 009 | Esta pantalla de inicio es donde indicamos las propiedades de conexión.

## Tabla 2 | Funciones de SQL Management Studio

MENÚ	FUNCIONES PRINCIPALES
FILE	Crear nuevas conexiones, abrir o crear nuevos archivos, guardar archivos nuevos o abiertos
EDIT	Operaciones relacionadas con la edición de texto: copiar, pegar, cortar, buscar, reemplazar, y más
VIEW	Visibilidad de paneles y barras de herramienta de la aplicación
TOOLS	Configuración de la aplicación
COMMUNITY HELP	Información y ayuda de SQL Server 2005, feedback al equipo de desarrollo del producto

EL PANEL OBJECT EXPLORER ES UNO DE LOS MÁS IMPORTANTES DE LA APLICACIÓN. LO UTILIZAREMOS PARA NAVEGAR POR LOS DISTINTOS OBJETOS Y COMPONENTES DEL SERVIDOR AL CUAL ESTAMOS CONECTADOS.

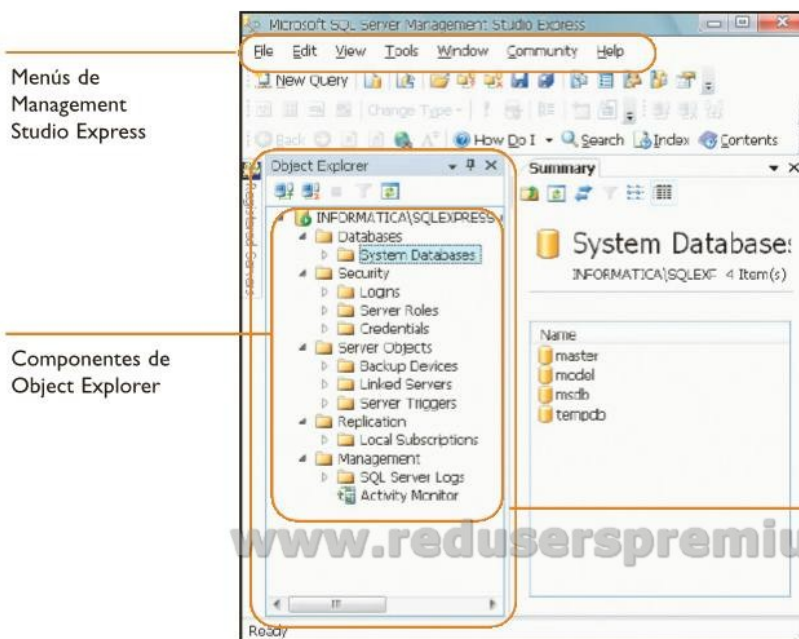
**Object Explorer**, uno de los más importantes de la aplicación, dado que se utiliza para navegar por los distintos objetos y componentes del servidor al cual estamos conectados. Cada panel de que disponemos en Management Studio puede ubicarse en el lugar de la pantalla que más cómodo nos resulte, simplemente, arrastrándolo desde la pestaña y soltándolo donde queramos.

## Object Explorer

A continuación, haremos un pequeño repaso de los componentes de Object Explorer:

- **Databases:** Administración de las bases de datos en el servidor, como: Crear, Eliminar, Cambiar nombre, Generar Scripts, Backup, Restore, entre otras funciones.

### ▶ Guía Visual 001 | Secciones del entorno de desarrollo



Componentes y menús de SQL Management Studio Express.



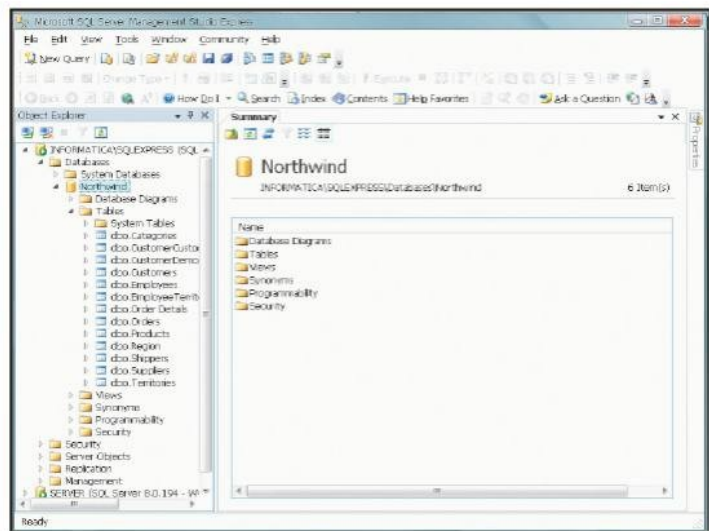


- **Security:** Administrar el inicio de sesión y los roles de usuarios en el servidor: Crear, Eliminar, Modificar y Generar Scripts, entre otras.
- **Server Objects:** Administrar las opciones de los objetos del servidor: Backup Device, Linked Server (servidores vinculados) y los DDL triggers de servidor (nuevo en SQL 2005).
- **Replication:** Administrar todo lo relacionado con la replicación entre servidores: Publicadores, Suscriptores, etc.
- **Management:** Monitoreo de la actividad del servidor, como así también la posibilidad de ver el Event Log (registro de eventos ocurridos en el sistema).

Para agregar una nueva tabla en la base de datos activa, simplemente debemos crearla desde el menú contextual del mouse, haciendo clic sobre la base en cuestión y seleccionando New Table.

### Características y propiedades

Veamos a continuación algunas propiedades y características de una base de datos desde la herramienta SQL Management Studio Express. Northwind, para quienes no la conocen, es una base de datos de ejemplo desarrollada por Microsoft. Si tenemos MS Access en nuestra computadora, veremos que en los ejemplos hay una base de datos con este mismo nombre. En las versiones de SQL Server,



## ▶ Guía Visual 002 | Tablas de Northwind

Base de datos seleccionada

Objeto de base de datos

Tablas

Vista de los registros pertenecientes a la tabla Customers

CustomerID	CompanyName	ContactName	ContactTitle	Address
ANATK	Affairs Publications	Maria Anders	Sales Representative	Olevo St. 57
ANATK	www.aflorhobby.com	Jana Bielekova	Owner	Jurata, Bielekova
ANATK	Alfreds Maternitet	Alfreda Jensen	Owner	Norlandsvej 2312
ANATK	Alfreds Maternitet	Thomas Hardy	Sales Representative	123 Maternitet St.
ANATK	Berglunds snabbköp	Christina Berglund	Order Administrator	Bergslagsvägen 8
ANATK	Blaauwke Gedej.	Pieter Bruijn	Sales Representative	Forstenweg 57
ANATK	Blondell perf.	Patrick Obrien	Marketing Manager	24, place King
ANATK	Bolsa Comercial	Patricio Simoes	Owner	27 Avenida, 67
ANATK	Bon app	Laura Lee	Owner	42, rue des Biscuits
ANATK	Bon app	Elizabeth Lincoln	Accounting Manager	23 Trantervej 8...
ANATK	Bon app	Hongki Anand	Sales Representative	Paradise Drive
ANATK	Chop-Ahu Chan	Yang Wang	Owner	1000001, 29

desde las primeras en adelante, también aparece esta base de datos con tablas y registros cargados, a modo de ejemplo. Podemos observar el contenido de esta base de datos desplegando el árbol desde la flecha ubicada a la izquierda del icono. Allí encontraremos **Database Diagrams**, **Tables**, **Views**, **Synonyms**, **Programmability** y **Security**, todos ellos objetos de la base de datos. Al desplegar **Tables**, veremos que se listan las tablas que la componen. Para acceder al contenido de una tabla –por ejemplo, de **Customers**–, basta con hacer clic derecho sobre ella y, del menú contextual que se despliega, seleccionar la opción **View Table**. Si queremos agregar una nueva tabla a la base **Northwind**, sólo debemos crearla desde el menú contextual del mouse, haciendo clic sobre la base de datos y seleccionando **New Table**. Para ver la definición de una tabla

existente –ya sea Nombre de los campos, Tipos de datos, Longitud del campo, etc.–, abrimos el menú contextual sobre la tabla en cuestión y elegimos **Design**.

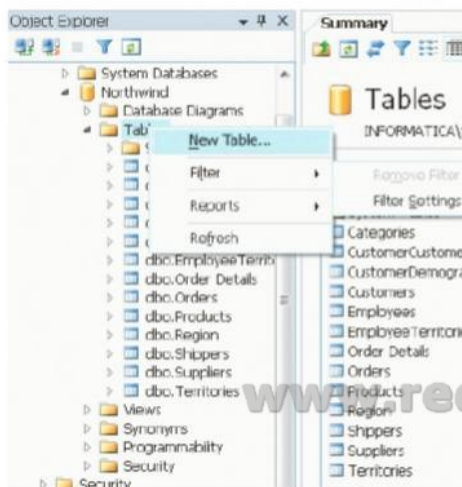
Esto mismo puede aplicarse a los otros objetos que componen la base de datos, como las vistas, de las cuales podremos ver su código, crear nuevas, borrarlas, etc.

### Conexión a otros servidores

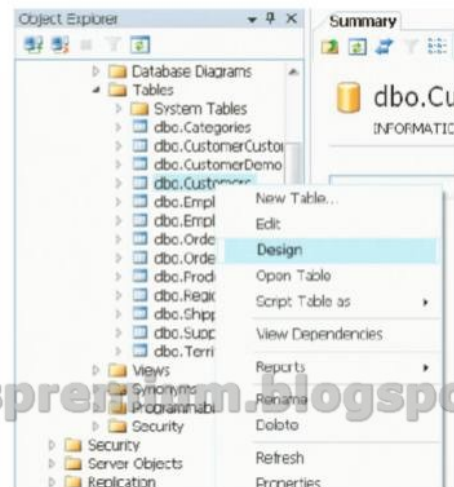
Object Explorer nos permite conectarnos a más de un servidor al mismo tiempo, pero con distinto login (nombre de usuario y contraseña). Esta opción es muy práctica cuando debemos trabajar con bases de datos ubicadas en otra computadora física de la red a la cual pertenecemos, o a nuestra misma PC pero con otra versión de SQL Server, por ejemplo, la 2000. Entonces, nos dirigimos al menú **File/Connect Object Ex-**

## ▶ Guía Visual 003 | Menús contextuales

Menú contextual para crear una nueva tabla



Menú contextual para vista de diseño de la tabla Customers



Manejo de menús contextuales sobre tablas y vistas.

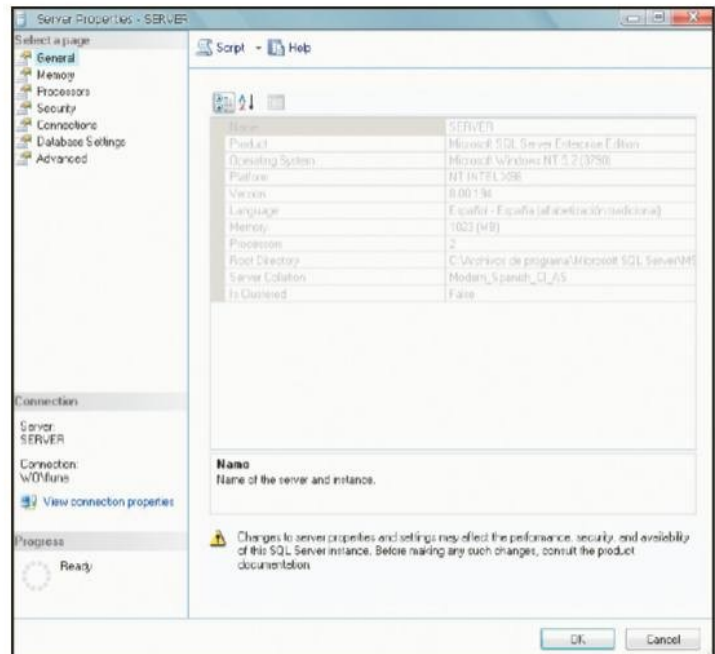




plorar. Se nos solicitarán los datos para conectarnos (Tipo de servidor, Nombre o dirección IP, Tipo de autenticación), luego de lo cual, en el panel **Object Explorer**, veremos nuestra conexión a un nuevo servidor.

En el ejemplo siguiente, vemos en sólo tres pasos, cómo conectarnos a un servidor SQL Server 2000, ubicado en un servidor de Red Windows 2003. Si luego deseamos desconectarnos de algún servidor, simplemente, hacemos clic derecho del mouse sobre la conexión que corresponda y, del menú contextual, elegimos **Disconnect**. Para acceder a las propiedades de un servidor SQL, nos ubicamos sobre el nivel superior del árbol, donde está el nombre del servidor; pulsamos el botón derecho del mouse y seleccionamos **Properties**.

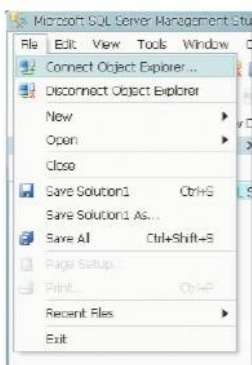
Éstas son algunas de las funcionalidades más importantes de Object Explorer. Como dijimos



antes, será la característica que más utilizaremos de la herramienta SQL Management Console Express.

## ▶ Guía Visual 004 | Conexión al servidor de base de datos

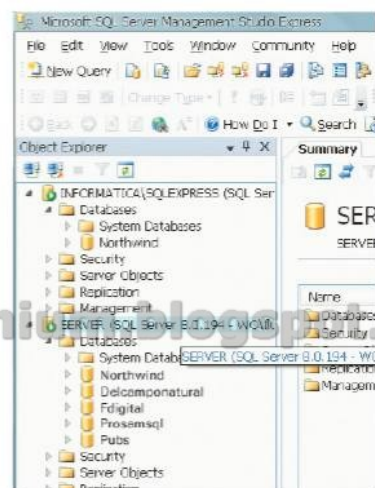
1. Seleccionamos del menú **File**, la opción **Connect Object Explorer...**



2. Ingresamos **tipo de servidor**, nombre o **dirección IP**, tipo de autenticación, y en el caso de seleccionar **SQL Server Authentication** ingresamos el **usuario** y **contraseña** correspondientes.



3. Una vez realizada la conexión, se listará en Object Explorer el nuevo Servidor de base de datos.



EL PANEL REGISTERED SERVER MUESTRA LOS SERVIDORES REGISTRADOS QUE ESTÁN DISPONIBLES. PODEMOS TENER MÁS DE UN SERVIDOR Y, ASÍ, CENTRALIZAR LA ADMINISTRACIÓN DE TODOS.

## Registered Server

El panel **Registered Server** muestra los servidores registrados que están disponibles. En él podemos tener más de un servidor y, así, centralizar la administración de todos. También es posible registrar un nuevo servidor o armar grupos en los que ubicaremos a los que están registrados. Supongamos que tenemos diez servidores, de los cuales cinco son de producción y el resto, de desarrollo. Una alternativa es registrar los diez juntos, lo cual nos traerá una complicación al momento de identificar cuáles son de cada tipo. La manera más prolija de realizar esta operación es, en principio, crear un grupo de servidores y, luego, registrarlos dentro de él. Para crear un grupo de servidores, abri-

mos el menú contextual sobre **Database Engine**, y elegimos **New/Server Group**. Aparecerá una ventana donde se nos pedirá que ingresemos los siguientes datos:

- **Group name:** Nombre del grupo de servidores que vamos a crear.
- **Group Description:** Breve descripción de cada uno.

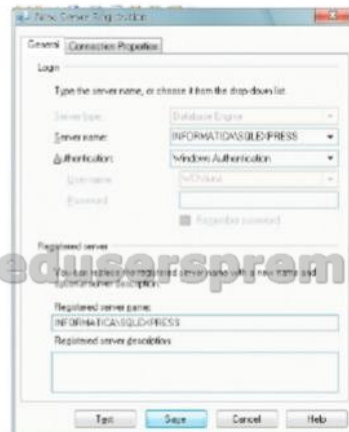
Presionando el botón **Save**, se guardará el nuevo grupo de servidores. Para registrar un nuevo servidor dentro del grupo, hacemos clic derecho sobre él y seleccionamos el menú **New/Server Registration**. Al realizar esta operación, aparecerá una ventana donde se nos solicitarán los datos del servidor.

## ▶ Guía Visual 005 | Registrar un nuevo servidor

Creación de un nuevo grupo de servidores



Registación de un nuevo servidor



Podemos registrar servidores no sólo SQL Server 2005, sino también SQL Server 2000.

www.reduserspremium.blogspot.com.ar





Podemos registrar servidores no sólo SQL Server 2005 sino también 2000; lo mismo es válido si la edición del servidor SQL Server es 2000 Standard o 2000 Enterprise y no deseamos usar el Query Analyzer o el Enterprise Manager, herramientas de esta versión.

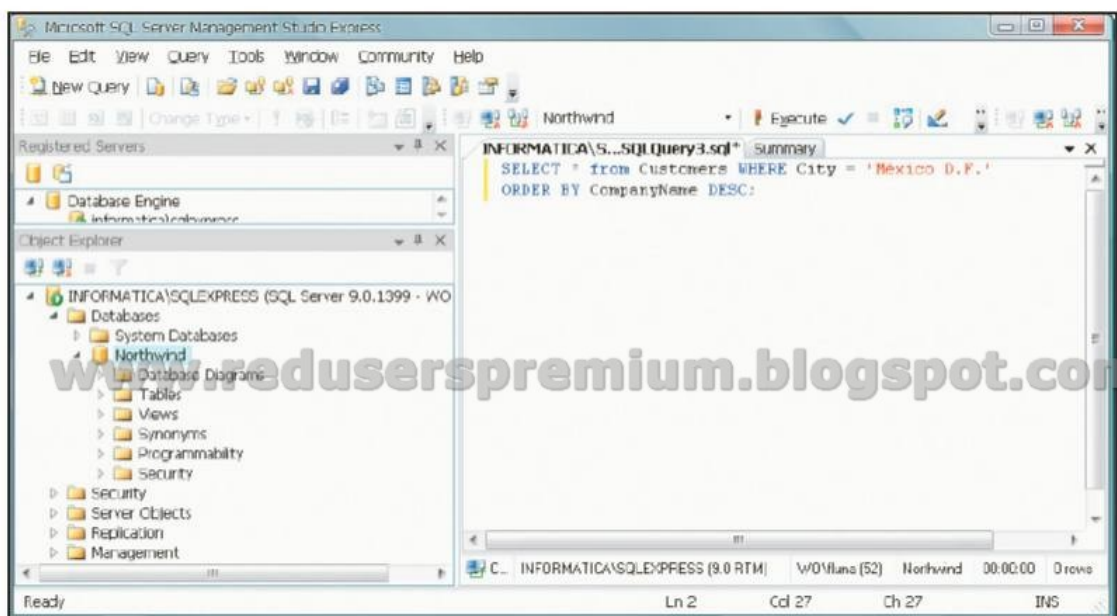
## Trabajar con queries y scripting

En versiones anteriores a SQL Server 2005, para crear consultas, se trabajaba con una herramienta conocida como **Query Analyzer** (analizador de consultas). Tanto ésta como **Enterprise Manager**, que antes eran herramientas independientes, ahora están integradas dentro de SQL Server Management Studio. Para crear una nueva query o consulta, dentro de Object Explorer, nos posicionamos sobre la conexión de base de datos en la cual trabajaremos, presionamos el botón **New Query**, ubicado en la barra de herramientas, o bien usamos la combinación de teclas

Tanto Query Analyzer como Enterprise Manager, que antes eran herramientas independientes, ahora están integradas en SQL Server Management Studio.

<Control+N>. En el panel derecho de Object Explorer se habilitará una ventana para escribir la consulta. Si pulsamos otra vez New Query, se abrirá una nueva solapa donde podemos escribir otra consulta; así podremos abrir cuantas solapas necesitemos al mismo tiempo, cada una de las cuales será un archivo plano con extensión .SQL distinto, que podremos abrir desde cualquier editor de textos, como el Bloc de notas. Para guardar la consulta, presionamos el botón Guardar o vamos a File/Save. Como mencionamos en la introducción a base de datos y SQL Server, el lenguaje

**FIGURA 010** | Ventana de Query con una consulta escrita en Transact-SQL.



EL RESULTADO DE LA EJECUCIÓN DE ESTA CONSULTA ESTÁ EN FORMATO DE GRILLA, PERO TAMBIÉN PODEMOS HACER QUE SE MUESTRE EN PANTALLA EN MODO TEXTO O ENVIARLO DIRECTAMENTE A UN ARCHIVO.

propio de éste es Transact SQL, por lo cual todo lo que escribamos dentro de las ventanas de queries debe respetarlo.

### Escribir y mostrar una query

Cuando nos disponemos a escribir una nueva consulta, se habilita una barra de herramientas con las funciones más comunes. Una de ellas es un combo desplegable destinado a seleccionar sobre qué base de datos vamos a trabajar. Una vez que seleccionamos la base de datos, por ejemplo, Northwind, escribimos la consulta que deseamos en la solapa New Query:

```
"SELECT * FROM EMPLOYEES;
```

Luego presionamos el botón **Execute**, de la barra de herramientas. Al hacerlo, obtenemos en pantalla una grilla, con el resultado correspondiente. En la parte inferior de la ventana, una barra de tareas nos da información relevante sobre la consulta realizada, como la base de datos sobre la que estamos trabajando, el usuario con quien estamos conectados a la base, el nombre de ésta, el tiempo que demoró la consulta en ser procesada y el total de registros que incluye.

### Otras opciones de manejo de queries

El resultado de la consulta realizada se obtuvo en formato grilla, pero también

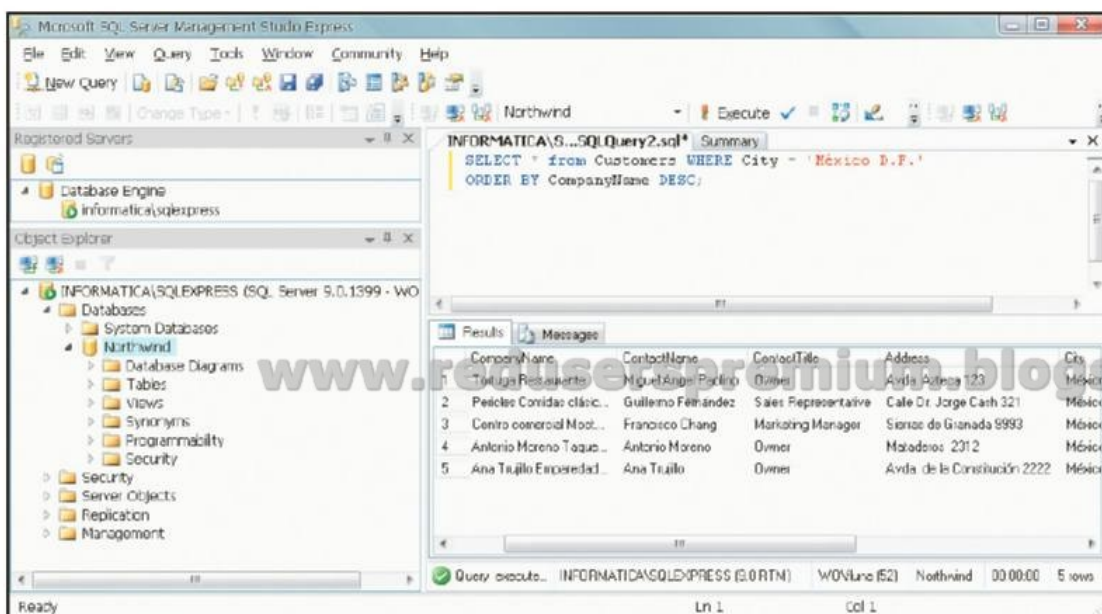


FIGURA 011 | Aquí vemos el resultado de nuestra consulta en formato de grilla.





podemos hacer que se muestre en pantalla en modo texto o bien enviarlo directamente a un archivo. En este caso, nos dirigimos al menú **Query**, y en la opción **Result To**, seleccionamos el modo que deseamos.

Desde este mismo menú, tenemos acceso a otras opciones, como: Ver el plan de ejecución, Revisar la sintaxis de las sentencias antes de ejecutar, Query Option, y otras.

Desde el submenú **Query Option** tenemos la posibilidad de configurar varios parámetros relacionados con la ejecución y la conexión. Otra de las opciones es **Template Explorer**, que permite utilizar una gran cantidad de plantillas para hacer la mayoría de las operaciones en T-SQL de una forma más eficiente.

Una de las novedades de esta herramienta es que permite abrir un archivo con extensión .SQL sin estar conectados a un servidor. Supongamos que tenemos los archivos y los hemos trasladado a una computadora portátil que estará desconectada de un servidor o que no tiene SQL Management Studio instalado. Pues bien, para abrir un archivo, no necesitaremos estar conectados a la base de datos, y de todos

modos podremos editarlo y trabajar con él aprovechando las funcionalidades de Management Studio, en vez de tener que usar Word-Pad o el Bloc de notas.

## Reportes

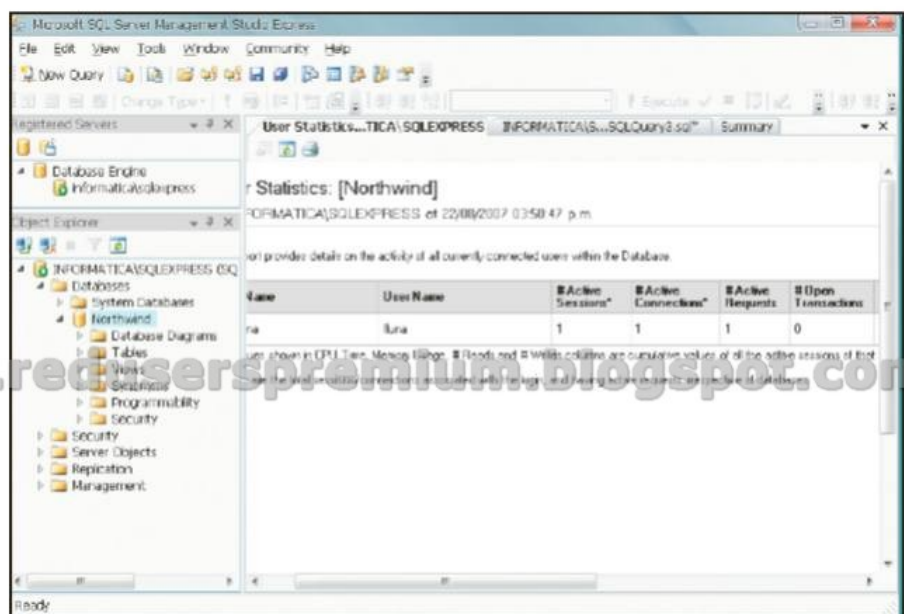
Otra de las novedades que incorpora Management Studio es la inclusión de una serie de reportes predefinidos para diferentes tareas de monitoreo y seguimiento.

Podemos acceder a estos reportes desde Object Explorer, pulsando el botón derecho del mouse sobre alguno de sus ítem (Database, Security, etc.) y, luego, del menú desplegable, seleccionando **Reports/Standard Reports**.

Por ejemplo, si nos posicionamos sobre la conexión, veremos un listado bastante importante de reportes. Entre ellos encontramos:

**Server Dashboard**, **Memory Consumption**, **Schemas** y **Change History**. Si realizamos el paso anterior y elegimos **Security/Logins**, se presenta otro tipo de reportes. Por ejemplo,

**FIGURA 012** | Aquí vemos las conexiones que se realizan con un mismo nombre de usuario a la base de datos.



LAS TAREAS DE SCRIPTING EN BASES DE DATOS NOS FACILITAN NUESTRA LABOR Y SON MUY IMPORTANTES PARA TODO DESARROLLADOR, PORQUE NOS PERMITEN AUTOMATIZAR PROCESOS LARGOS O TEDIOSOS, COMO LA CREACIÓN DE UNA BASE DE DATOS.

hay uno llamado **Login Failure**, que muestra un cuadro indicativo de la cantidad de intentos de inicio de sesión que no fueron válidos, por ejemplo, porque la contraseña ingresada no coincide.

La incorporación de todos estos reportes facilita las tareas de control y monitoreo que un DBA (administrador de base de datos) debe hacer sobre el servidor. Ésta no es la única opción, pero resulta sumamente práctica y útil en muchos casos.

### Facilitar tareas con scripting

Casi siempre, ya sea como desarrolladores independientes o trabajando para una empresa, las tareas de scripting en bases de datos nos facilitan nuestra labor y son muy importantes para todo desarrollador o DBA, porque nos permiten automatizar procesos largos o tediosos, como la creación de una base de datos con sus tablas, vistas y relaciones.

En SQL Server Management Studio disponemos de un botón **Script** en las ventanas principales del sistema, como la de creación de base de datos, tablas, vistas, etc.

Supongamos que vamos a generar una nueva base de datos y tablas para un sistema y, luego, tenemos que repetir este paso en el servidor SQL de nuestro cliente. Al momento de crear la base, presionamos el botón **Script** de dicha ventana, y todo el proceso que realicemos, se grabará en el lenguaje Transact-SQL, de modo que podremos guardarlo en un archivo con extensión **.SQL**.

Este archivo de script puede ejecutarse en cualquier otro SQL Server Management Studio, y así generaremos la base de datos y las tablas en forma automática.

Ya conocimos algunas de las principales herramientas con las que trabajaremos en el diseño y administración de bases de datos. A continuación, realizaremos un recorrido por los principales conceptos teóricos, lo que nos permitirá desarrollar bases confiables.

### Tabla 3 | Opciones que brinda el botón Script

Script Action to new Query Windows	Indicamos que el script se debe generar en una nueva ventana.
Script Action to File	Permite indicar que el script se genere automáticamente en un archivo con extensión <b>.SQL</b> .
Script Action to Clipboard	Permite indicar que el script se copie al Portapapeles de Windows, para así utilizarlo en el lugar en que lo necesitamos.





# Diseño de bases de datos

Llegó el momento de aprender a diseñar una base de datos, respetando la integridad relacional.

Los sistemas de bases de datos, en general, tienen un conocimiento muy limitado de lo que significan los datos que almacenan. Normalmente, sólo comprenden ciertos tipos de valores atómicos y, quizás, algunas restricciones de integridad simples que pueden aplicarse a ellos. Cualquier interpretación más allá de este nivel queda reservada a los humanos.

Debido a estas limitaciones, surgen los modelos entidad-relación, que son útiles para comprender y diseñar las bases de datos.

Los conceptos que aquí explicaremos tienen importantes fundamentos teóricos, en lo que corresponde tanto al modelo entidad-relación, como a la normalización y la teoría que la sustenta. Aquí presentamos, de una manera algo informal, algunas de estas ideas y ciertos ejemplos destinados a clarificarlas.

## Diagramas entidad-relación, y el diseño lógico y físico

Todo desarrollador de bases de datos tiene como tarea principal transformar y modelar los datos del mundo real para representarlos mediante objetos del gestor correspondiente. En las primeras etapas del relevamiento de los requerimientos y el diseño de los sistemas, solemos utilizar modelos que representan la información que debe almacenarse en la base de datos. Estos modelos tienen como principal objetivo representar semánticamente, mediante algún tipo de clasificación y relación, el universo de los datos importantes del sistema.

Una de las técnicas de representación de datos más usada, y que ha tenido mayor éxito, son los diagramas del modelo entidad-relación. Éstos representan, mediante símbolos

gráficos, el significado de los datos para crear un modelo conceptual.

La mayoría de los desarrolladores utilizan los diagramas entidad-relación para comprender de forma visual la base de datos. El primer paso es entender las entidades de un modelo puramente lógico, que, luego, se transformarán en tablas del modelo físico. Por lo tanto, el proceso de diseño de bases de datos puede dividirse, desde el punto de vista teórico, en dos fases: una en la que se diseña el modelo entidad-relación lógico, y otra en la que se lo transforma en un modelo puramente físico, constituido por tablas y relaciones de una base de datos. En este modelo físico, no siempre existe una correspondencia directa uno a uno entre las entidades lógicas de los diagramas del modelo entidad-relación y las tablas físicas de la base de datos. Sin embargo, puede decirse que, por lo general, existe una gran correlación entre ellas.

## Entidad

Cuando hablamos de entidades, lo primero que identificamos son los objetos del mundo real; por lo tanto, una definición informal

Todo desarrollador de bases de datos tiene como tarea principal transformar y modelar los datos del mundo real para representarlos mediante objetos del gestor correspondiente.

expresaría que las entidades son los objetos distinguibles del modelo que queremos representar. Por este motivo, suele decirse que las entidades son los sujetos o sustantivos de nuestro modelo. Una de las cosas que podemos expresar es que el mundo está hecho de entidades. Entonces, lo que podemos hacer es agrupar estas entidades reales y verlas como tipos de entidades. Podemos hacer un paralelo diciendo que las clases equivalen a los tipos de entidades, y que las instancias de estos tipos de entidades son los equivalentes a los objetos propiamente dichos. Por ejemplo, si regresamos a lo visto en los capítulos anteriores acerca de las clases, diremos que un modelo de auto o un auto determinado con ciertas características es una instancia de

una entidad que llamaremos auto. Esto nos da la ventaja de que podemos asignar a estas entidades ciertas propiedades, que todas las instancias tienen en común. Algunos ejemplos clásicos de entidades podrían ser: Proveedores, Clientes, Productos, Contactos, Facturas, Libros, etc.

### Propiedad

Es una pieza de información que describe la entidad y corresponde, en líneas generales, a los atributos del modelo relacional. Podemos distinguir propiedades simples o propiedades compuestas, de otras propiedades simples; propiedades clave, que son únicas dentro de un contexto; propiedades base y propiedades derivadas, como la suma de alguna otra propiedad.

## Modelo entidad-relación

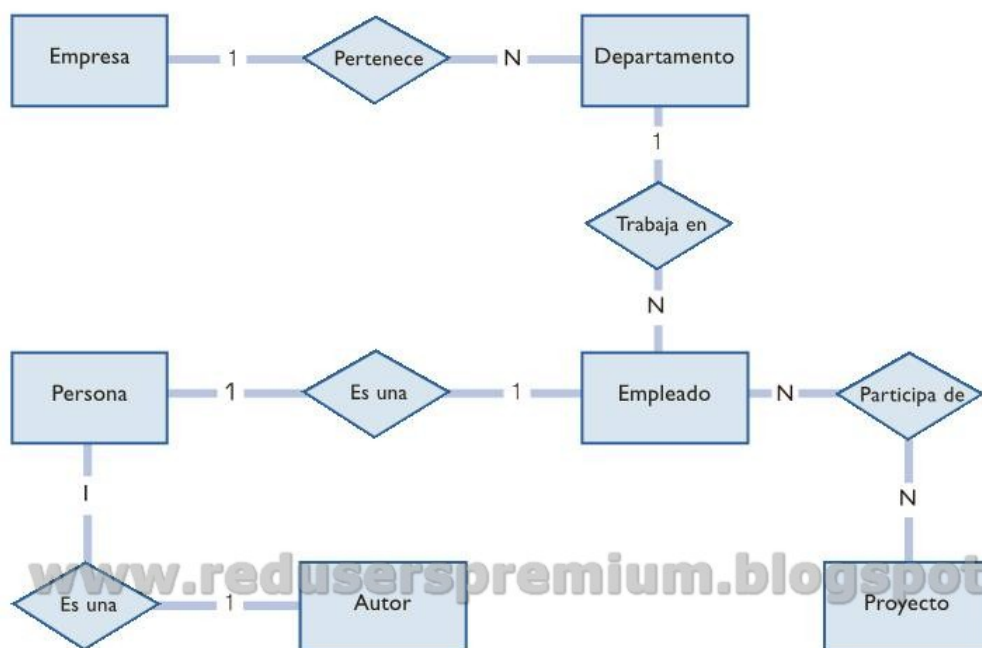


FIGURA 013 | Modelo entidad-relación clásico, en el que pueden verse entidades, relaciones y su cardinalidad.





PODEMOS HACER UN PARALELO DICHIENDO QUE LAS CLASES EQUIVALEN A LOS TIPOS DE ENTIDADES, Y QUE LAS INSTANCIAS DE ESTOS TIPOS DE ENTIDADES SON LOS EQUIVALENTES A LOS OBJETOS PROPIAMENTE DICHOS.

### Relacionamiento

Es una especie de entidad que permite interconectar otras dos (o más) entidades. Por este motivo, suele considerarse a las relaciones como los verbos que vinculan las entidades del modelo. Por lo tanto, existen relaciones uno a uno, muchos a uno y muchos a muchos. Estas relaciones se representan de distintas formas en un modelo relacional y se implementan físicamente de diversas maneras. Suelen modelarse de distintos modos, pero siempre se las asocia con una acción o verbo. En general, se modelan relaciones binarias entre dos entidades. Por ejemplo, si tenemos las entidades **persona** y **documento**, podríamos decir que un documento “pertenece” a una y sólo una persona, y que una persona “posee” de 1 a N documentos.

### Subtipo

Se dice que una entidad B es un subtipo de la entidad tipo A si y sólo si cada B es necesariamente un A. Esto permite aplicar un concepto de herencia de atributos y relaciones. Por ejemplo, un programador es una persona, y un empleado administrativo es una persona. Existen características comunes entre el programador y el administrativo, que coinciden con los atributos de la entidad persona. Sin embargo, cada uno de ellos tiene propiedades específicas no comunes, por lo que realizan una especialización de la entidad persona mediante un subtipo

que hereda las características de esa entidad. Este concepto no fue incluido inicialmente en el modelo de Chen, pero se lo agregó más tarde, y para quienes conocen la teoría de objetos, es similar a la herencia de clases.

## De la teoría a la práctica

De la primera idea del modelo entidad-relación, en la que el objetivo principal era describir las bases de datos, llegamos a un proceso de modelado y diseño en varias etapas, en general ayudado por herramientas del tipo CASE (*Computer Aided Software Engineering*, que significa desarrollo de software asistido por computadoras). Este modelado ocurre, normalmente, en dos o en tres etapas.

En la primera se realiza un modelo conceptual del diseño de la base de datos mediante el modelo entidad-relación, que representan en forma gráfica la base de datos desde el punto de vista de los conceptos. En este estado no existe un diseño de tablas, sino de los datos que serán almacenados en las entidades. Esto quiere decir que el modelo es tecnológicamente independiente y representa los datos de manera exclusiva.

El segundo paso del diseño corresponde al diseño lógico, y es en este momento cuando se mapean las entidades del modelo conceptual a tablas lógicas, verificándose que se cumplan

**Relacionamiento es una especie de entidad que permite interconectar otras dos (o más) entidades.**

ciertas características, como la normalización. Este modelo lógico también es tecnológicamente independiente, ya que se modela en términos de tipos de datos genéricos. Por último, la tercera etapa corresponde al diseño físico, momento en el cual se adapta el modelo lógico a la plataforma de destino específica, y se seleccionan las diferentes opciones de implementación, como los ti-

pos de datos específicos, y el hecho de que los nombres de las tablas, objetos y columnas, entre otros, sean nombres válidos. En esta última etapa suelen realizarse tareas para identificar las columnas que son candidatas para crear índices, las que son las claves primarias o las claves secundarias, y otros componentes requeridos para que la implementación física de la base de datos funcione de manera eficiente.

Un diagrama entidad-relación conceptual suele contener sólo los nombres de las entidades y las relaciones entre ellas. Habitualmente, no resuelve la manera de implementación de las entidades que se relacionan de la forma muchos a muchos, y tampoco es común incorporar los atributos. Lo más

## Subtipos

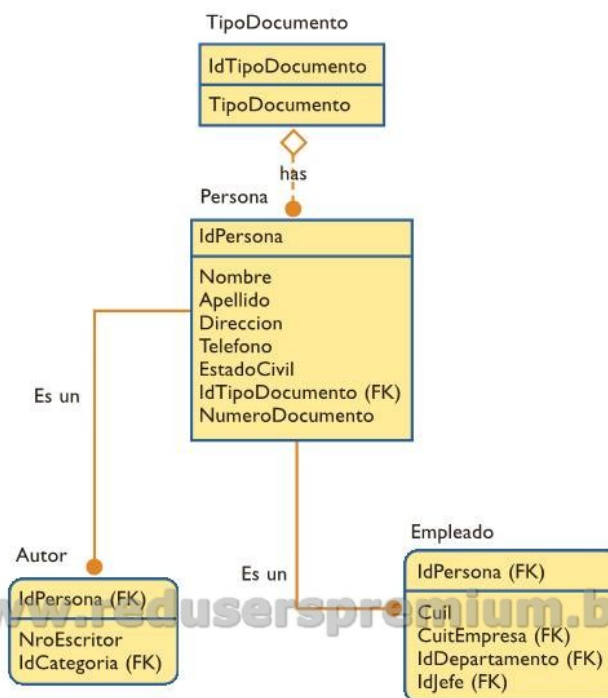


FIGURA 014 | Ejemplo de herencia de atributos mediante subtipos.





importante es que el modelo nos da una idea de cómo será el diseño final de la base de datos. Dado que existe un cierto tipo de correspondencia entre entidades y tablas, muchos desarrolladores que usan herramientas CASE suelen saltar la primera etapa conceptual, y comienzan directamente en el modelo lógico y físico. Luego, del modelo

conceptual se puede pasar al modelo lógico. Para hacerlo, sobre la base de los requerimientos y análisis realizados, se agregan todas las propiedades y atributos a las entidades. Estos atributos serán las columnas de nuestras tablas. Otra tarea que debemos hacer en este modelo es transformar las relaciones de muchos a muchos en entidades

## Modelo lógico

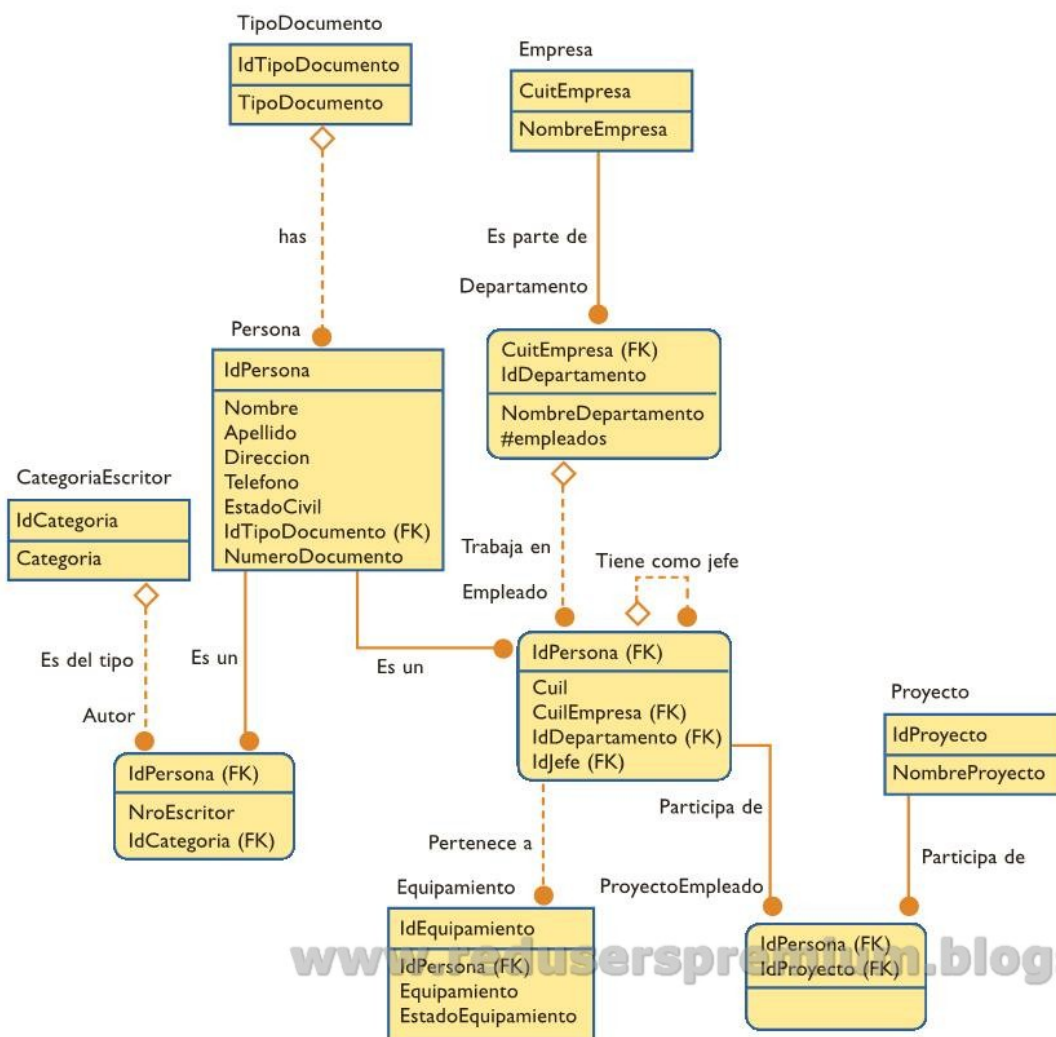


FIGURA 015 | Modelo lógico equivalente al modelo conceptual anterior.

asociativas. Éstas serán nuevas tablas que contienen registros con los atributos o columnas de las claves primarias de las tablas que, originalmente, se relacionaban de muchos a muchos, para identificar qué combinaciones existen en la relación. Otro paso que suele realizarse es la creación

de tablas de referencia, que se utilizan para limitar la cantidad de posibles valores que puede tener un atributo o columna, ya que mediante la relación de la clave foránea, se puede asegurar la existencia del dato en esta tabla de referencia, y se impide asignar valores inadecuados al atributo. La gran diferencia entre

## Modelo físico

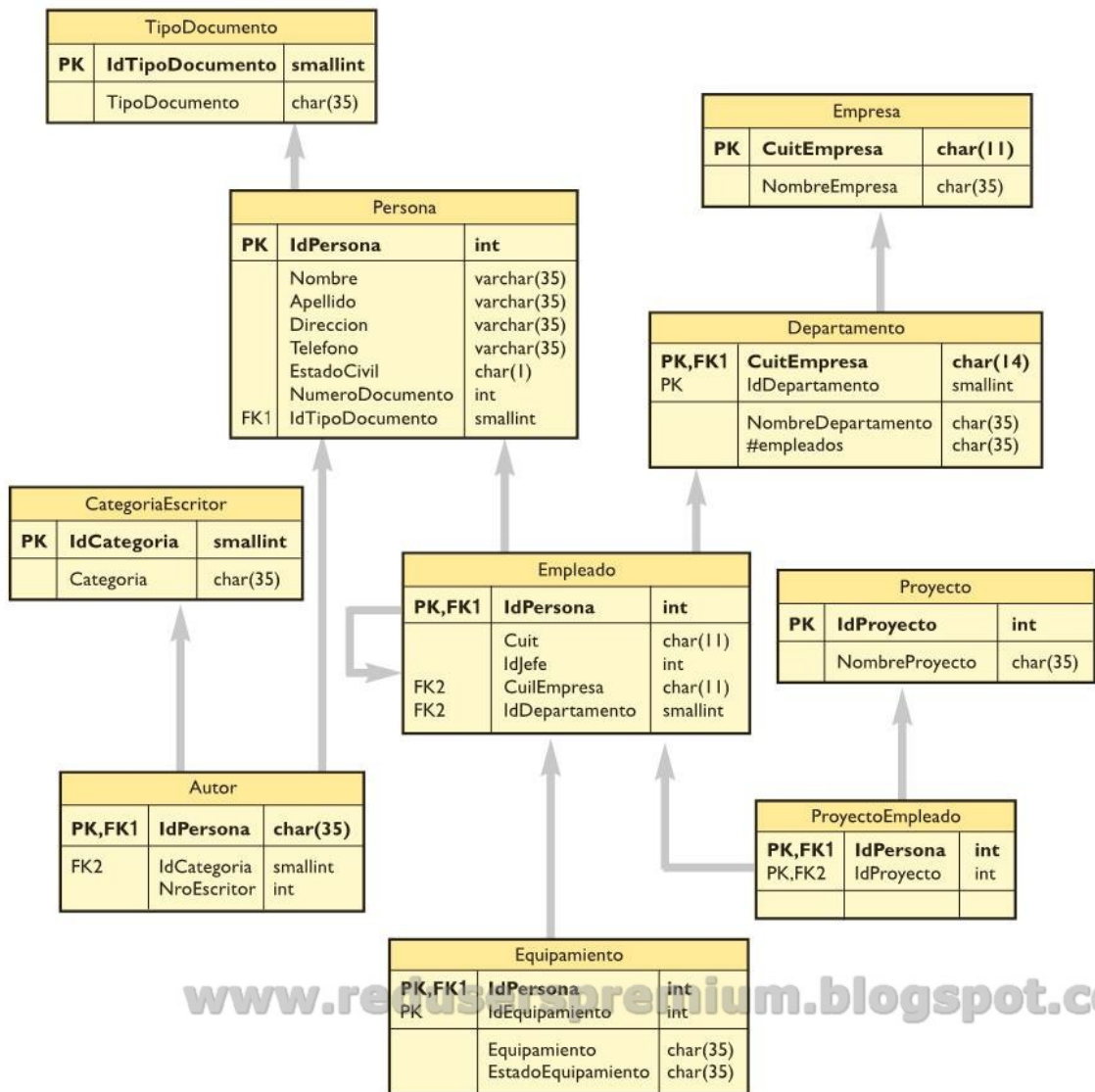


FIGURA 016 | Modelo físico equivalente obtenido mediante herramientas CASE.





usar una tabla de referencia y una restricción simple radica en que una restricción debe ser modificada ante una necesidad de nuevos valores por restringir, y debe aplicarse en la base de datos, mientras que una tabla de referencia sólo requiere una modificación de los datos de la tabla referenciada y no necesita cambios en el diseño de la base. Otra tarea que suele efectuarse en esta etapa es la creación de supertipos y subtipos. Cuando existen entidades muy similares, suele ser necesario crear una que absorba todos los atributos comunes. Luego, la especialización mediante subtipos se lleva a cabo generando las entidades correspondientes.

### Herramientas que simplifican la tarea

Si bien las herramientas de ayuda para la creación de software son de gran utilidad porque permiten trabajar de manera más ágil y eficiente con modelos de base de datos, debe-

De la primera idea del modelo entidad-relación, llegamos a un proceso de modelado y diseño en varias etapas, en general ayudado por herramientas CASE.

mos tener en claro que las mismas tareas pueden realizarse sin inconveniente, tanto manualmente como con herramientas que no llegan a considerarse CASE.

El modelo elegido en este caso no contiene ninguna particularidad destacable en cuanto a diseño por seguir. Fue creado, simplemente, como manera de mostrar algunas de las características y consecuencias del diseño en lo que se refiere al modelado relacional.

## Valores de un atributo

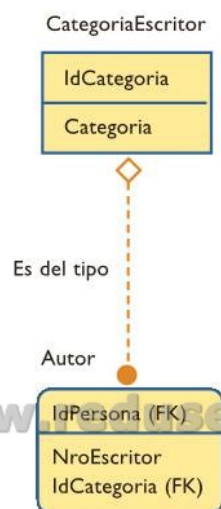


FIGURA 017 | Tabla de referencia para limitar posibles valores de un atributo.

Este modelo expresa que una persona es un autor y/o un empleado. Dependiendo de cómo se lo considere, tienen distintas propiedades. Por tal motivo, se ha seleccionado un supertipo persona con dos subtipos: empleado y autor. Cada empleado puede trabajar en uno y sólo un departamento. Cada departamento pertenece a una y sólo una empresa. No puede existir un departamento

aislado de una empresa, y de allí la relación de entidad débil y fuerte. A diferencia de la relación identificativa entre empresa (fuerte) y departamento (débil), en la relación entre empleado y departamento se ha priorizado la herencia de atributos para la entidad empleado, por lo que no se ha incorporado la clave primaria de la entidad departamento como parte de la clave primaria de la entidad

## SQL Server Management Studio

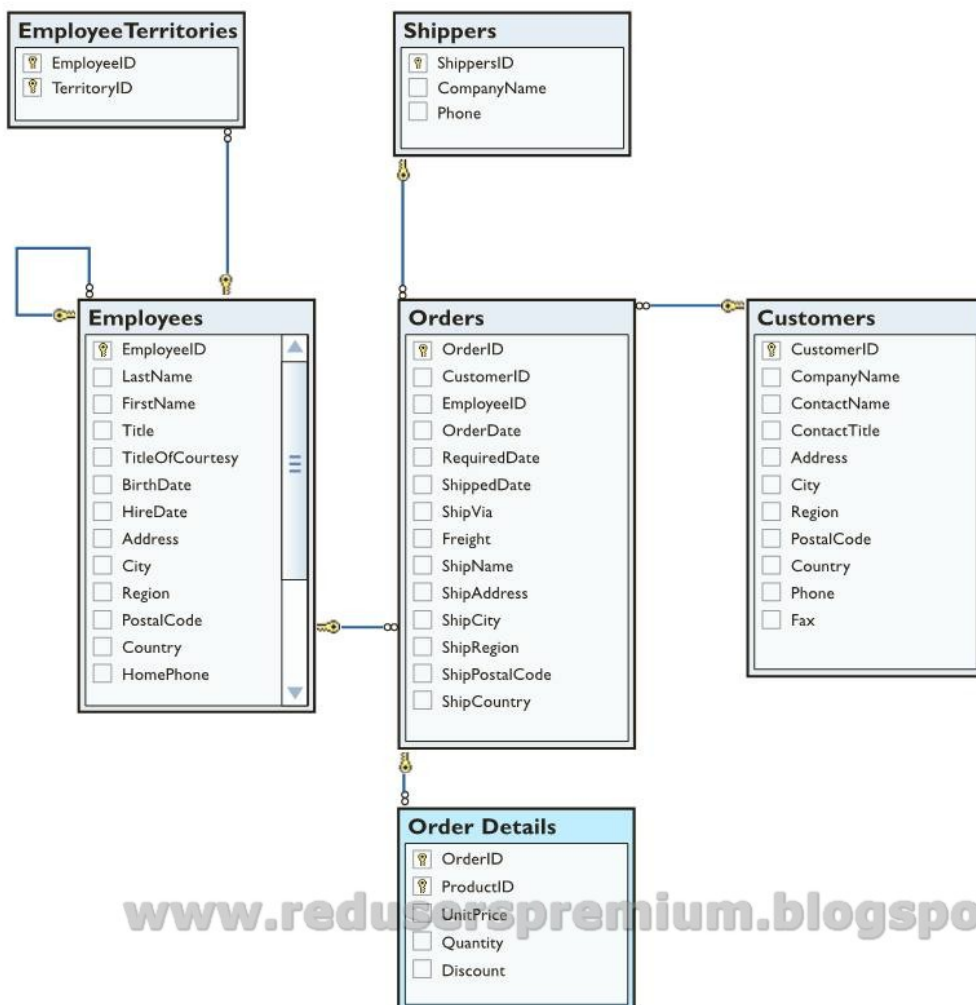


FIGURA 018 | Ejemplo de un diagrama de la base de datos Northwind hecho en SQL Server Management Studio.





empleado. En su lugar, para forzar por un lado y restringir los posibles valores por otro, se ha obligado a que el valor del atributo que indica el departamento del empleado no pueda ser nulo y, además, a que exista la referencia a la tabla Departamentos mediante una clave foránea.

Otra característica destacable del modelo es que un empleado puede participar en varios proyectos, así como en un proyecto hay varios empleados, lo que se consigue mediante la tabla asociativa. Una limitación del modelo de ejemplo es que un proyecto no está relacionado a un departamento o empresa, lo cual no suele ser un limitante en este caso ficticio. Podría ser importante destacar que si en un proyecto participan dos empresas, debe haber al menos un empleado de cada una que establezca la relación, para que aparezca la relación indirecta a través de departamento.

Si desarrollamos un modelo utilizando una herramienta CASE, el pasaje del modelo lógico al físico suele ser simple y directo, porque la herramienta toma decisiones que permiten respetar tanto las reglas físicas como las lógicas. La relación entre ambos modelos es mantenida y supervisada por la herramienta. Además, algunas herramientas no distinguen entre modelos lógicos y modelos físicos. Las de diseño permiten no sólo realizar el diseño preliminar, sino también dar soporte al ciclo de desarrollo completo. Entre algunas de las capacidades más comunes que tienen, podemos mencionar que permiten realizar cambios en el modelo y aplicarlos en la base existente, sin perder los datos actuales. Esto se consigue generando las sentencias adecuadas para resguardar los datos y aplicar los cambios, documentar el modelo de datos y mantenerlo actualizado, o actualizar el modelo y los diagramas importando los cambios efectuados directa-

**El modelo entidad-relación nos da una idea de cómo será el diseño final de la base de datos.**

mente en la base de datos (aplicando un proceso de ingeniería reversa), resolver conflictos y asegurarnos de que, desde el punto de vista del diseño, se cumplan ciertas reglas mínimas para garantizar que se está generando un modelo relacional coherente.

## Consejos para la creación de una base de datos

Para que el diseño de una base de datos sea correcto, debemos tener en cuenta ciertos aspectos que nos evitarán modificar la base luego de crearla:

- **Estudio:** Estudiar correctamente el sistema al cual se le aplicará el modelo de automatización y codificación de datos, realizando reuniones de estudio con los usuarios vinculados a él para consultar necesidades de operación, tipos de información para manipular, y más.
- **Análisis:** Determinar las claves o identificadores de los objetos de la base de datos.
- **Vinculación:** Establecer relaciones entre los componentes y los sectores del sistema.
- **Graficación:** Dibujar o graficar el modelo de datos obtenido sobre la base de la información recopilada.
- **Identificación:** Identificar los aspectos más importantes de la entidad del sistema, que sean de interés.
- **Depuración:** Eliminar las relaciones redundantes y las que puedan obtenerse combinando otras asociaciones.



UN EJEMPLO DE HERRAMIENTA DE DISEÑO SIMPLE SON LOS DIAGRAMAS DE BASES DE DATOS DE SQL SERVER, QUE PERMITEN CREAR MODELOS VISUALES DE LA IMPLEMENTACIÓN FÍSICA DE LA BASE.

**Proceso de diseño**

El proceso que vamos a describir a continuación de manera informal se realiza en varios ciclos durante las etapas de análisis y de diseño iterativas. Esto significa que, si bien parece lineal, se puede avanzar mediante varias etapas de refinamiento hasta llegar al modelo definitivo.

Para crear un diagrama entidad-relación, lo primero que debemos identificar son las entidades más importantes. Para cada objeto importante de nuestro universo de datos, debemos asignar una entidad. Normalmente, asignamos un rectángulo a cada entidad o una tabla si trabajamos directamente con el modelo lógico.

El segundo paso del diagrama corresponde a la identificación de las relaciones entre las entidades. En este punto es necesario distinguir entre los distintos tipos de relaciones que pueden producirse entre las entidades, que deben ser expresados dentro del diagrama. Entonces, podemos agregar a nuestro

diagrama las relaciones de uno a uno, uno a muchos y muchos a muchos que tienen las entidades ya identificadas del modelo.

Luego, mapeamos las entidades del modelo con tablas de la base de datos, y asignamos los atributos de las entidades generando las columnas que correspondan. Para hacerlo, a cada entidad fuerte se le asigna, en general, una tabla del modelo de base de datos. Si se usa una herramienta CASE, no conviene agregar las columnas que serán claves foráneas, dado que la herramienta nos ayuda copiando la definición de la clave primaria de la tabla foránea al establecer la relación. Si estamos usando una herramienta CASE, en esta etapa es habitual comenzar con la asignación de tipos de datos y de propiedades, como la posibilidad de que la columna acepte un valor **null**. En este paso es importante asignar las claves primarias de cada tabla, porque ayudarán a comprender el modelo y crear atributos debido a las relaciones.

**Tabla 4 | Herramientas de modelado más utilizadas**

Diagramas de SQL Server	Permiten mantener documentada la base de datos y generar diagramas del diseño físico. Están incorporados en SQL Server desde sus últimas versiones.
Microsoft Visio	Permite visualizar los modelos y trabajar el ciclo completo del desarrollo, sincronizando cambios desde o hacia la base de datos.
Erwin	Una herramienta clásica con soporte al ciclo completo de desarrollo permite una amplia gama de modelado.





Para las entidades débiles, se asigna una tabla en la que parte de la clave primaria será, a su vez, una clave foránea, mediante una restricción que hace referencia a la tabla que representa a la entidad fuerte. Este tipo de relación se llama **identificativa**, porque obliga a la existencia del registro de la tabla que representa a la entidad fuerte. Esto es así porque no se puede asignar un valor null a las columnas de la relación, dado que son parte de la clave primaria de la tabla representante de la entidad débil.

Para cada relación **uno a muchos**, en la tabla cuya cardinalidad de la relación es **muchos**, se agrega una columna que es clave foránea y que hace referencia a la clave primaria de la tabla cuya cardinalidad es **uno**. Nos referimos a este tipo de relaciones como **no identificativas**, ya que la columna que es clave foránea no forma parte de la clave primaria de la tabla; de allí que si la columna acepta valores nulos y se le asigna éste, pue-

Para crear un diagrama entidad-relación, lo primero que debemos hacer es identificar las entidades más importantes. Para cada objeto importante de nuestro universo de datos, tenemos que asignar una entidad.

de haber un registro independientemente de la existencia del registro de la tabla relacionada. Obviamente, si se le asigna un valor a la columna, se debe respetar la restricción.

En caso de que existan relaciones **muchos a muchos**, es necesario crear, además de las dos tablas que se relacionan, una tercera llamada **tabla asociativa**, que permita establecer la relación. Esta tabla tiene como clave

## Entidades fuerte y débil



FIGURA 019 | Ejemplo de relación identificativa entre una entidad fuerte y una débil.

## Relación de uno a muchos y de referencia recursiva

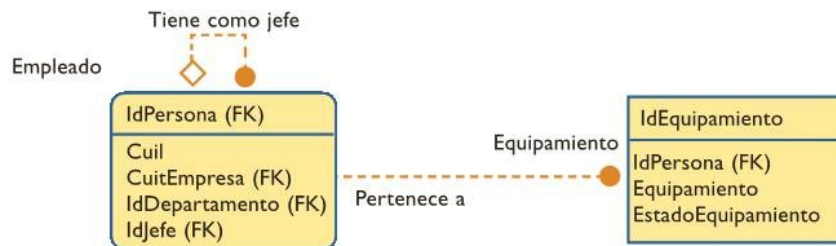


FIGURA 020 | Podemos ver la forma en que se representan las referencias recursivas y la relación uno a muchos.

primaria las columnas de las dos claves primarias de las tablas relacionadas, y las dos restricciones de claves foráneas con cada una de estas tablas base. Quizás el modelo entidad-relación sea la parte más tediosa, pero su conocimiento es fundamental para que cual-

quier desarrollo de sistemas sobre base de datos se haga de la manera más acertada posible. Así, se evitará la redundancia de información y se logrará una mejor performance en la aplicación que acceda a consultar o almacenar datos.

## Relación muchos a muchos

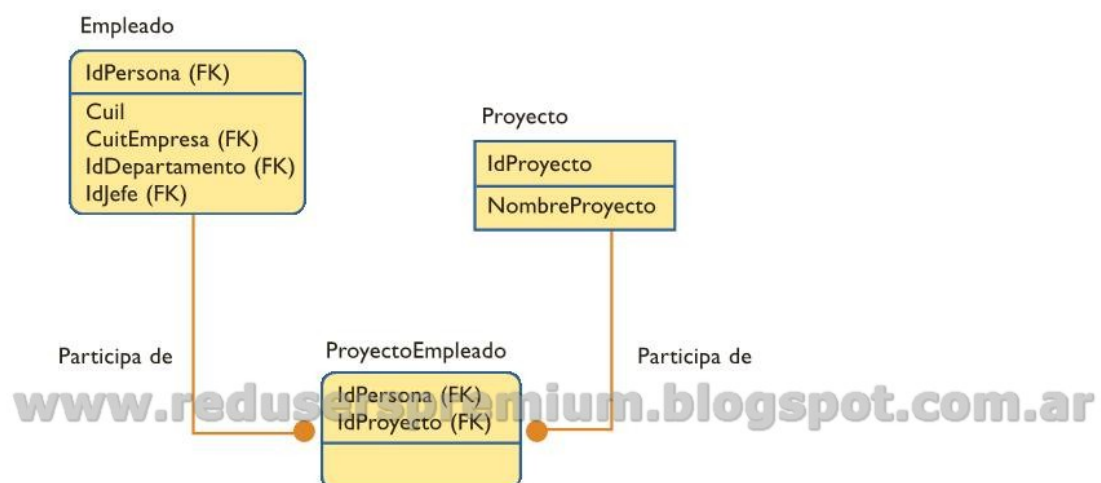


FIGURA 021 | Implementación de una relación muchos a muchos mediante una tabla asociativa.



**USERS**

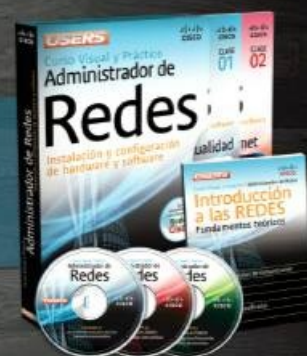


CURSOS.REUSERS.COM

# CURSOS INTENSIVOS



Los temas más importantes del universo de la tecnología desarrollados con la mayor profundidad y con un despliegue visual de alto impacto: Explicaciones teóricas, procedimientos paso a paso, videotutoriales, infografías y muchos recursos mas.



Brinda las habilidades necesarias para planificar, instalar y administrar redes de computadoras de forma profesional. Basada principalmente en tecnologías Cisco, es una obra actual, que busca cubrir la necesidad creciente de formar profesionales.

- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 3 CDs / 1 Libro



- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 4 CDs

Curso para dominar las principales herramientas del paquete Adobe CS3 y conocer los mejores secretos para diseñar de manera profesional. Ideal para quienes se desempeñan en diseño, publicidad, productos gráficos o sitios web.

Obra teórica y práctica que brinda las habilidades necesarias para convertirse en un profesional en composición, animación y VFX (efectos especiales).

- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 2 CDs / 1 DVD / 1 Libro



- ▶ 26 Fascículos
- ▶ 600 Páginas
- ▶ 2 DVDs / 2 Libros

Obra ideal para ingresar en el apasionante universo del diseño web y utilizar Internet para una profesión rentable. Elaborada por los máximos referentes en el área, con infografías y explicaciones muy didácticas.

[www.reduserspremium.blogspot.com.ar](http://www.reduserspremium.blogspot.com.ar)

Llegamos a todo el mundo con OCA \* y DHL \*\*

✉ [usershop@redusers.com](mailto:usershop@redusers.com) ☎ +54 (011) 4110-8700

[usershop.redusers.com.ar](http://usershop.redusers.com.ar)

\*\* Válido en todo el mundo excepto Argentina. \* Sólo válido para la República Argentina



Argentina \$8,90 (recargo al interior \$0,20) / México: \$45

**USERS**

**Microsoft®**

Curso teórico y práctico de programación

# Desarrollador .net

Con toda la potencia  
de **Visual Basic .NET** y **C#**

La mejor forma de aprender  
a programar desde cero



Basado en el programa  
Desarrollador Cinco Estrellas  
de Microsoft

**5**

## Bases de datos

Sistema relacional - Diagrama  
entidad/relación - Normalización

## Lenguaje SQL

Tablas - Agrupar y aplicar funciones  
Consultas de inserción - Actualización



ISBN 978-987-1347-43-8



9 789871 347438

