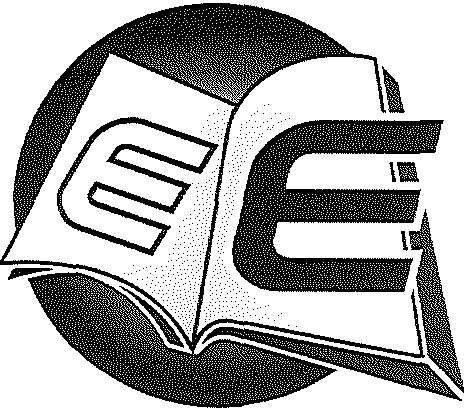
**Diccionario de datos en**

UD4

**sistemas ERP- CRM**



UF1888 Operaciones de Mantenimiento y Consulta de Datos

•

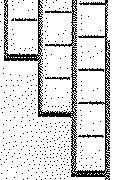
•

1. **Objetos del diccionario de datos**

El diccionario de datos es un repositorio de **metadatos** que contiene las definiciones de los oqjetos de datos, descripciones y relaciones entre sr. El diccionario es muy importante en el desarrollo de aplicaciones, ya que permite a los involucrados en el desarrollo conocer a detalle las bases de datos.

El diccionario de datos es, por tanto, una descripción detallada de todas las tablas que se encuentran dentro de la base de datos creada por el usuario/diseñador. Así, el diccionario contiene al menos todos los nombres de atributos y características para cada tabla del sistema .

•••

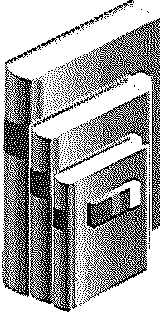


---,

**Tab**, **las**;-,-

--: **In**,**di**,**c**..**e**..**s**...,-,-

--**V**,**istas**,-,-



**Usuarios**

--

-.

' ' i i

**----- guemM,**

1 ' i i :

- **Procedimientos**

**-, ,-:-i ,-**

-i-1 -i-!-

---!-¡-;! -

1 ! i,-

-· : !¡-¡!

¡i -

;¡-

---,- ---1-:-:.;-

-1, j -'

.' -;-

'

•

••

El diccionario de datos define los datos haciendo lo siguiente:

* + Describe el significado de los flujos y almacenes que se muestran en el DFD.
  + Describe la composición de agregados de paquetes de datos que se mueven a lo largo de los flujos, es decir, paquetes complajos que pueden descomponerse en unidades más elementales.
  + Describe la composición de los paquetes de datos de los almacenes
  + Específica los valores y unidades relevantes de piezas elementales de información en los flujos de datos y los almacenes de datos

## •

El propósito de un diccionario de datos es asegurar que todos los miembros de los equipos de diseño e implementación de bases de datos usen la misma tabla, y los mismos nombres y características de los atributos. El diccionario de datos internamente guardados del DBMS contiene información adicional acerca de tipos de relación, verificaciones y aplicaciones de entidad e integridad referencial, así como tipos y componentes del índice. Esta información adicional es generada durante la etapa de implementación de la base de datos.

El diccionario de datos a veces se describe como "la base de datos del diseñador de bases de datos" porque registra las decisiones de diseño acerca de tablas y sus estructuras.

Al igual que el diccionario de datos, **el catálogo del sistema** contiene **metadatos.** El catálogo del sistema se puede definir como un diccionario de datos específico del sistema que describe todos los objetos dentro de la base de datos, incluidos datos acerca de nombres de tabla, el creador de la tabla y fecha de creación, el número de columnas de cada tabla, el tipo de datos correspondiente a cada columna, nombres de archivo de índice, creadores de índice, usuarios autorizados y privilegios de acceso. Como el catálogo del sistema contiene toda Ja información requerida del diccionario de datos, es frecuente que los términos catálogo del sistema y diccionario de datos se usen indistintamente De hecho, el software actual de bases de datos relacional por Jo general contiene sólo un catálogo del sistema, del cual se puede derivar información del diccionario de datos del diseñador. El catálogo del sistema es en realidad una base de datos creada por el sistema cuyas tablas guardan las características y contenido de Ja base de datos creada por d usuario/diseñador. Por tanto, las tablas del catálogo del sistema se pueden consultar al igual que cualquier tabla creada por el usuario/diseñador.

En efecto, el catálogo del sistema produce automáticamente documentación de una base de datos. Cuando se agregan nuevas tablas a la base de datos, esa documentación también permite que el RDBMS verifique y elimine homónimos y sinónimos. En términos generales, Jos **homónimos** son palabras que suenan igual pero que tienen significados diferentes. Por ejemplo oso (animal salvaje) y oso (del verbo osar), o palabras que se pronuncian de manera idéntica pero con diferente significado, por ajemplo Tarifa (una península) y tarifa (de precios). En el contexto de una base de datos, la palabra homónimo indica el uso del mismo nombre de atributo para marcar diferentes atributos. Por ajemplo, se podría usar C\_NAME para marcar el atributo del nombre de un cliente de una tabla CUSTOMER y también usar C\_NAME para marcar el atributo del nombre de un consultor de una tabla

## •

CONSULTANT Para reducir la confusión, se deben evitar homónimos en bases de datos, el diccionario de datos es muy útil en este aspecto.

En el contexto de una base de datos, un **sinónimo** es lo opuesto de un homónimo e indica el uso de nombres diferentes para describir el mismo atributo. Por ajemplo, carro y auto se refieren al mismo objeto. Deben evitarse sinónimos.

**Contenido de un registro del diccionario**

Todas las partes de un sistema de información (transacciones, consultas, reportes, salidas, archivos y bases de datos) dependen de los datos. El diccionario contiene dos tipos de descripciones para el flujo de datos dentro del sistema : elementos de datos y estructuras de datos.

* **Elemento dato:** son los bloques básicos para todos los demás datos del sistema, por si mismos no le dan un significado suficiente al usuario. Se agrupan para formar una estructura de datos. Cada entrada en el Diccionario de Datos consiste de un conjunto de detalles que describen los datos utilizados o producidos por el sistema. Cada uno esta identificado con un nombre, descripción, alias y longitud, junto con el intervalo de valores específicos para el dato permitidos por el sistema bajo estudio.
* **Estructura de datos:** es un grupo de datos elementales que están relacionados con otros y que en su conjunto describen un componente del sistema. Los flujos y los almacenes de datos son estructuras de datos. Están formados por elementos relevantes que describen la actividad o entidad bajo estudio.
  1. **Funcionalidades del lenguaje de programación establecido**

Tanto el software de sistemas como el de aplicaciones están escritos en esquemas codificados y se les conoce como **lenguajes de programación.**

La principal función de un lenguaje de programación es proporcionar instrucciones al sistema de la computadora para que pueda realizar una actividad de procesamiento.

## •

Profesionales especializados en sistemas informáticos trabajan con el lenguaje de programación, el cual consiste en grupos de símbolos y reglas utilizados para escribir códigos de programas. La programación incluye Ja traducción de lo que quiere lograr un usuario a un código que pueda comprender y ejecutar la computadora. Al igual que al escribir un informe o un ensayo en español, Ja escritura de un programa de computación en un lenguaje de programación requiere que el programador siga un grupo de reglas. Cada lenguaje de programación utiliza un grupo de símbolos que tiene un significado específico. Cada lenguaje también tiene su propio grupo de reglas, o la sintaxis del lenguaje. Dicha sintaxis establece cómo deben combinarse los símbolos en enunciados capaces de hacer llegar instrucciones significativas al CPU.

En la actualidad podemos encontrar diferentes tipos de lenguajes de programación con los que trabajar (Java, PHP, C++, C#, etc.); por lo tanto, las funcionalidades van a depender del tipo de lenguaje de programación al que estemos haciendo referencia.

* 1. **Definición de librerías, clases y métodos del lenguaje de programación**

En este punto vamos a ver Ja definición de librerías, clases y métodos del lenguaje de programación.

* + 1. **librerías del lenguaje de programación**

En informática, una biblioteca del inglés *(library)* es un conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca.

A diferencia de un programa ajecutable, el comportamiento que implementa una biblioteca no espera ser utilizada de forma autónoma (un programa sí: tiene un punto de entrada principal), sino que su fin es ser utilizada por otros programas, independientes y de forma simultánea. Por otra parte, el comportamiento de una biblioteca no tiene porqué diferenciarse en demasía del que pudiera especificarse en un programa. Es más, unas bibliotecas pueden requerir de otras para funcionar, pues el comportamiento que definen refina, o altera, el comportamiento de Ja

biblioteca original; o bien la hace disponible para otra tecnología o lenguaje de programación.

Las bibliotecas pueden vincularse a un programa (o a otra biblioteca) en distintos puntos del desarrollo o la ajecución, según el tipo de vínculo que se quiera establecer.

La mayoría de los sistemas operativos modernos proporcionan bibliotecas que implementan Jos servicios del sistema. De esta manera, estos servicios se han convertido en una "materia prima" que cualquier aplicación moderna espera que el sistema operativo ofrezca. Como tal, la mayor parte del código utilizado por las aplicaciones modernas se ofrece en estas bibliotecas.

* + 1. **Clases**

En la unidad didáctica primera hicimos referencia a las clases cuando hablamos de la programación orientada a objetos.

Una clase es un patrón o plantilla en la que se basan oQjetos que son similares.

Cuando un programa crea un objeto de una clase, proporciona datos para sus variables y el objeto puede entonces utilizar los métodos que se han escrito para la clase. Todos los oQjetos creados a partir de la misma clase comparten los mismos procedimientos para sus métodos, también tienen los mismos tipos para sus datos, pero los valores pueden diferir. Una clase también es un tipo de datos. De hecho una clase es una implementación de lo que se conoce como un tipo abstracto de datos. El que una clase sea también un tipo de datos significa que una clase se puede utilizar como tipo de datos de un atributo.

**Tipos de clases**

En los programas orientados a oQjetos hay tres tipos de clases: clases de control, clases entidad y clases interface.

* + - * Las **clases de control** gestionan el flt¿jo de operación de un programa (por ajemplo, el programa que se ajecuta es un objeto de esta clase).
* Las **clases entidad** son las que se utilizan para crear objetos que manajan datos (por ajemplo, clases para personas, objetos tangibles o eventos) .
* Las **clases interface** son las que manajan la entrada y la salida de información (por ajemplo, las ventanas gráficas y los menús utilizados por un programa).

En los programas orientados a objetos, las clases entidad no hacen su propia entrada/salida. El teclado es manajado por objetos interface que recogen los datos y los envían a los objetos entidad para que los almacenen y los procesen. La salida impresa y por pantalla la formatea un oqjeto interface para obtener los datos a visualizar de los objetos entidad. Cuando los objetos entidad forman parte de la base de datos, es el SGBD el que se encarga de la entrada/salida a ficheros. El resto de la entrada/salida la manajan los programas de aplicación o las utilidades del SGBD. Muchos programas orientados a objetos tienen un cuarto tipo de clase: la clase contenedor. Estas clases contienen, o manajan, múltiples objetos creados a partir del mismo tipo de clase. También se conocen como agregaciones. Las clases contenedor mantienen los oqjetos en algún orden, los listan e incluso pueden permitir búsquedas en ellos. Muchos SGBD orientados a objetos llaman a sus clases contenedor *extents* (extensiones) y su objetivo es permitir el acceso a todos los oqjetos creados a partir de la misma clase.

**Componentes**

Una clase es un contenedor de uno o más datos (variables o propiedades miembro) junto a las operaciones de manipulación de dichos datos (métodos). Las clases pueden definirse como estructuras *(struct) ,* uniones *( unían)* o clases (c/ass) pudiendo existir diferencias entre cada una de las definiciones según el lenguaje. Además las clases son agrupaciones de objetos que describen su comportamiento.

La sintaxis típica de una clase es:

**class Nombre** {

/ / **\!ar ia.b les miernbro ( l".Labitualrr1e11te pr i..,l'adas)**

**miembro l; / / l ista de miembros**

miei:nbro-2; miembro 3;

/ / **Func ior,1.es o :métodos ( habitu.almente públ ieas)**

**f uncion\_:mierribro\_l** ( ) ; / / **f 11nc iones mie:tribro conoc idas f unc ion\_miembro\_2** ( ) ; / / **f unc iones corC10 :rr1étodos**

/ / **Prop iedades ( hab itualrnente púlo l icas)**

propiedad\_1; propiedad\_2; propiedad\_3; propiedad\_-'!;

Las clases habitualmente se denotan con nombres abstractos como Animal, Factura... aunque también pueden representar procesos o acciones como DarAlta.

*Variables miembro*

Las propiedades o atributos son características de los et/jetos. Cuando definimos una propiedad normalmente especificamos su nombre y su tipo. Nos podemos hacer a la idea de que las variables son algo así como el almacén de los datos de estado relacionados con los oqjetos.

Habitualmente, las variables miembro son privadas al objeto (siguiendo las directrices de diseño del Principio de ocultación) y su acceso se realiza mediante propiedades o métodos que realizan comprobaciones adicionales.

Suelen denominarse con **nombres.**

*Métodos en las clases*

Implementan la funcionalidad asociada al oqjeto. Los métodos son el equivalente a las funciones en programación estructurada. Se diferencian de ellos en que es posible acceder a las variables de la clase de forma implícita.

Cuando se desea realizar una acción sobre un objeto, se dice que se le manda un mensaje invocando a un método que realizará la acción.

Habitualmente, los métodos suelen ser **verbos.**

*Propiedades*

Las propiedades son un tipo especial de métodos. Debido a que suele ser común que las variables miembro sean privadas para controlar el acceso y mantener la coherencia, surge la necesidad de permitir consultar o modificar su valor mediante pares de métodos: GetVariable y SetVariable.

Los lenguajes orientados a oQjetos más modernos (por ajemplo Java, C#) añaden la construcción de propiedad que es una sintaxis simplificada para dichos métodos:

tipo Propiedad

get

'*l*

set

Las propiedades se denominan con nombres como las variables.

* + 1. **Métodos**

En la programación orientada a objetos, un **método** es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, cómo es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia. Análogamente a los procedimientos en los lenguajes imperativos, un método consiste generalmente de una serie de sentencias para llevar a cabo una acción, unjuego de parámetros de entrada que regularán dicha acción o, posiblemente, un valor de salida (o valor de retorno) de algún tipo.

Algunos lenguajes de programación asumen que un método debe de mantener el invariante del oQjeto al que está asociado asumiendo también que éste es válido cuando el método es invocado. En lenguajes compilados dinámicamente, los métodos pueden ser oQjetos de primera clase, y en este caso se puede compilar un método sin asociarse a ninguna clase en particular, y luego asociar el vínculo o contrato entre el oQjeto y el método en tiempo de ajecución. En cambio en lenguajes no compilados dinámicamente o tipados estáticamente, se acude a precondiciones para regular los parámetros del método y postcondiciones para regular su salida (en caso de tenerla) . Si alguna de las precondiciones o postcondiciones es falsa el método genera una excepción. Si el estado del objeto no satisface la invariante de su clase al comenzar o finalizar un método, se considera que el programa tiene un error de programación.

La diferencia entre un procedimiento (generalmente llamado función si devuelve un valor) y un método es que éste último, al estar asociado con un oQjeto o clase en particular, puede acceder y modificar los datos privados del oQjeto correspondiente de forma tal que sea consistente con el comportamiento deseado para el mismo. Así, es recomendable entender a un método no como una secuencia de instrucciones sino como la forma en que el objeto es útil (el método para hacer su trabajo). Por lo tanto, podemos considerar al método como el pedido a un objeto para que realice una tarea determinada o como la vía para enviar un mensaje al oQjeto y que éste reaccione acorde a dicho mensaje.

1. **Herramientas**

**para**

**la creación y el**

**mantenimiento del diccionario de datos**

El diccionario de datos es un documento que corresponde al diseño del sistema. Por lo tanto es independiente al sistema de bases de datos y al lenguaje de programación usado en el sistema. Aunque con el auge de los frameworks de mapeo ka elaboración manual del diccionario de datos haya caído en desuso, Ja documentación del proyecto siempre estará incompleta si no existe el diccionario de datos.

Queda claro que la creación del diccionario de datos pertenece al diseño de Ja base de datos, sin embargo en la actualidad podemos elegir entre diferentes herramientas o programas para la creación de un diccionario de datos. Nosotros vamos a centrarnos en el programa **Oracle9i Designer.**

El diccionario de datos es quizá una de las partes más importantes de Oracle. Se trata de un cofliunto de tablas de sólo lectura que mantiene los **metadatos** (descripción del esquema) de una base de datos. Esta compuesto por tablas y vistas que facilitan el acceso de los usuarios a la información.

* + **Información:** nombres de usuarios, seguridad (accesos permitidos, privilegios y roles), información sobre Jos objetos, restricciones de integridad, asignación de espacios y diversas estadísticas
* **Las tablas del diccionario:** son propiedad del usuario SYS. El usuario SYSTEM es propietario de diversas vistas sobre esas tablas, las cuales pueden ser utilizadas por el resto de los usuarios de la base de datos.
* **Prefijos** en las vistas del diccionario de datos:
  + USER\_ OQjetos que pertenecen al propio usuario. o ALL\_ Todos los objetos accesibles por el usuario.
  + DBA\_ Todos los objetos existentes (sólo para el uso del ABO).

Para la implementación de un diccionario de datos con Oracle9i Designer se deben seguir los siguientes pasos:

* + - 1. Cargar la herramienta de Oracle9i Designer versión 9.0.2.94.11, el cual muestra la siguiente ventana :



**.!.J.sername**

**f::as:::s:word**

**Connect**

**Sl:ring**

IUNIUPDB

-

-----------------·----

OK

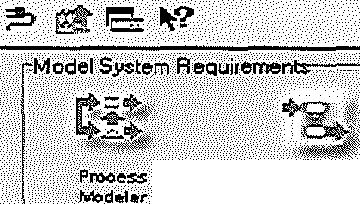
**1**

Cancel Help

Se digita el nombre del usuario, la contraseña y conectándose a la base de datos de desarrollo.

* + - 1. Se realiza el proceso de ingeniería inversa:

## •



**EJle ew Iools tlelp**

TransfOIJll Ptelílrinar º"'¡"·.··•••••.•.•..·• r [º"'ÍJIO ,ond G"'Jor

'

-

;

j

; *:,;* **i,J'.:1\_ 1101.Ú**

- **D:rpm,n:ier**

**'.Fuhi:itlon**

- **'Hk:mrohy** \_

**o;agr21ním\_er\_** -

**'.tl31:1b:a-.:.**

**:'.De'•Js;in**

- **'Tmtdi'ló r**

'

• .;i

**Ji;pp1ie\_:.són**

**-;ce:;ign·**

**:l'ta'fi$'1'(tl'm**

- **-:Jt'V**

**-:!Jt'\_OJ'd:hi¡:i**

·**eianmmiecr**

**-- )t**

**01\_3,Q·ni· ;;**

* **:Fl;p\_oÓt;I\_** \_

**-:\_N'.illo\_n**

**:Ut-lrrt)r\_**

**-,,-epoSlt\_ory**

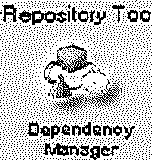
**\_obl,tít**

**'tlbvigator'**

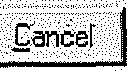
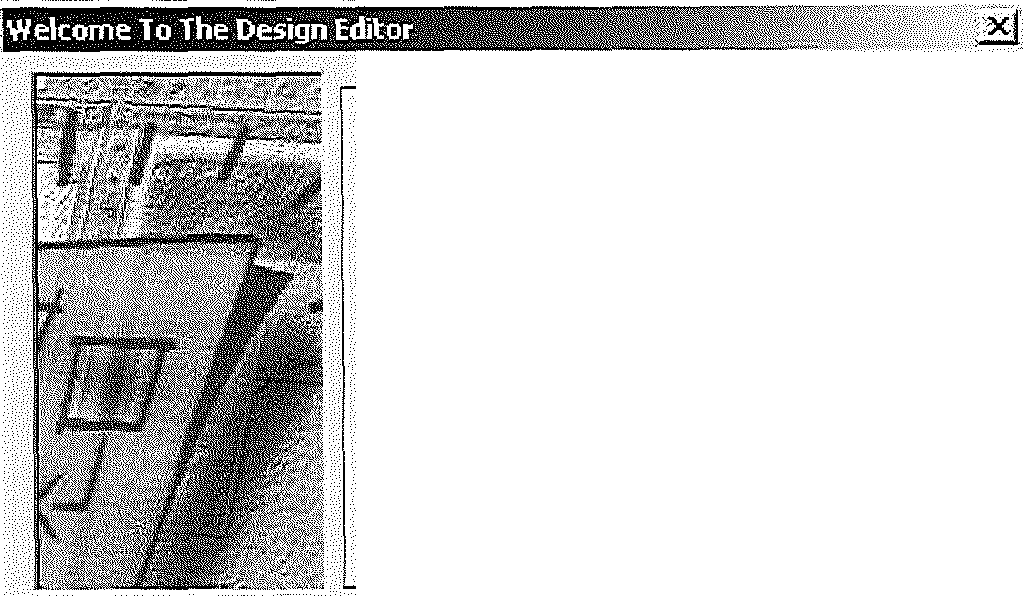
**Rj)6$itOÍy**

**JcpOrt$\_**

Seleccione el icono Design Editor:



Where·•would you like>tosta1t?-------



t"'I .. .

"'ffi.

¡5!1.·

· ·

r···:Dat<llíaseé\_dminfhlration

*E* OperiNewDia&iram



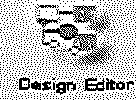
- "- ·- .- - -.-- --·

•P"· Displ<lÚhb$Cleenon startup

f!élp . ,



. . ..



Seleccione la opción **Server Model:**



Th s Guide tak.es you thl'ough wahat you need to do to ere.ate databas:e obiects: that make up your s:el'ver model de:sign

Choo:se where .YOU want to :start

1 lt-- CreatelE d1t Databsse O biei:: t:s

Run the D eign Capt1..Jre util tv

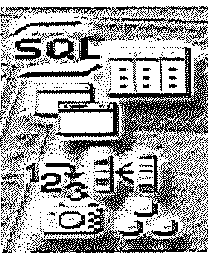
1 .

,Gontext Container

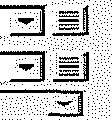
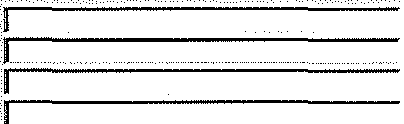
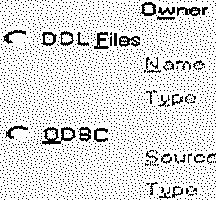
 

I.rack Selection in Navigator

Posteriormente, seleccione el contenedor en el cual va a realizar el proceso.



Luego haga clic en la opción Run the Desing Capture utility.



Source:l}:r.\ ii;:.J\_:\_-\_-\_

So>,,.1roc of O 0'9ign Capl:i...ireo-

i:.?.!?.$

g::;em.o.rnc

E:assword

onnOct

1

------------------------------------·--·--------

T rget Cont.ei1r'IOt

C.opture lmplementcticns: tnto

Additional Conteiners

I?". Stop on -.orror

(;;;' Cop,••m:;t Col"lSraints:

Post Dc:isn Cept•.ue:

Browse...

P Show Flesults on new Sor'y'(!)'r Model Diogram

R C-l:llpturcdndci-1e::;

P' Ceptun::o Tti99et:s=

Commit

*C* erow::.elE dit modo

Diligencie la anterior ventana detallando lo siguiente: el nombre del usuario de la base de datos, la contraseña, el nombre de la base de datos, los demás valores los toma por defecto. Posteriormente, selecciones la ficha Objects:



Source - Ob)éct:s: J

**Av.aileible ob¡ecls: for Capture..••**

Dori't Cature

MER\_Ao:.\DEMUSOFT\_32.DEFAJ

!-..D Cluster:<;::

I·-· CJ Colloetion TJJpe<:S

1¡---:J Functions

\_!-..CJ Jva Obiects

;---CJ Moteriolizod Views

i·""O Non-Persistont Queues

-·· D Obie T..:ible">

:\_.. [J Obiect Views

!-<::D Or.:icle DOiect T}'pes

$·- [) Pacae:s

",.-- CJ PO"::ment Q ueues

¡!¡.·CJ Procedures

1-- - D Queue Tables

$...CJ Relational T able:s: i·-· D Relational Views El·· D Sequences

Él·.. CJ S;,inonyrm:

'""""----- ···.··. .....!.l

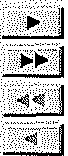
noel.?lr

Seleccione la opción Relational Table, agregando las tablas necesarias para Ja construcción gráfica del modelo de datos.



\_S\_oureo ·ab¡O:!.CtS: "I

* + - * 1. **vaiil.oble obiects fÓI'- C-.1:1ptu11te•..**



Don'J: C.:.pt1..o1ro

S··L;:J Relational Teblie:s:

*i* ····L:!Iill ACTIVI DAD

*i* .... AREAICFES

····!filfil ASOCIACIONCIENTIFI

····EETIJ ASPECTDSPR OGRAl'Y

Jir1íl ASPIRANTE

¡...::AUD\_A AEACONOCI M

i,.··.·.·. .. :.., AUD\_A AEAICFES

l.l.c.!!J AUD\_ASPECT OSPR O

i-····ffiilil AUD\_CALIFIC,.O..CION

1 ··-- Ef:fil AUD\_CANCELACION:::

1-- ··Effi AUD\_CATEGOAIAESC

1 -···!IElII AUD\_CJCLOCURRICU

* AUD\_CICLOPA OGRA

·-··füII AUD\_CIRCUNSCRI PC

···Ef:tl AU D\_CLASE SEMANA

····fil[) AUD\_CLASIFICACr;i

•••• R;:;::t A l J I""'> \_ r- "-"Drtr..lr:::r.,.1-:fJ ...

C.eipturc

MER\_ACADEM USOFT\_32. DEFAl

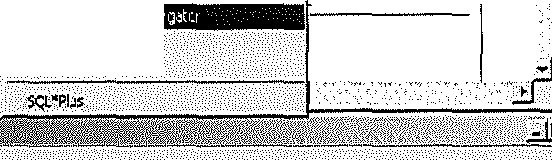
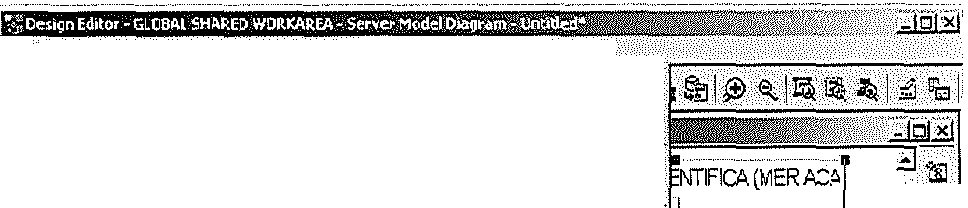
St.:;rt Cancelar **1** Ayudo



Clase:

Luego seleccione la opción de Stara, terminando dicho proceso, clic en

i f



CDS-15. 317 \llJJJrning: inwlid Tablesp<?ice reference \ACADEMIC0 1inTJJble 'ASOCl.6.00NCIEtffiFICA'. igncnd *J*

CopyiEh: (e) Orac e·19S5 ¿002. .e.11 rig'lt;; reseMd. , ·­

CDS-1.5319 Warning: imialirl Tablespace reference rAC"A.OEMIC0:1 in User Object lnde)( Storage for 'ASCl\_PK' . ;g'I

Proces5hf Compl\*: O erro ). 2 warning(s) .

' "

li1;.J:a:-c;·-

} G.1 liil %¡ 1 ::JI 1

.-Pro¡:i,\_rty P.&ltte

))it.!b: iQ"tor:..

+o S:lri:'e;Mo(erJ i¡·.tiii°fDB'mi1"1 \r,lori1

'h

i¡:: "EP.INSCff



-ÓA;sdemlco

-dCONT.ASIUO>.D i¡::-Ó CONTPATOSUP i¡::-Ú CONVENIOS

i¡::-,:1CcntaOllldad

-él Etesados

*o¡-ú* OE"PA.



-dGenerales ':'-ÜHOSPIX E-Ci' MA.TP.ICLLA

(.$-:.-.d-' M....A....T...P....I.C. l.W-\2·................

....

f'lcdubD\_5):i;..í't \V1;i;;i(d

f:o..i\_i:ible(orrr,t•r1o::r.: rwll1ui1 tdt...

-rMod-ISi.id,,,

MtciYieiéatbr G.Jld·e·., lD

c:·Adm[ r!aJ SU,\_ . ---tNCtMBRE

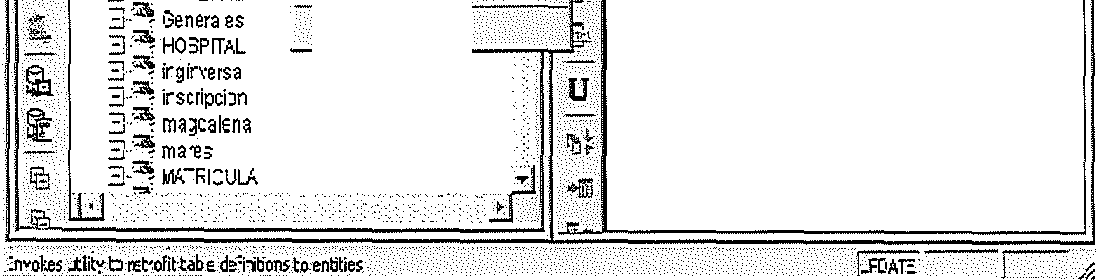
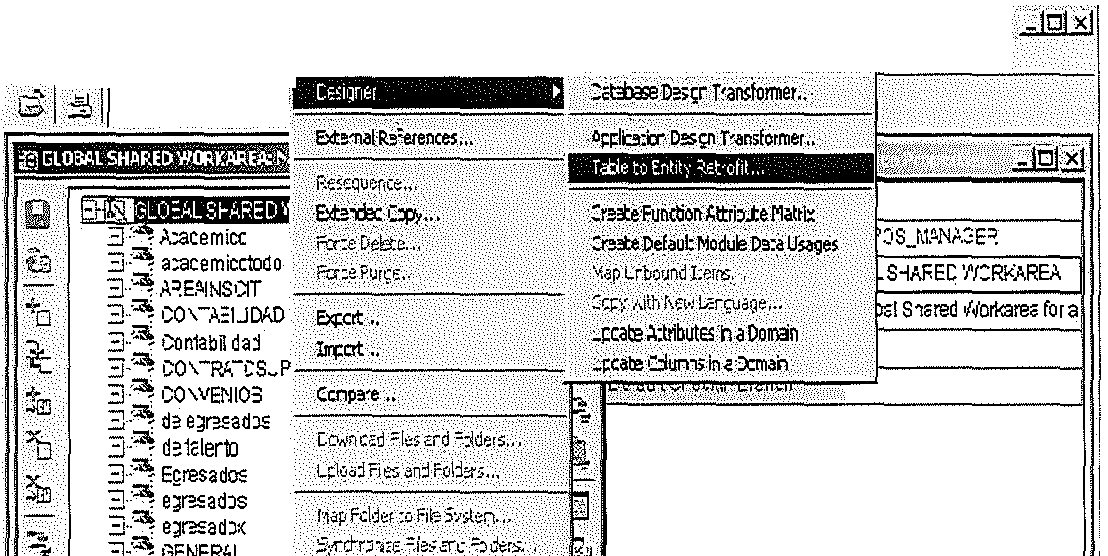
R((»\lcj)í;'rl:s; OEJE1030C!AL

aá".'Ch-GnC'.-atc-· FECKAll'lGRESO

!kl:ch:D:;:;b.lo REGISTR.ADOPOR FECHACAMBIO

-endené'/lige:r

trx :iiaor ér



Qlij.li. !M1J§i§.jf fü@#i§•ti'·1f jt- j@i#1 i@ ij1::§Aia :Qi1Ji.J·t't·% 1'f@l:tj D1liM'...fY.'.:.:,

Fie Idi:' iew ·N9ator ·.1ieistm.!·.o;Jitiei- Ti»:s OpbiS :.¡Jccw P

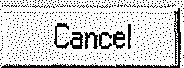
EdlFleF.ego .,

## •

· i:,;

,J

* tf



Seleccione el menú **utilities/Designer/Table to Entity Retrofit,**

apareciendo después del proceso la siguiente pantalla .



Contai er 11.cademico

.[;;reate ···· ¡

- • • =cc==c= 7•=••c•=••=•·=-=

OK

**1**

Help

Seleccione el contenedor:



stem

Owner

....

REPOS\_MJ.Nl.G

acadero.icotodo 1.REJ.INSCIT CONTJ.BILIDJ.D

REPOS\_Ml.Nl.G . INSCITJ.CION REPOS\_Ml.Nl.G

Contabilidad REPOS\_Ml.Nl.G

-2;;; U\_P

CONTJ.BILIDJ.D/pagaduria REPOS\_Ml.Nl.G .

.R

1

,; s-¿¿\_:]

.,,,·--·-·--w'"-·--;-;--;-;·----;<c-;·c: ;--- -- -o--;:-c:-7\_-,,- ·•·:

··

. . L •• • •.. . •·



.

.

.

..

.

.· . .

. .·..·

.

r: Updatejdniqué identifiersfor eisting entities

··c;;;;id;1'b1.;;··1·

Grid ave *{*

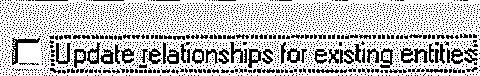
ftetrofit

.(;lose j

J::!.elp · 1

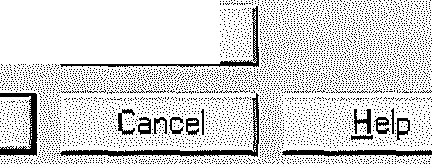


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| •··· | .•••... ·. t61eNám . ·•· | ··.•···•••. .E.ntit,YName · ··. |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |





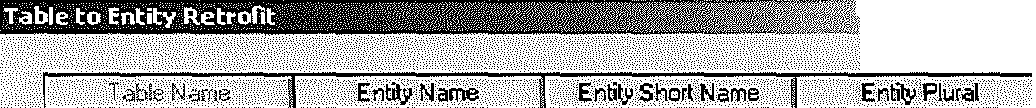
Y haz clic en **Candidate Tables** para seleccionar las tablas:



Select All

1

OK



fJ1\1iJó?J :;:'.\_:\_: -\_<-- - **\_:gj**

- -- - \_ --:--:\_ --:< - - :;':-\_,-\_

'

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| ¡¡<;OCIACIONCIENTIFJéi | ASOCI ACIONCIENTIFIC |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |





- - - < - - --

:-:. " - - -

*\_,":*':\_,- ,·;·, ''- '" '• -n\_ ---:-· ,' ' ,:,··,: *:* ' > ,' ' •:

['. Updi)le,télatíophl..ipsfpi ¡;,\ífi,og er\'tiei.•.

Ir- . .. .. ..

w· 1

**-... ! te ab.[e\_&** ...:..J

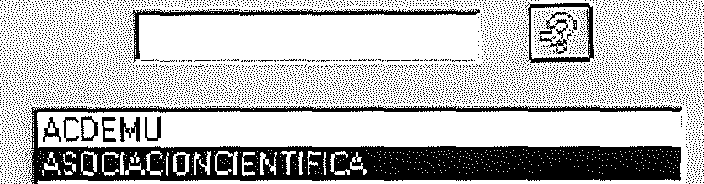
6ri.l l

··· !JUe!iní9ueidel1titierdor.,,.Sting•Einnlies

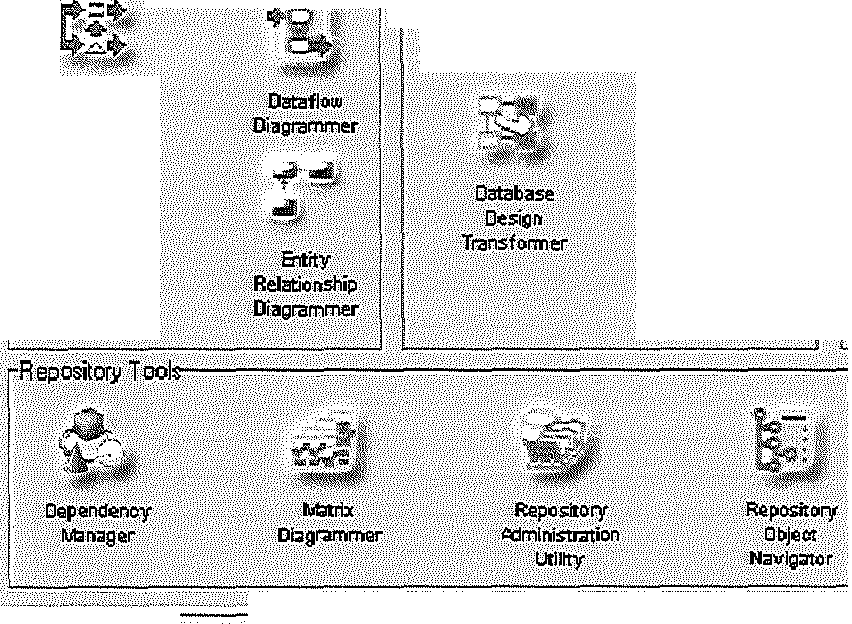
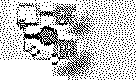
fietrolit 1

···§ridF1; ; ¡ J:!elp

Posteriormente, haz clic en **Retrofit/Close/Close.** Finalmente, falta construir gráficamente el MER, para lo cual vamos a la siguiente ventana:



Elle l'.jew Iools t!elp



Mode S.l!stemRéi:iuÍreJÍlén't"s'"."'"'"""' , ·•f1ran•loiíti IéliMinaÍlf'li!e ,gn• .• > .• ... Deiign ariq Generat.

1

'

**;pJ'OC;d!i'**

**-,-l'ubdeleí-**

·

'1'

Y··.

,, ,,

**/'-,Pj¡**

**.,n\_**

**Oe\_sig,r{it01r**

.

.;.•\.

, **:\_-J:)\_e\_sig -'\_**

**Funétioll**

**'Jran:sformeÍ"**

**::J;ifirit.y**

**-, gra!J)\_Íner,**

**Ré¡)ICO('j¡**

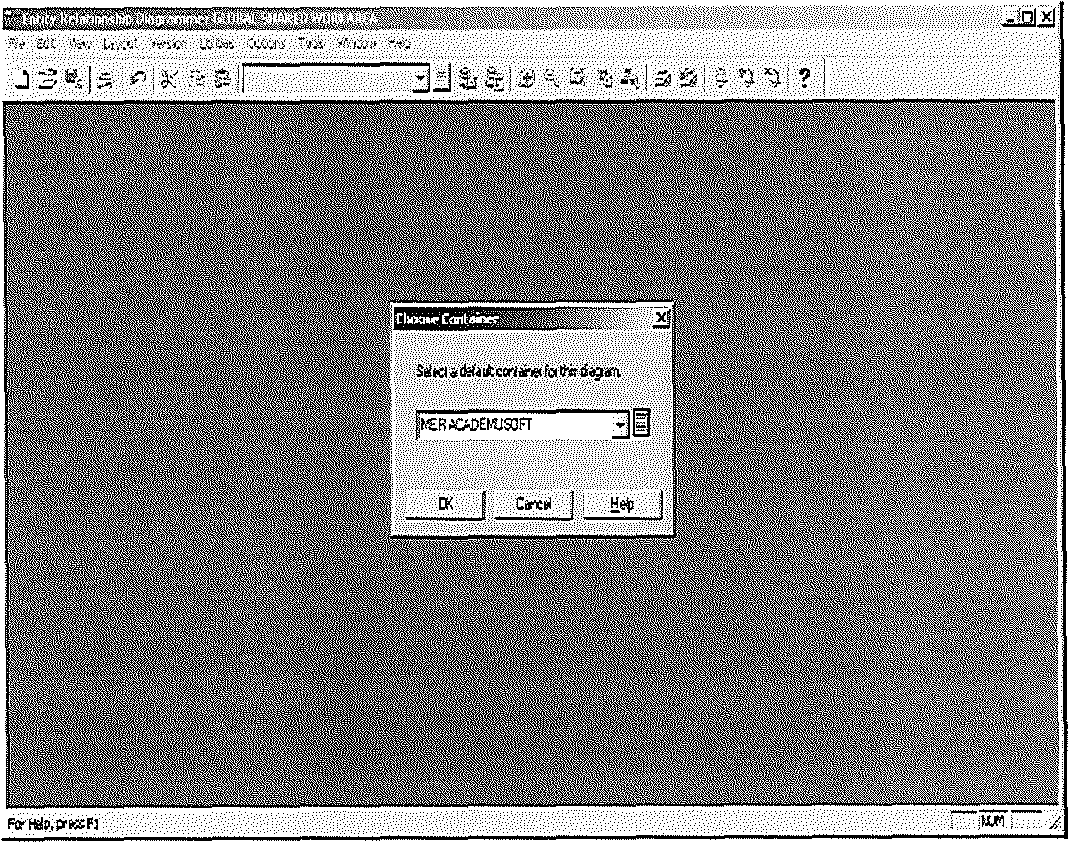
**RQp\_trtts** ,

IREPOS.:Jo!ANAGER

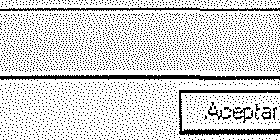
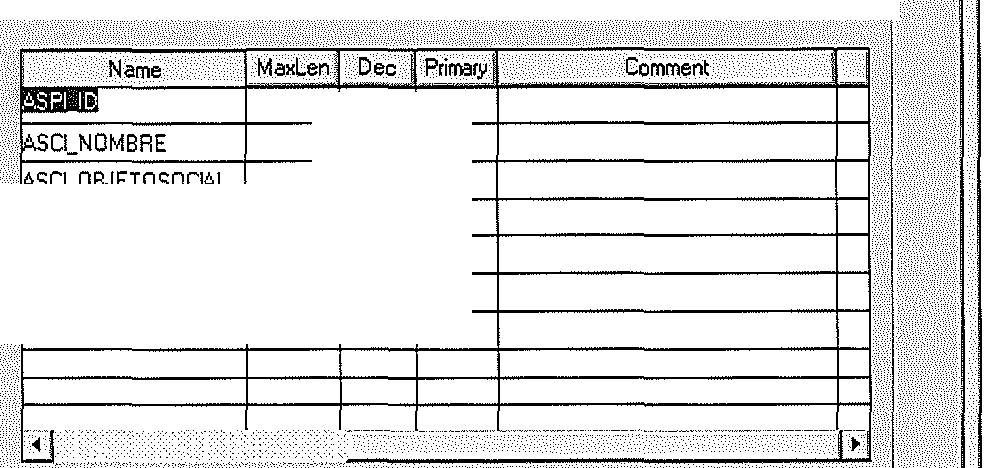
jUNIUPDS

i<l'LOSAL SHARED WORKAR.EA

Seleccione la opción **Entity Relationship Diagrammer:**



Seleccione del menú File/new/ seleccionando el contendor y proceda a incluir las tablas Edit/lncluye, seleccionando las tablas deseadas. Ahora adicione los comentarios de cada columna o campo a las tablas describiendo la funcionalidad de la tabla y el objeto del campo. Esta ventana carga cuando se selecciona la tabla incluida y se da un Enter. Al terminar guarde.



l@ffiktjfj@á !ff #;!i-.-,.J§ilti-l#i·ii-;1-ff"t·iff U@13i-7;igi§if·­

Q\_•finitionJ fuin<> I!!IDs 8ttributes j tttltn r tt. ··I ié<t l

.l\t1ributes-·,.,· "'"""""'C==0'C="""=""""'C="""=="""""'"""""c='=c'="""""'l

**Prim:ir\_y**

30

50

100

30

1

1

Ei

Ei D D D D D

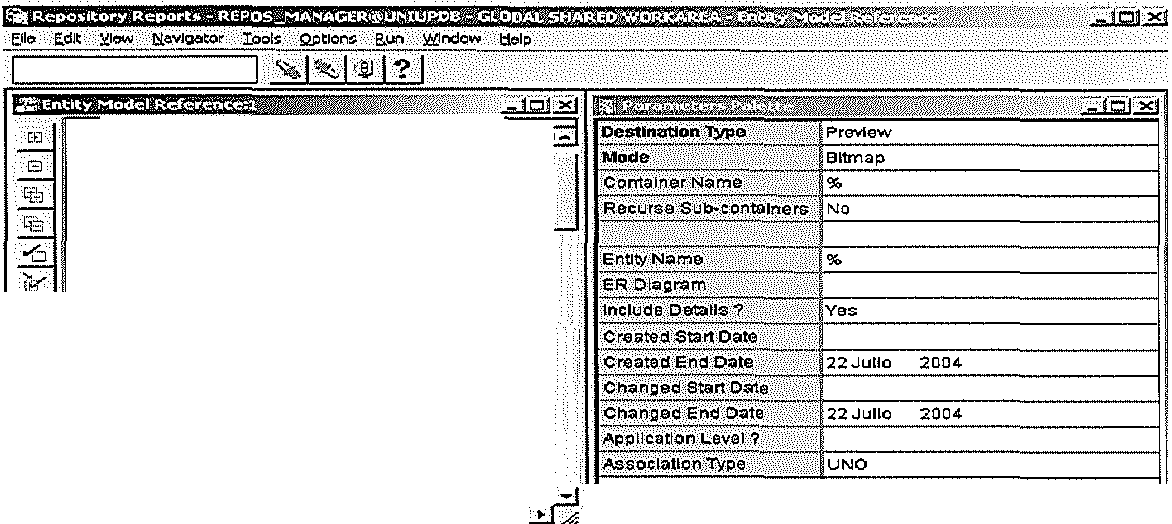
**lnsert Row** .J

Delete Row 1 R t Df ltj

Cancelar

8Yuda

Finalizado el proceso de descripción de tablas, se realiza mediante la herramienta el diccionario de datos, el proceso anterior se debe hacer con todas las tablas de la base de datos. Después seleccione el menú TOOLS/Repository Reports.



|  |  |
| --- | --- |
|  | |
|  | ¡;:}-0 **EnUl:yJRelonshlp Modelllno**  ,--·--1]] Entlty De'flnltlon - ck>antdef  1 L--·O Entltles and thelr Attrlbutes - ckatltl14  1 :..--r:::J. Attrlbute De'flnitlon - ckattdef  i L--0 Attrlbutos In a Domaln - cddom:2  ' L-[!) Functlon to Entlty wtatrlx- cdrentm  ¡I r/5 **1@¡&!1@&,:,1:.g::.**4a1rlx **G.j. &iTa:**  - " Dll System Olossa¡y- Ck!JIOSS  - -!El <Element> De'flnttlon - ckueeld  - • IJ::l Access to Modules by Roles - Cdugrp  r ----Lml) ActMty Sased Casting - ckaOcos1  ¡-- m Ali Text Htald agalnst an Element- cdeltxt  Area Metrlc - cdmetr1c:  r· --[[] Assoclauon Type Deflnltlon - c:du-edel'2  1-- Dll Business **Unlt** Doflnl1ion - c:k:bund0f  1...-[JI Chanped Erements- cdc:hange  ·- - Grl CluS'ler OaTlnLtion - cdocl  '--Eil Co11.1mn Oeflnltlon - ck:coldt  r-·-G:l Column Olsplay Usagec (byTable)- C!J r'bb  l·--·O Columris In a Domaln - cddom1  '··--0-3 Common Functr ons - cdmfun , . |
|  |  |

Despliegue Entity Relationship Modelling y seleccione Entity Model Referente ckemordrf posteriormente, llene la plantilla que aparece, en la propiedad ER

diagram, busque el nombre del archivo con el cual guardó anteriormente.

Es recomendable, generar el archivo en Word, para facilitar el arreglo de la información.

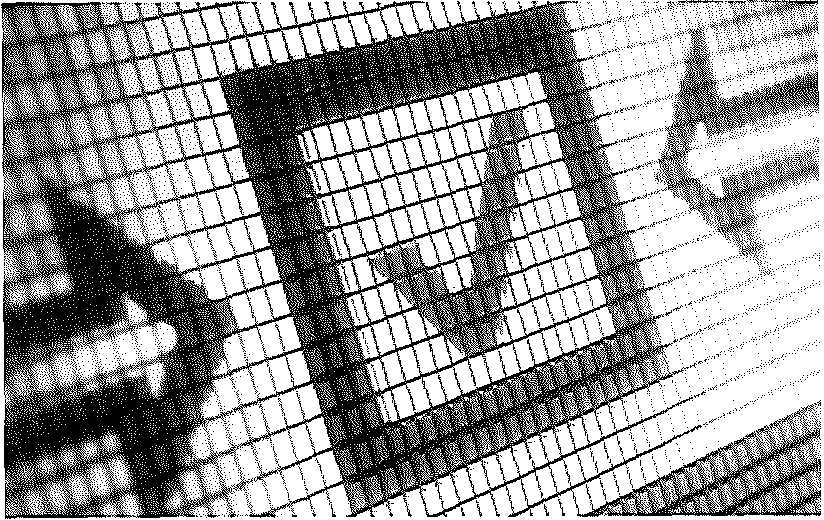
1. Crear Diccionario:

Se crea los archivos que contengan el diccionario de datos según las especificaciones o diseños contenidos soportados por la herramienta, teniendo en cuenta que cumpla con los requisitos mínimos de:

* + Modelo Relacional.
  + Descripción de tablas.
  + Nombre y descripción de los campos de las tablas.
  + Nombre y descripción de las restricciones.
  + Tipos de dato, longitud y escala de los campos de las tablas.
  + Aceptación o no de valores nulos para los campos de las tablas.

**Mantenimiento de un diccionario de datos**

El Sistema Gestor de Base de datos es el encargado de mantener y reparar el diccionario de datos.



Un **Sistema de Gestión de Bases de Datos (SGBD)** es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Los SGBD también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y recuperar la información si el sistema se corrompe. Permite presentar la información de la base de datos en variados formatos. La mayoría de los SGBD incluyen un generador de informes. También puede incluir un módulo gráfico que permita presentar la información con tráficos y cartas.

Hay muchos tipos de SGBD distintos según manajen los datos y muchos tamaños distintos según funcionen sobre ordenadores personales y con poca memoria a grandes sistemas que funcionan en mainframes con sistemas de almacenamiento especiales. Algunos ajemplos de SGBD son Oracle, 082, PostgreSQL, MySQL, MS SQL Server, etc.

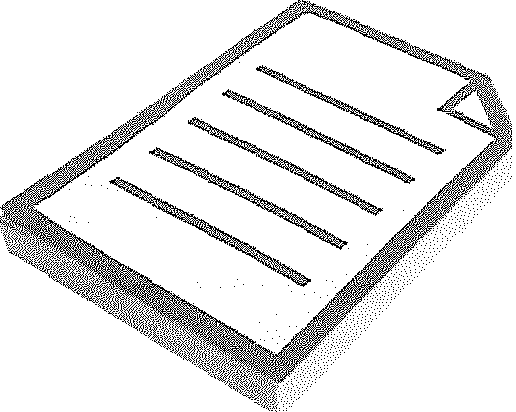
Generalmente se accede a los datos mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican la interrogación y la presentación de la información. Un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las Bases de Datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero.

Un SGBD debe permitir:

* + **Definir una base de datos:** especificar tipos, estructuras y restricciones de datos.
  + **Construir la base de datos:** guardar los datos en algún medio controlado por el mismo SGBO.
  + **Manipular la base de datos:** realizar consultas, actualizarla, generar informes.

Las características de un Sistema Gestor de Base de Datos SGBD son:

* + - **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
    - **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
    - **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complajidad de los cálculos hace necesaria la aparición de redundancias.
    - **Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
    - **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
    - **1 ntegridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
* **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
* **Control de la concurrencia.** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
  1. **Definición de herramientas de documentación**



Generalmente, la creación de la documentación del diccionario de datos está relacionada directamente con el programa de diseño, que es el encargado de generarla automáticamente al final del proyecto.

## •

**RECUERDA**

* + - El diccionario de datos es un repositorio de **metadatos** que contiene las definiciones de los objetos de datos, descripciones y relaciones entre sí. El diccionario es muy importante en el desarrollo de aplicaciones, ya que permite a los involucrados en el desarrollo conocer a detalle las bases de datos.
    - Todas las partes de un sistema de información (transacciones, consultas, reportes, salidas, archivos y bases de datos) dependen de los datos. El diccionario contiene dos tipos de descripciones para el flujo de datos dentro del sistema: elementos de datos y estructuras de datos.
    - Tanto el software de sistemas como el de aplicaciones están escritos en esquemas codificados y se les conoce como **lenguajes de programación.**
    - La principal función de un lenguaje de programación es proporcionar instrucciones al sistema de la computadora para que pueda realizar una actividad de procesamiento.
    - En informática, una **biblioteca** del inglés *(líbrary)* es un conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca .
    - Una **clase** es un patrón o plantilla en la que se basan objetos que son similares.
    - En la programación orientada a objetos, un **método** es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, cómo es el caso de los métodos de clase o estáticos, como a un oeto, como es el caso de los métodos de instancia. Análogamente a los procedimientos en los lenguajes imperativos, un método consiste generalmente de una serie de sentencias para llevar a cabo una acción, unjuego de parámetros de entrada que regularán dicha acción o, posiblemente, un valor de salida (o *valor* de retorno) de algún tipo.
    - La creación del diccionario de datos pertenece al diseño de la base de datos. **Oracle9i Designer** es una herramienta que nos ayuda a crear y mantener el diccionario de datos.
* El Sistema Gestor de Base de datos es el encargado de mantener y reparar el diccionario de datos.
* Un **Sistema de Gestión de Bases de Datos (SGBD)** es un conjúnto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.
* Generalmente, la creación de la documentación del diccionario de datos está relacionada directamente con el programa de diseño, que es el encargado de generarla automáticamente al final del proyecto.

**Preguntas de Autoeval uación**

1. **Completa la siguiente afirmación con la opción correcta:**

"El diccionario de datos es un repositorio de que contiene las definiciones de los oqjetos de datos, descripciones y relaciones entre sí".

1. Datos.
2. Metadatos.
3. Tablas.
4. **Tanto el software de sistemas como el de aplicaciones están escritos en esquemas codificados y se les conoce como:**
5. Lenguajes de programación.
6. Métodos de programación.
7. Sintaxis de programación.
8. **¿Qué es una clase?**
9. Un conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca.
10. Un valor que no puede ser alterado durante la ajecución de un programa.
11. Un patrón o plantilla en la que se basan objetos que son similares.
12. **Confirma si la siguiente afirmación es verdadera o falsa:**

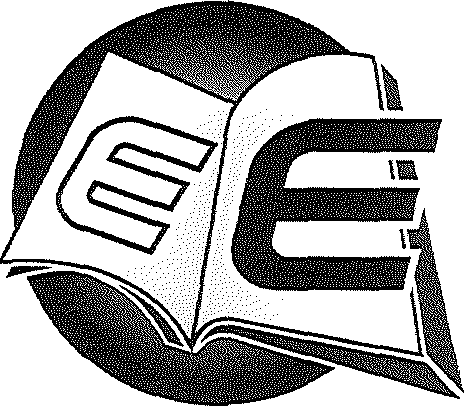
*"La principal función de un lenguqje de programación es proporcionar instrucciones al sistema de la computadora para que pueda realizar una actividad de procesamiento".*

1. Verdadero.
2. Falso.
3. **¿Cuál de las siguientes afirmaciones es incorrecta?**
4. La creación del diccionario de datos no pertenece al diseño de la base de datos.
5. Oracle9i Designer es una herramienta que nos ayuda a crear y mantener el diccionario de datos.
6. El Sistema Gestor de Base de datos es el encargado de mantener y reparar el diccionario de datos.

**Diseños de**

UD5

**pantalla**



UF1888 Operaciones de Mantenimiento y

C. o-" ns- u, l- ta, ,. '

d, ' e-

1. '

a" - t,os,

* 1. **Definición de pantallas de recogida de datos**
     1. **Herramientas para el diseño de tablas y**

**mantenimientos**

Básicamente, el diseño de una tabla consiste en definir los distintos campos y sus propiedades, pero a esto debemos llegar después de un análisis del problema y aplicando unas reglas lógicas, que proceden del modelo relacional, para conseguir un rendimiento eficiente, lo que se conoce como **«normalización».**

Empezaremos con el diseño lógico, es decir, pensando que campos en qué tablas necesitamos para trasladar el esquema del problema en el mundo real a una base de datos. Es algo que podemos hacer previamente sobre el papel o al tiempo que vamos creando las tablas, pero siempre teniendo en cuenta una serie de reglas lógicas derivadas del modelo relacional que veremos a continuación.

Luego entraremos en el diseño físico de las tablas, definiendo campos y propiedades, utilizando la vista de modo diseño de las tablas de **Microsoft Access.** Para ello necesitamos conocer los distintos tipos de campo y algunas de sus propiedades más importantes.

* + - 1. **Reglas lógicas**

La normalización es el proceso de aplicación al diseño de una base de datos de un conjunto de reglas lógicas, ya definidas, que tienen como objetivo garantizar la integridad de los datos, evitar redundancias y conseguir un funcionamiento eficiente.

Recuerde que son reglas lógicas. Con su cumplimiento, más eficiente será nuestra base de datos y más sencillo será su desarrollo en el futuro, pero siempre hay que tener presente la lógica, puesto que «no puede resultar que nos cueste más el seguro que la casa». Por eso, a veces, se rompe intencionalmente con esas reglas para alcanzar un oQjetivo preciso; es lo que se conoce como

**«desnormalización».** Por ajemplo, es frecuente utilizar varios campos para los distintos teléfonos de un contacto, cuando lo «normalizado» sería crear una tabla de teléfonos relacionada con la del contacto.

•

Durante el desarrollo del modelo relacional, el propio Codd definió un conjunto de reglas que se debían aplicar en el diseño lógico de una base de datos, a las que llamó «formas normales». Aunque posteriormente otros autores añadieron más reglas, las más conocidas son las tres primeras del propio Codd:

**1FN (Primera Forma Normal): Atomicidad.**

Las tablas deben tener una clave primaria que identificará de forma inequívoca y exclusiva cada registro. Esta clave no puede contener valores nulos, es decir, no puede quedar sin valor, pues así se garantizará que todos los registros sean únicos. Access se ocupa de sugerir siempre una clave primaria, aunque permite no ponerla o que usemos una creada por nosotros.

Los datos deben ser indivisibles, es decir, se tiene que buscar la unidad de información más pequeña. Por ajemplo, si creamos un campo Nombre y queremos ordenar alfabéticamente por apellidos, tendremos que crear tres campos (Nombre. Apellido 1 y Apellido 2) .

Debemos utilizar el sentido común y pensar en el propósito de nuestra base de datos a la hora de buscar la atomicidad y no llevarla a extremos exagerados. Un caso típico es la creación del campo destinado a incluir los datos de la dirección postal, puesto que esta se puede descomponer en población, vía, portal, número, escalera, letra, etc.; dependiendo del objetivo final de la base de datos, se podrán crear más campos (como algunos formularios de Hacienda, tan incómodos para introducir este tipo de información) o solamente uno (envío por correspondencia) .

Por el contrario, si analizamos nuestra tabla Cuentas, vemos que no estamos cumpliendo con la 1FN. El campo Cuenta, que hemos introducido con 20 dígitos, tiene distintas secciones: entidad, oficina, DC (Dígito de Control) y cuenta, que podrían ser campos independientes.

¿Qué hacer? En caso de duda, debemos aplicar siempre la norma, pues si cambiamos de criterio posteriormente, es mucho más fácil unir varios campos en uno que separar un campo en varios. Piense, por ajemplo, en una persona con apellidos compuestos incluidos en un solo campo; más adelante resultará imposible saber cuál es cada apellido. En el caso de la cuenta, tener separadas las distintas secciones nos puede facilitar más adelante, entre otras cosas, utilizar una fórmula para calcular el CCC (Código Cuenta Cliente) y verificar así si hemos introducido correctamente el número.

**2FN (Segunda Forma Normal): Dependencia funcional.**

Todos los campos de la tabla deben depender funcionalmente de la clave primaria completa y no de parte de ella. Esta norma solo podría incumplirse si tuviéramos una clave primaria compuesta, es decir, en el caso de que hubiésemos utilizado la combinación de varios campos para obtener una clave única.

Por ajemplo, una tabla de viviendas de una urbanización podría tener como clave principal una que estuviera compuesta por portal, escalera, piso y letra, identificando así, unívocamente, cada vivienda. El propietario de la vivienda depende funcionalmente de la clave completa; sin embargo, el presidente de la comunidad de vecinos dependería únicamente del portal, esto es, una parte de la clave, y, por tanto, no estaríamos cumpliendo la 2FN si se incluyese en la tabla de viviendas. Estaríamos duplicando información innecesariamente y cada vez que se cambiara de presidente, tendríamos que modificar el dato en todas las viviendas.

**3FN (Tercera Forma Normal): Dependencia funcional transitiva.**

No deben existir dependencias transitivas entre campos que no sean clave. Por ejemplo, en la tabla creada en la unidad didáctica anterior, Cuentas, se ha incluido un campo Oficina para identificar la sucursal bancaria en la que tenemos esa cuenta; sería muy interesante disponer del número de teléfono de la sucursal, pero resultaría incorrecto incluirlo en Ja misma tabla, pues el teléfono depende funcionalmente del campo Oficina, no de la clave que identifica a la cuenta. Lo correcto sería crear una nueva tabla para sucursales bancadas y relacionarla con la tabla Cuentas usando una columna de búsqueda. Observe que la segunda y la tercera forma son muy similares y lo que las diferencia es que en la segunda la dependencia es con parte de Ja clave y en la tercera, con campos que no son clave.

En resumen, con la aplicación de estas reglas lógicas se trata que los campos contengan la unidad de información más pequeña posible, añadiendo para ello los campos que sean precisos y que funcionalmente dependan siempre de la clave principal, añadiendo las tablas que sean necesarias cuando no ocurra así. Cuando empiece a trabajar con formularios, subformu/arios y consultas, irá entendiendo majar el objetivo de estas realas y, a medida que vaya diseñando tablas, se olvidará de las formas normales y pensará que únicamente se guía por el sentido común para conseguir un funcionamiento eficiente de Ja base de datos; seguramente será así, pero siempre, en caso de duda, aplique las formas normales, que están pensadas para eso.

* + - 1. **Diseño de campos**

A continuación vamos a ver el diseño de campos con Microsoft Access 2010. Para ello abra Access **(lnicio>Programas>Microsoft Office>Microsoft Access 2010);** en la pantalla inicial seleccione la opción **Base de datos en Blanco** y pulse en el botón **Crear** situado en la parte izquierda de la pantalla. En la pestaña de la cinta de opciones **Crear,** en el grupo de **Tablas.,** observaremos las distintas maneras de iniciar la creación de una tabla.

ªTabla =

.

'i

Listas

Diseño

tabla SharePoint •

*d e*

Tablas

En el grupo **Tablas** destaca el botón Listas de SharePoint, que despliega una galería de plantillas para crear listas en un entorno SharePoint. A partir de la versión 2007 de Office, Microsoft da mucha importancia a la integración entre Access y SharePoint, un entorno de colaboración en la web que requiere, además del propio servidor SharePoint, conocimientos específicos, por lo que no es objeto de tratamiento en el presente manual.

La primera opción, **Tabla,** abre una nueva tabla tal como lo acaba de hacer Access al abrir un archivo recién creado. La opción Plantillas de tabla nos proporciona plantillas definidas para crear una tabla.

**tabla.**

Para crear una tabla, la forma más común es pulsando el botón **Diseño de**

Al pulsar sobre ese botón, se abrirá una tabla en blanco en la vista de modo diseño, tal como se abriría si lo hiciéramos con una tabla existente solo que sin ningún campo aún. Como ya tenemos varias tablas creadas, lo que vamos a hacer es continuar con su diseño en vez de empezar con una nueva.

Podemos abrir una tabla existente en este tipo de vista de varias maneras distintas:

* + - * + En el panel de exploración seleccionamos la tabla Cuentas; pulsamos el botón derecho del ratón y, en el menú contextual que se muestra, elegimos **Vista Diseño.**

Otra forma de acceder a esta vista es hacer doble clic en el panel de exploración sobre el nombre de la tabla para que se abra en vista hoja de datos; seguidamente, en la cinta de opciones iremos a Inicio, grupo de opciones Vistas, y pulsaremos sobre el icono con el

siguiente dibujo . 11' :

Todavía existe otra opción para presentar la tabla en este tipo de vista. En la parte inferior derecha de la ventana existen opciones para el cambio de vista; la barra en la que se encuentran se llama -barra de estado» y, opcionalmente, puede estar oculta.

En cualquiera de los casos, se abrirá la vista de modo diseño de la tabla, que se presenta así:

.....,,,.,,,.,......, -

·- '" -,.;..¡ ...

---. -.

' ' ·'°""'""'"'"'º" *,,:::."::.,* """ *";;'.;";,;,•:'* \_..,,,'<::,,.."" "'"'"""' D:: :;"';'





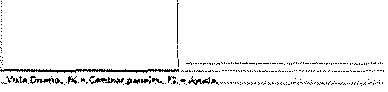
Vn'"'"""'"'''"'"""""''"'"'"'·..

....

,',",',".,".'.".'.".,','"'"'"''"'''"'"'"'"'"'"""''"""""''"'""

"- "'"''"""º





En la **Vista Diseño** de la tabla se observan varias secciones. En la parte superior tenemos tres columnas para Nombre del campo, Tipo de datos y Descripción, que conforman las propiedades esenciales de los campos. En la parte interior, contamos con una hqja de propiedades, las cuales variarán según el tipo de dato que hayamos elegido en la parte superior para ese campo. Ambas partes (propiedades esenciales y hoja de propiedades) son propiedades de los campos, solo que Nombre del campo y Tipo de datos existen en todos los campos, mientras que las de la parte inferior existirán o no y tendrán un rango de valores u otro

##### •

dependiendo del tipo de dato elegido. La Descripción no es una propiedad esencial y únicamente tiene la finalidad de ayudarnos a identificar el campo: es perfectamente prescindible si el nombre del campo es suficientemente descriptivo.

En función del tipo de dato que hayamos definido para cada campo, este tendrá un comportamiento distinto, mostrándose y permitiendo la entrada de datos de una manera específica.

Para diseñar la tabla, debemos ir definiendo para cada campo su nombre y tipo de datos en la sección superior y cambiando sus propiedades en la hoja de propiedades de la parte inferior para delimitar más el tipo de datos y decidir otras características sobre su comportamiento.

* + - 1. **Propiedades que definen el tipo de datos**

Cada tabla está compuesta por **campos.** Las propiedades de un campo describen las características y el comportamiento de los datos que se agregan a ese campo. El tipo de datos de un campo es la propiedad más importante porque determina el tipo de datos que el campo puede almacenar. En este punto se describen los tipos de datos y otras propiedades de los campos que se encuentran en Microsoft Access 2010, e incluye información adicional en una sección de referencia detallada de tipos de datos

Los tipos de datos pueden parecer confusos, por ejemplo, si un tipo de datos es Texto, puede almacenar datos que constan de caracteres de texto o numéricos. Pero un campo cuyo tipo de datos es Número solo puede almacenar datos numéricos. Por lo tanto, deberá conocer qué propiedades se usan con cada tipo de datos. El tipo de datos de un campo determina muchas otras cualidades importantes del mismo, entre otras:

* + - * + Los formatos que se pueden usar con el campo.
        + El tamaño máximo de un valor del campo.
        + Cómo se usa el campo en las expresiones.
        + Si el campo se puede indicar.

El tipo de datos del campo se puede predefinir o se seleccionará un tipo de datos según la manera en que se ha creado el nuevo campo. Por ejemplo, si se crea un campo desde la vista Hoja de datos y:

Se usa un campo existente que proviene de otra tabla, el tipo de datos ya está definido en la plantilla o en la otra tabla.

Se especifican datos en una columna (o campo) en blanco, Access 201O asigna un tipo de datos al campo en función de los valores que se especifiquen, o se puede asignar el tipo de datos y formato para el campo.

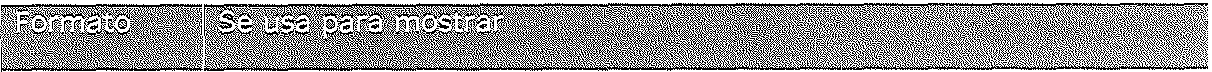
En la ficha Modificar campos, en el grupo Campos y columnas, haga clic en Agregar campos, Access 201O muestra una lista de tipos de datos que puede seleccionar.

Cuándo usar un determinado tipo de datos

Imagínese el tipo de datos de un campo como un coajunto de cualidades que se aplica a todos los valores incluidos en el campo. Por ajemplo, los valores almacenados en un campo Texto solo pueden contener letras, números y un conjunto limitado de caracteres de puntuación, y un campo Texto puede contener un máximo de 255 caracteres.

Los tipos de datos disponibles en Access 2010 son: Tipos básicos, Número, Fecha y hora, Sí/No e Inicio rápido. Las tablas siguientes muestran los formatos disponibles para cada tipo de datos y explican el efecto de la opción de formato.

TIPOS BÁSICOS



Texto

Valores alfanuméricos cortos, como un apellido o una dirección

Número

Valores numéricos, como distancias. Tenga en cuenta que existe un tipo de datos independiente para la moneda.

Moneda Valores monetarios.

SI/No Valores Sí y No, y campos que contienen uno de dos valores posibles Fecha/hora Valores de fecha y hora para los años 100 a 9999.

Texto

enriquecido

Texto o combinación de texto y números a los que se puede aplicar formato mediante controles de color y de fuente

Campo

calculado

Resultado de un cálculo. El cálculo debe hacer referencia a otros campos de la misma tabla. Para crear el cálculo se usa el Generador de expresiones.

**Datos**

**adjuntos**

**Hipervínculo**

**Memo**

**Búsqueda**

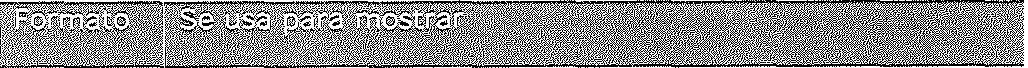
NÚMERO

Imágenes, archivos de hoja de cálculo, documentos, gráficos y otros tipos de archivos admitidos adjuntos a los registros de la base de datos, de forma similar a adjuntar archivos a los mensajes de correo electrónico.

Texto o combinación de texto y números almacenados como texto y usados como dirección de hipervfnculo.

Bloques de texto largos. Una utilidad típica de un campo Memo sería una descripción de producto detallada.

Muestra una lista de valores recuperada de una tabla o una consulta. o un conjunto de valores especificado en el momento en que se creó el campo. Se inicia el Asistente para búsquedas y se puede crear un campo Búsqueda. El tipo de datos de un campo Búsqueda es Texto o Número, según las opciones seleccionadas en el asistente.



**General**

Números sin formato adicional, exactamente como han sido almacenados.

**Moneda** Valores monetarios en general

**Euro** Valores monetarios en general almacenados en formato UE.

**Fijo** Datos numéricos

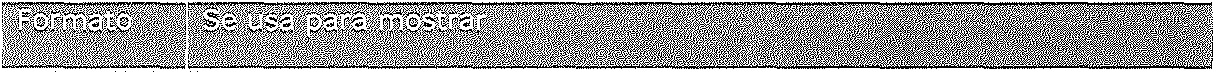
**Estándar** Datos numéricos que contienen decimales

**Porcentaje** Porcentajes

**Científico** Cálculos

FECHA Y HORA

**Fecha corta**



Muestra la fecha con un formato abreviado. Depende de la configuración regional de

**Fecha mediana**

**Fecha larga**

**Hora**

a.m/p.m

**Hora**

fecha y hora. Por ejemplo, 14/3/2009 para España.

Muestra la fecha con un formato mediano. Por ajemplo, 3-Abr-09 para España.

Muestra la fecha con un formato largo. Depende de la configuración regional de fecha y

hora. Por ajemplo, Miércoles, 14 de marzo, 2009 para España

Muestra la hora con un formato de 12 horas que responderá a los cambios en la configuración regional de fecha y hora.

Muestra la fecha seguida por a.m./p.m.

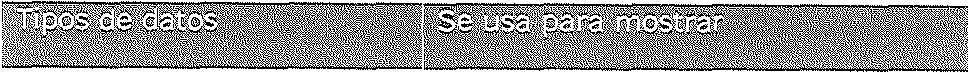
mediana

Hora 24

horas

Muestra la hora con un formato de 24 horas que responderá a los cambios en la configuración regional de fecha y hora.

SÍ/NO



Casilla de verificación

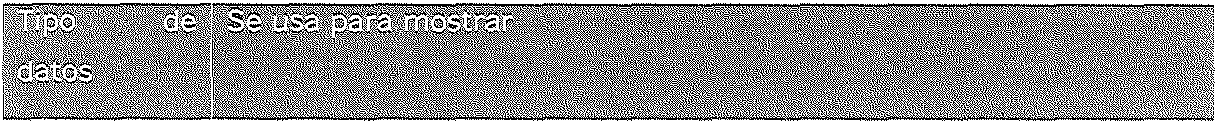
Una casilla de verificación

Sí/No Opciones Sf o no.

Verdadero/Falso Opciones Verdadero o Falso

Activado/Desactivado Opciones de activado o desactivado

I N I CIO RÁP I DO



Dirección

Campos para la dirección postal completa

Teléfono Campos para el teléfono particular, teléfono móvil y teléfono de trabajo

Prioridad Un cuadro desplegable con las siguientes opciones de prioridad: Baja. Media, Alta

Estado

Un cuadro desplegable con las siguientes opciones: Sin iniciar, En curso, Completada,

Cancelada

Etiquetas Muestra hasta tres etiquetas.

1.2. Configuración de pantallas de entrada de datos

El diseño de la entrada consiste en el desarrollo de especificaciones y procedimientos para la preparación de datos, la realización de los procesos necesarios para poner los datos de transacción en una forma utilizable para su procesamiento.

Un buen diseño de los formatos y las pantallas de entrada debe satisfacer los objetivos de eficacia, precisión, facilidad de uso, consistencia, sencillez y atracción.

La eficacia significa que las formas y las pantallas de entrada satisfagan propósitos específicos del sistema de información de la administración, mientras

que la precisión se refiere a un diseño tal que asegure una realización satisfactoria. La facilidad de uso implica que tanto las formas como las pantallas serán explícitas y no requerirán de tiempo adicional para descifrarse. La consistencia, en este caso, significa que las formas y las pantallas ordenen los datos de manera similar de una aplicación a otra, mientras que la sencillez se refiere a mantener en un mínimo los elementos indispensables que centren la atención del usuario. La atracción implica que el usuario disfrutará del uso o tránsito a través de las formas y las pantallas cuyos diseños les sean más atractivos.

**Diseños de Salidas**

Lo más usual para un diseñador es que el usuario juzgue el sistema por como esta presentado en funcionamiento o por el resultado que este genere al momento de ser utilizado, si la salida no es de calidad, se pueden convencer de que todo el sistema es tan innecesario que eviten su utilización y, por lo tanto, posiblemente ocasionen errores y que el sistema falle.

La salida es la información que reciben los usuarios del sistema de información. Las salidas pueden tomar distintas formas: los reportes impresos tradicionales y salidas en formatos, tales como pantallas en monitor, micro formas y salidas de audio. Los usuarios confían en las salidas para la realización de sus tareas; y con frecuencia, juzgan el mérito del sistema exclusivamente por sus salidas.

1. **Herramientas de creación de mantenimientos**

El mantenimiento de las bases de datos de **Access** es muy sencillo, basta con compactar y reparar, y hacer copias de seguridad; pero debemos hacerlo con frecuencia.

Con el uso, las bases de datos de Access tienden a crecer en tamaño y a fragmentar la información. También pueden estropearse distintos objetos de nuestra base de datos por cerrar Windows sin cerrar previamente Access, por cortes de luz, etc. Para solucionar estos problemas tenemos la herramienta **Compactar y reparar base de datos.** Además también se nos da la opción de cifrar con una contraseña la base de datos.

Llegamos a ambas herramientas pulsando sobre el Archivo y eligiendo la opción **1nformación.**

[

**Ccmpactar y reparar**

base de dates

Com pactar y repa ra r

**Use Compactar y· reparar para evitar y corregir problemas con les**

**archives de bases de dates.**

,b ::r:

ljl -"--

**Cifrar con contraseña**

Cifra r con contraseiía

**Use una contraseña para re\_o::tringir el acceso a la base de datc-s. Les archives que usan e! formato de archive Microsoft Access 2007 e posterior estcl n cifrados.**

Al compactar y reparar, Access recupera el espacio ocupado por oQjetos ya borrados, reorganiza la información interna e intenta reparar tablas, formularios, informes y módulos dañados. No realiza directamente los cambios en el archivo original, sino que hace una copia sobre la que efectúa las distintas operaciones y, una vez finalizadas, sustituye el archivo original por la copia; por ese motivo, al compactar y reparar, siempre se reinicia la aplicación.

Es conveniente llevar a cabo esta operación con frecuencia, pues conseguiremos que nuestra aplicación funcione de manera más ágil y evitaremos problemas.

Entre las opciones de Access, que veremos más adelante, existe una para que compactar y reparar se ajecute siempre al cerrar la base de datos.

* 1. **Configuración de herramientas para crear tablas de datos**

Anteriormente vimos como el servidor MySQL nos da la opción de realizar operaciones de creación, modificación y borrado de tablas. Ahora vamos a centrarnos en ver la creación de tablas con Access 201O.

Para crear una tabla nueva con Access 201O sólo tenemos que seguir los siguientes pasos:

1. En la ficha Archivo, haga clic en Nuevo y, a continuación, haga clic en Base de datos en blanco.

**Plantili disponibles**

**Inicio**

\_::, ::

--

\_.-:-T"·:.,

*v:'1*

*--;c:>J*

\_\_,

**Base de datos en Base de datos**



**blanco web en blanco**

**Plantillas de Office.com**

**Plantillas Plantilla::; de Mis plantillas reciente:5 ejemplo**

::. \_::, ::. ::.



1 fl ;-;::b-11

-:[c. ,

-··'

i ***'t ·-***

-! B:

'::;;,%

**Activos Contactos Problemas y** ONG' **Proyectos**

**tareas**

1. En el cuadro Nombre de archivo, escriba un nombre de archivo para la nueva base de datos.
2. Para guardar la base de datos en otra ubicación, haga clic en el icono de carpeta.
3. Haga clic en Crear.

Se abre la nueva base de datos y se crea una nueva tabla denominada Tabla1, la cual se abre en la vista Hoja de datos.

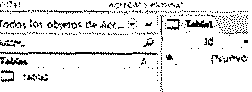
il:'m *;,v.,* "" ....,,.,\_,.,., ...,,,....,...,.,.,.,,,,""' ·•"'' C-o« '""''

¡;, : A....B......1..2.. ,""'""

: T- ,:'. \_ -· \_f,; ..¡,

::. ,...,..,,,,.,,.





También podremos crear una tabla en una base de datos existente con Access 201O, para ello debe seguir los siguientes pasos:

* 1. Haga clic en la pestaña Archivo, seleccione Abrir y, a continuación, realice una de las siguientes acciones:
     + Si la base de datos que desea usar aparece en Bases de datos recientes, haga clic en la base de datos en esa lista.
     + Si la base de datos no aparece en Bases de datos recientes, en Abrir documento, haga clic en la opción que desee.
  2. En el cuadro de diálogo Abrir, seleccione la base de datos que desee y

haga clic en Abrir.

* 1. En la ficha Crear, en el grupo Tablas, haga clic en Tabla.

Se inserta una nueva tabla en la base de datos y la tabla se abre en la vista Hoja de datos.

## •

**RECUERDA**

* Básicamente, el diseño de una tabla consiste en definir los distintos campos y sus propiedades, pero a esto debemos llegar después de un análisis del problema y aplicando unas reglas lógicas, que proceden del modelo relacional, para conseguir un rendimiento eficiente, Jo que se conoce como **«normalización».**
* La **normalización** es el proceso de aplicación al diseño de una base de datos de un conjunto de reglas lógicas, ya definidas, que tienen como objetivo garantizar la integridad de Jos datos, evitar redundancias y conseguir un funcionamiento eficiente.
* Durante el desarrollo del modelo relacional, el propio Codd definió un coajunto de reglas que se debían aplicar en el diseño lógico de una base de datos, a las que llamó «formas normales».
* Las tablas deben tener una clave primaria que identificará de forma inequívoca y exclusiva cada registro. Esta clave no puede contener valores nulos, es decir, no puede quedar sin valor, pues así se garantizará que todos los registros sean únicos. Access se ocupa de sugerir siempre una clave primaria, aunque permite no ponerla o que usemos una creada por nosotros.
* Todos los campos de la tabla deben depender funcionalmente de la clave primaria completa y no de parte de ella. Esta norma solo podría incumplirse si tuviéramos una clave primaria compuesta, es decir, en el caso de que hubiésemos utilizado la combinación de varios campos para obtener una clave única.
* No deben existir dependencias transitivas entre campos que no sean clave.
* Cada tabla está compuesta por **campos.** Las propiedades de un campo describen las características y el comportamiento de los datos que se agregan a ese campo. El tipo de datos de un campo es la propiedad más importante porque determina el tipo de datos que el campo puede almacenar.
* El diseño de la entrada consiste en el desarrollo de especificaciones y procedimientos para la preparación de datos, la realización de Jos procesos necesarios para poner los datos de transacción en una forma

utilizable para su procesamiento.

* La salida es la información que reciben los usuarios del sistema de información. Las salidas pueden tomar distintas formas: los reportes impresos tradicionales y salidas en formatos, tales como pantallas en monitor, micro formas y salidas de audio. Los usuarios confían en las salidas para la realización de sus tareas; y con frecuencia, juzgan el mérito del sistema exclusivamente por sus salidas.
* El mantenimiento de las bases de datos de **Access** es muy sencillo, basta con compactar y reparar, y hacer copias de seguridad; pero debemos hacerlo con frecuencia .

## •

**Preguntas de Autoevaluación**

1. **Completa la siguiente afirmación con la opción correcta:**

"La ----------- es el proceso de aplicación al diseño de una base de datos de un coajunto de reglas lógicas, ya definidas, que tienen como oQjetivo garantizar la integridad de los datos, evitar redundancias y conseguir un funcionamiento eficiente".

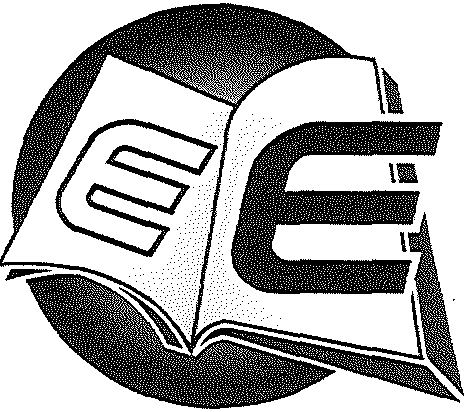
1. Normalización.
2. Desnormalización.
3. Atomicidad.
4. **¿Cuál es la segunda regla que propone Cood para que se aplique al diseño de una base de datos?**
5. Atomicidad.
6. Dependencia funcional
7. Dependencia funcional transitiva.
8. **¿Cuál de las siguientes afirmaciones es incorrecta siguiendo las reglas lógicas que nos propone Cood?**
9. Las tablas deben tener una clave primaria que identificará de forma inequívoca y exclusiva cada registro.
10. Todos los campos de la tabla deben depender funcionalmente de la clave primaria completa y no de parte de ella.
11. Deben existir dependencias transitivas entre campos que no sean clave.
12. **Confirma si la siguiente afirmación es verdadera o falsa:**

*"El diseño de la entrada consiste en el desarrollo de especificaciones y procedimientos para la preparación de datos, la realización de los procesos necesarios para poner los datos de transacción en una forma utilizable para su procesamiento".*

1. Verdadero.
2. Falso.
3. **¿Qué opción debemos elegir de Access para llegar a las herramientas de mantenimiento de la base de datos?**
4. Inicio.
5. 1 nformación.
6. Administración.

Operaciones de consulta

UD6



UF1888 Opecraciones de Mantenimiento y Consulta de Datos

* 1. **Herramientas de búsqueda ágiles para el usuario**

A continuación vamos a ver las funcionalidades de los protocolos de consulta, la configuración de pantallas de búsqueda y el establecimiento de criterios de búsqueda para cadenas o números.

* + 1. **Definición de las funcionalidades de los protocolos de consulta**

Una **consulta** es un conjunto de instrucciones que se pueden usar para trabajar con datos. Se ajecuta para que se lleven a cabo estas instrucciones. Además de devolver resultados que se pueden almacenar, agrupar o filtrar, una consulta también puede crear, copiar, eliminar o cambiar datos. Para realizar o ajecutar consultas en tablas de bases de datos estructuradas, son necesarios unos valores o criterios de consulta.

Un criterio de consulta es una regla para identificar los registros que se desean incluir en el resultado de una consulta. No todas las consultas deben incluir criterios, pero si no le interesa ver todos los registros que están almacenados en el origen de registros subyacente, deberá agregar criterios a una consulta cuando la diseñe. Un criterio es similar a una fórmula (es una cadena que puede consistir en referencias de campos). Los criterios de consulta también se conocen como expresiones en Microsoft Access.

Los criterios pueden ser muy distintos unos de otros, dependiendo del tipo de datos del campo en el que se apliquen y de Jos requisitos en cada caso. Algunos criterios son sencillos y usan operadores básicos y constantes. Otros son complajos y usan funciones y operadores especiales, e incluyen referencias de campo.

* + - 1. **Tipos de consulta**

Access 2010 nos ofrece Jos siguientes tipos de consulta:

### •

* + - * + **Consultas de selección:** obtienen los datos de una o más tablas y muestran los resultados en una hoja de datos en la que podemos actualizar los registros (con algunas restricciones). Este tipo de consulta es la más habitual y se usan también para agrupar los registros y calcular sumas, cuentas, promedios y otros tipos de totales.
        + **Consultas de acción:** realizan cambios o desplazamientos de muchos registros en una sola operación. Hay cuatro tipos de consultas de acción:

Consulta de eliminación: suprime un grupo de registros de una o más tablas. Siempre se eliminan registros enteros, no sólo los campos seleccionados dentro de los registros.

Consulta de actualización: realiza cambios globales en un grupo de registros de una o más tablas. Modificamos los datos de las tablas existentes.

Consulta de datos anexados: agrega un grupo de registros de una o más tablas al final de una o más tablas.

Consulta de creación de tabla: crea una tabla nueva a partir de la totalidad o bien una parte de los datos de una o más tablas. Son útiles para crear una tabla que se desee exportar a otra base de datos de Microsoft Access, o una tabla histórica que contenga registros antiguos.

* + - * + **Consulta de tabla de referencias cruzadas:** se utilizan para calcular y reestructurar datos, de manera que su análisis resulte más sencillo. Calculan una suma, una media, un recuento u otro tipo de totales de datos y se agrupan en dos tipos de información; uno vertical, en el lado izquierdo de la hqja de datos, y otro horizontal, en la parte superior.
        + **Consulta de parámetros:** cuando se ajecuta, muestra un cuadro de diálogo que solicita información para recuperar registros o un valor que desea insertar en un campo. Podrá diseñar la consulta para que solicite más de un dato, por ajemplo, entre dos fechas. De esta forma, Access podrá recuperar todos los registros que se encuentren en ese período de tiempo e imprimir el informe correspondiente.
* **Consulta SQL:** está creada con instrucciones SQL (Structure Query Language) para consultar, actualizar y administrar bases de datos relacionales. Cuando se crea una consulta de la Vista Diseño, Access construye, en segundo plano, las instrucciones SQL equivalentes. Las consultas SQL específicas son:
  + Consulta de unión: combina dos o más tablas o consultas en un solo campo. Para crear una consulta de unión, seleccionamos la sección Consultas en el Panel de navegación, hacemos clic sobre la ficha Crear de la Cinta de opciones y, a continuación, hacemos clic sobre el botón Diseño de consulta, del grupo Consultas. Aparecerá el cuadro de diálogo Mostrar tabla. Lo cerramos sin agregar ninguna tabla, y ajecutamos el comando Unión, del grupo Tipo de consulta de la ficha Diseño, en la Cinta de opciones. Por último, escribimos las sentencias SQL en la ventana de consulta.
  + Consulta paso a través: se utiliza para crear, eliminar o modificar tablas enviando comandos directamente a las bases de datos ODBC.
  + Consulta de definición de datos: crea, elimina o modifica tablas, o crea índices en una tabla de base de datos.
  + Subconsulta: es una instrucción SQL SELECT dentro de otra consulta de selección o de acción. Podemos introducir este tipo de instrucciones dentro de la fila Campo de la cuadrícula de Diseño de la consulta para definir un campo nuevo, o bien en la fila Criterios para definir los criterios de un campo.
    - 1. **Consultar datos con MySQl**

Vamos a hacer ahora una consulta de los datos que acabamos de insertar en la base de datos. Las consultas se hacen con executeQuery() y nos devolverán un ResultSet.

El ResultSet de alguna forma representa una conexión hacia los datos. En el ResultSet NO están todavía los datos. Según se los vayamos pidiendo, los irá trayendo de base de datos. Esto quiere decir que si una consulta devuelve muchos resultados, no se nos va a llenar la memoria por el hecho de hacer la consulta.

Para traer el primer resultado, debemos llamar el método next() del ResulSet. Para el siguiente otro next() y así sucesivamente hasta que next() devuelva false, indicando que ya no quedaban datos. El código puede ser así

ResultSet rs = st .executeQuery ( "SELECT \* FROM contacto") ; while ( rs.next () )

{

System .out .pr intln ( "nombre= "+rs. getObject ( "nombre") +

**rr , apel.lidos=''+rs . getObject ( "apellidosrr** +

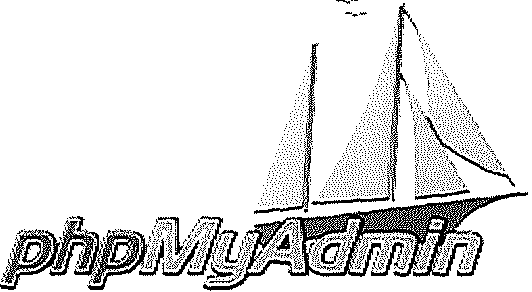
,,*/* **telef ono=''+rs. getObject ( ''te.lef ono'') )** *¡*

}

rs. clase () ;

* + 1. **Configuración de pantallas de búsqueda**

A continuación vamos a explicar como buscar a través de una base de datos usando **phpMyAdmin.** En este punto usaremos como ajemplo una base de datos con una tabla.



Empecemos por ver la pantalla principal de phpMyAdmin. En el lado izquierdo es donde encontrarás enlistada tu base de datos, abajo de esta base de datos estará cualquier tabla que contenga. En el lado derecho, en la gran pantalla blanca, se llevara acabo la acción .

# •

*...,'ii!W-IYJ* !}) S<> ....l<.lor: lo(allo<>St @ 8....o <lo <fo\Oli: C<>l!Trfor

lifill!j]IT!l][9 orn•ettor" SOL ..:·am 1 Go11<>1"' un cn10au11,, 1:ii!b:!!:""'" lhnort.lr 01..,11..dor goorodo110 ;:'¡!?rMlo IM t;En.11;., •.,

Tabla A.;<:l<in Roglm.,..1 Tie• Co1•JOmlon10 Tam1160 Roolduo dt>p11r.u

,\_a.•\_\_•\_dd.11!.º' -··· o wp\_o:<>mmentm a *r,.;* ;; r· MylSAM url3..senoral\_ei "·º '°'

" "

(1 ----."'.'.J o wp\_.:om1non11 " " "' }< "' " "'""" ute..cen•r<il\_c1

(on1ro1or

(14)

o wp\_c-ooklo\_compll...neo\_log " " "

te "' "

MylSAM 11n1\_dl&h\_c1

'·' }:ll

:.1K1>

!!ll "l>- • o wp\_«Oklo\_compll11neo\_vhl1or " F ·t; X • """"" IA11nl\_ dltll\_ci '·º "''

l··lil"l>--. --

o w.,\_p,\_,.p,.-.iet.> "' " " "' X ... Myl&o.M uffl\_generar\_ei "· *n*

ll!l "l>.=-nl• o wp\_llnb " :?-' '111" ;x MylSAM utta\_gonornl\_e1 l.O !(ll

llll wp.,.cool<lo\_..M

l!ll"'ll.COOlílO--or

o wp\_opllon& " " "' "' ;x m ""'"" l!tll\_gon•ml\_er I.*1* l«l

·--

o wp\_1erm\_t.lJO:onomy

o wp\_,.,...,,

" " " ?-" " ·º·º 1(11 :oo ¡¡y,,.,

!lll wpJ<!t'!ll o wp\_tem1\_rot.nl&nohl l"' !ti1 r:i " :?< 111 ;x ll .,.... ot1El\_gonorii1\_e1 -= ni

;x m MylSAM lll'S\_goMml\_el

o wp\_...." " " " ?-' " " "''""" ur1El\_genoml\_cí •.lK!I

"

llll "l>JorN

o " " " ¡.; 111 " MylSAM url3\_sonerol\_ei ¡l.Z l

l!!l"l'J-

l!!l"'l>. 1

1111 ""'-,,\_.

., .:s ¡;¡ "' "

MylSAM utfl\_goner<>l\_cl <.Z !ltl

!11""1'.)Cll'll...l""""',..,...,i°' o wp\_in:•rmetol " ¡;¡ !-' ti! " tt MylSAM "l1El\_genoral\_ci U.7 l'll

o wp\_xmt\_st1&m,.p\_generator 0!1El.Jlort<lral\_ei ,o !(E)

" " " "' " ' "'"""'

14 t<>l>l<>(t\ Nilme10 do fil<>• OM MylSAM l<11rn1\_..w•dkt1\_cr n.,D u

L M"rc;<r1 doJ..o *I* De<>m<rc.1f lodo / M<rcat I•" t•bl•> con t«id"o a dour.r

,p:;;;·¡oo1-.m. -.;· "ii: 2:0.ilú\·;;; d2:·!



".:J] C1""' n""" "'blo ..nl.> h..... <101<>11 01111 1..r---------------------------

Númol'Q d' camino•:;

En el panel del lado izquierdo, haz clic en la base de datos que quieres buscar. Con esto se mostrarán los detalles de la base de datos en la pantalla del lado derecho. Hasta arriba de esa pantalla haz click en "Buscar", esto abrirá una ventana de búsqueda. En el recuadro en blanco, debes ingresar las palabras o valores que quieres buscar en la base de datos. Puedes refinar tu búsqueda de distintas maneras. Un método que puedes usar es haciendo click en el botón de radio que limita el alcance de tu búsqueda, tal como todas las palabras, o la frese exacta, etc. Sí tienes mas de una tabla puedes limitar tu búsqueda a tablas específicas, usando la lista de desplazamiento. Cuando estés listo haz click en "1r" para empezar la búsqueda.

"""·----- ----- · ----·----·--------------·--·------------- ---

*php]Wgf)i!JYJ:JJj;tJ W* s MJon Jo<:.1lh""1 B.,. d& do19•: comr. or

@Il [g) d!Ewucrur.i . SOL ;:·B"'""' ¡@iGUQt run,.com;1rlt11 ¡jlbExeartM Tu\JhuponM "1"'m,.,..1l "dor gopnr.,d<>nM ;;l)P1Ml$ i01< :¡:;;r.11r,,;o.ir

l}uoea1 \*"l,1lh'l> <l<> da11><----------------------------------;



Polobrn(•) ovalor(<l'!>) a bucar (comodín:"%"):

co1111..for (14)

·--­

rm ""'...-"

·!lll "'P-.-..-•

- -

!lll ""'..,oooklo--pkir>Oo\_IOg

llli ""'.,ooo<lo.,ooonpl""""-""'ltor

Jlll wp-'"""fllollci<i"""'

Encontrado:

En la() tabla();

0 •I m"M una da o5t• p1>l•bras'

O TodM los palabrn;1

O t.a frnu oxoet•

*y* como ••praoi n regulor(!l

wp\_commonlmGlll "."i

'.: i: mpllance\_log ..J



·- ­

§;; =m1·.:, : .:L...--·-

llll "'P•

lll""l'J""""

!lil "'PJ"""-1"'"""""Y ml "'P-"-

Oont"l daleampo:

La pantalla siguiente te mostrará si se obtuvo algún resultado, y si se encontró algo, en donde se encuentra. Ahora puedes hacer clic en "Navegar" al final de los resultados y se te mostraran. Puedes hacer cambios para a ese registro si así lo quieres.

* + 1. **Establecimiento criterios de búsqueda para cadenas o números**

En MySQL hay dos formas de buscar dentro de campos de texto esto se hace con las cláusulas **LIKE o MATCH AGAIN.** Ambos buscan subcadenas dentro de un campo de MySQL pero presentan una diferencia. A continuación explicaremos como funcionan cada uno de ellos para entenderlos con mayor claridad.

Si tienes una frase tal como "Optimización de las consultas clientes" en un campo de MySQL que se llama texto en la tabla ejemplo y aplicas la siguiente Quera clave = "ultas"

SELECT texto FROM ajemplo WHERE texto LIKE '%clave%'; la query te dará una coincidencia en la búsqueda debido a que la cláusula LIKE te busca una subcadena en una cadena y los comodines % le indican que antes de esta subcadena puede haber cualquier cosa y después de ella también, esta subcadena la encuentra en hermosa; esto es útil cuando buscas por ajemplo nombres en un campo de nombres que incluyen el apellido por ajemplo buscar todos los Juanes, Pedros, Rosas, etc.

Ahora con la cláusula **MATCH AGAIN.** La sintaxis va así: clave = "ultas" SELECT texto FROM ejemplo Where MATCH texto AGAIN 'clave' ;

Esto te dará una coincidencia debido a que encuentra la subcadena pero es un método menos efectivo ya que MATCH AGAIN ajecuta un algoritmo que explicaremos más adelante y que en el caso de búsquedas de una sola frase hace más lenta la búsqueda por la ajecución de este algoritmo en cosas simples.

Ahora bien, si se quiere hacer la búsqueda de una frase y también que te encuentre las coincidencias sin importar el número de palabras ni el orden de estas tenemos que seguir las siguientes indicaciones.

clave = "consultas clientes"

SELECT texto FROM ajemplo WHERE texto LIKE '%clave%' ;

El ejecutar este query no te devolverá coincidencia porque LIKE únicamente busca de manera secuencial y aunque las dos palabras están en la frase, éstas no se encuentran en orden por lo tanto LIKE no nos sirve para una búsqueda con clave de más de una palabra si estas no están seguidas.

Ahora lo vamos a explicar con MATCH AGAIN clave ;"consultas clientes"

SELECT texto FROM ajemplo WHERE MATCH texto AGAIN 'clave'; te devolverá coincidencia porque MATCH AGAIN ajecuta un algoritmo en el cual primero se buscan las palabras de una por una sin tomar el orden en el que se encuentran y va sumando puntos en caso de encontrarlas y después selecciona el orden de las palabras en la búsqueda y de igual manera suma puntos si se encuentran en el orden correcto y pone como coincidencias principales las que obtienen una suma mayor, es decir si la clave es consultas clientes, al estar las dos palabras la tomara como major coincidencia en relación con hermosa novia porque además de tener las palabras en el campo de búsqueda las tiene major ordenadas, tal y como va en los buscadores públicos como Yahoo o Google.

**RECUERDA**

* Una **consulta** es un coajunto de instrucciones que se pueden usar para trabajar con datos. Se ajecuta para que se lleven a cabo estas instrucciones. Además de devolver resultados que se pueden almacenar, agrupar o filtrar, una consulta también puede crear, copiar, eliminar o cambiar datos. Para realizar o ajecutar consultas en tablas de bases de datos estructuradas, son necesarios unos valores o criterios de consulta .
* Un criterio de consulta es una regla para identificar los registros que se desean incluir en el resultado de una consulta. No todas las consultas deben incluir criterios, pero si no le interesa ver todos los registros que están almacenados en el origen de registros subyacente, deberá agregar criterios a una consulta cuando la diseñe.
* Access 201O nos ofrece los siguientes tipos de consulta: de selección, de acción, de tabla de referencias cruzadas, de parámetros, consulta SQL.
* La configuración de pantallas de búsqueda se hace a través de

**phpMyAdmin.**

* En MySQL hay dos formas de buscar dentro de campos de texto esto se hace con las cláusulas **LIKE o MATCH AGAIN.** Ambos buscan subcadenas dentro de un campo de MySQL pero presentan una diferencia.

**Preguntas de Autoevaluación**

1. **Una consulta:**
2. Es un conjunto de instrucciones que se pueden usar para trabajar con datos.
3. Es un elemento autocontenido utilizado por el programa.
4. Son extractos de datos procedentes de tablas o consultas y que tienen como propósito principal el ser impresos en papel, aunque también pueden simplemente visualizarse en pantalla.
5. **Confirma si la siguiente afirmación es verdadera o falsa:**

*"Un criterio de consulta es una regla para identificar los registros que se desean incluir en el resultado de una consulta. No todas las consultas deben incluir criterios, pero si no le interesa ver todos los registros que están almacenados en el origen de registros subyacente, deberá agregar criterios a una consulta cuando la diseñe".*

1. Verdadero.
2. Falso.
3. **Las consultas de acción de Access:**
4. Obtienen los datos de una o más tablas y muestran los resultados en una hoja de datos en la que podemos actualizar los registros (con algunas restricciones) .
5. Se utilizan para calcular y reestructurar datos, de manera que su análisis resulte más sencillo.
6. Realizan cambios o desplazamientos de muchos registros en una sola operación.
7. **¿Qué programa podemos usar para la configuración de las pantallas de búsqueda?**
8. Open Office.
9. phpMyAdmin.

e) MySQL.

1. **Confirma si la siguiente afirmación es verdadera o falsa:**

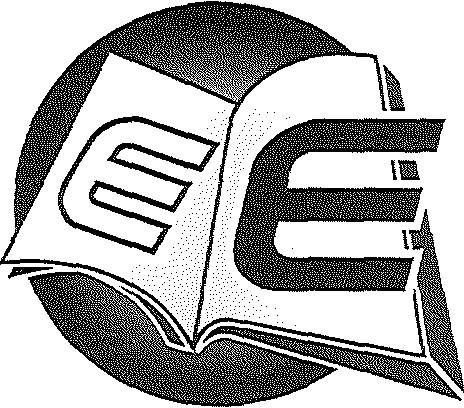
*"En MySQL hay dos formas de buscar dentro de campos de texto esto se hace con las cláusulas LIKE o MATCH AGAIN".*

1. Verdadero.
2. Falso.

**:ormularios** e **informes en sistemas ERP-**

UD7

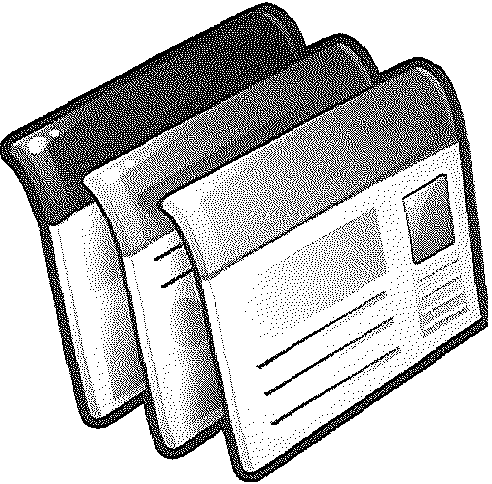
**CRM**



UF1888 Operac iones de Mantenimiento y Consulta de Datos

* 1. **Formularios**

Las bases de datos se usan para almacenar datos. Pero, ¿cómo se añaden los datos a la base de datos? Para eso se usan los **formularios.**



* + 1. **Características de los formularios e informes**
       1. **Características de los formularios**

Los formularios se conocen a veces como **"pantallas de entrada de datos".** Son las interfaces que se utilizan para trabajar con los datos y, a menudo, contienen botones de comando que ajecutan diversos comandos. Se puede crear una base de datos sin usar formularios, editando los datos de las hojas de las tablas. No obstante, casi todos los usuarios de bases de datos prefieren usar formularios para ver, escribir y editar datos en las tablas.

Los formularios proporcionan un formato fácil de utilizar para trabajar con los datos. Además, se les puede agregar elementos funcionales, como botones de comando. Puede programar los botones para determinar qué datos aparecen en el formulario, abrir otros formularios o informes, o realizar otras tareas diversas. Por ajemplo, podría crear un formulario denominado "Formulario de cliente" para trabajar con datos de clientes. El formulario de cliente podría tener un botón para abrir un formulario de pedido en el que se pudiese escribir un pedido nuevo del cliente.

•

Los formularios también permiten controlar la manera en que otros usuarios interactúan con los datos de la base de datos. Por ajemplo, puede crear un formulario que muestre únicamente ciertos campos y que permita la ajecución de determinadas operaciones solamente. Así, se favorece la protección de los datos y se facilita la entrada correcta de datos.

* + - 1. **Características de los informes**

Los informes sirven para resumir y presentar los datos de las tablas. Normalmente, un informe responde a una pregunta específica, como "¿Cuánto dinero se ha facturado por cliente este año?" o "¿En qué ciudades están nuestros clientes?" Cada informe se puede diseñar para presentar la información de la major manera posible.

Un informe se puede ejecutar en cualquier momento y siempre reflajará los datos actualizados de la base de datos. Los informes suelen tener un formato que permita imprimirlos, pero también se pueden consultar en la pantalla, exportar a otro programa o enviar por correo electrónico.

Los formularios y los informes se parecen en muchos aspectos, pero responden a propósitos distintos. La principal diferencia es que los formularios suelen usarse para introducir datos, mientras que los informes se usan para ver datos en pantalla o sobre papel. De cualquier forma, todo lo que podemos mostrar en pantalla con un informe podemos verlos también con un formulario. Los informes y formularios se basan en los datos de las tablas o consultas subyacentes, pero únicamente con un formulario podemos añadir datos o modificar datos originales.

* 1. **Búsqueda de herramientas estándar que faciliten la obtención de la información**

Generalmente la herramienta más utilizada para trabajar con formularios e informes es **Microsoft Access 2010.**

Microsoft Access 2010 es una herramienta de diseño e implementación de base de datos para aplicaciones que se puede usar para realizar un seguimiento de

## •

la información importante. Puede conservar los datos en el equipo o publicarlos en la Web, de forma que otros usuarios puedan usar Ja base de datos.

Access le permite desarrollar de forma fácil y rápida bases de datos relacionales que le ayudarán a administrar la información. Puede crear una base de datos que le ayude a realizar un seguimiento prácticamente de cualquier información, como un inventario, contactos profesionales o procesos de negocio. Incluye plantillas que puede utilizar directamente para realizar un seguimiento de diversos datos, lo que facilita la tarea incluso para un principiante.

Esta nueva versión dispone de majoras en la interfaz y funciones interactivas que permiten un tratamiento sencillo de los datos. No es necesario tener conocimientos avanzados sobre bases de datos, pues gracias a los asistentes y aplicaciones predefinidas, será una tarea rápida y sencilla. Es posible recopilar la información en formularios por correo electrónico o bien importarla desde aplicaciones externas. La capacidad de compartir información con las listas de la tecnología Microsoft Windows SharePointServices permite realizar auditorías y copias de seguridad periódicas desde la interfaz de usuario de Access 201O.

Algunas majoras de Microsoft Access 2010 son:

* + - Herramientas de creación de ot¡jetos más eficaces.
    - Operaciones de agrupación y ordenación simplificadas en los informes.
    - Diseños de controles majorados para facilitar la creación de formularios e informes optimizados.
    - Nuevos tipos de datos y controles.
    - Campos calculados y multivalor.
    - Campos de datos acjjuntos de archivos.
    - Calendario para seleccionar fechas.
    - Major presentación de datos.
    - Herramientas majoradas de ordenación y filtrado.
    - Seguridad majorada.
    - Herramientas de corrección majoradas.
    - Majores métodos de solución de problemas.
  1. **Exportación de la información a otros sistemas, por ejemplo, Excel**

Más adelante explicaremos la forma de crear un formulario y un informe a través de las herramientas que nos aporta Microsoft Access 201O, pero ahora nos vamos a centrar en explicar cómo exportar la información desde Access a otros sistemas, por ejemplo, Excel.

Mediante el **Asistente para exportación,** puede exportar datos de una base de datos de Access en un formato de archivo que Microsoft Excel puede leer.

Cuando exporta datos a Excel, Access crea una copia de los datos seleccionados y, a continuación, almacena los datos copiados en un archivo que se puede abrir en Excel. Si copia datos de Access a Excel con frecuencia, puede guardar los detalles de una operación de exportación para usos futuros e incluso programar la operación de exportación para que se ejecute automáticamente en intervalos establecidos.

* + 1. **Prepararse para la operación de exportación**

Antes de realizar un procedimiento de exportación, es una buena idea revisar los datos que desea exportar para asegurarse de que no contienen indicadores de error o valores de error. Si hay algún error, intente resolverlo antes de exportar los datos a Excel. De lo contrario, se pueden producir problemas durante la operación de exportación y pueden aparecer valores nulos en las celdas de la hoja de cálculo de Excel.

Si el objeto de origen es una tabla o una consulta, decida si desea exportar los datos con formato o sin él. Esta decisión afecta a dos aspectos del libro resultante: la cantidad de datos que se exportan y el formato de presentación de los datos. La tabla siguiente describe el resultado de exportar datos con y sin formato.

# •



**La configuración de la propiedad**

.9

"E'

2

e

*¡¡;*

.9

E"'

.E

e

o

<.>

Tabla o

**consulta**

**\*Los formularios y los informes no se pueden**

**exportar sin**

**formato.**

**Tabla, consulta, formulario o informe**

Se exportan todos los campos y registros del objeto **subyacente.**

Solo se exportan campos y **registros que se muestran en la vista o el oqjeto actual. No se exportan ros registros filtrados, las columnas**

**ocultas en una hoja de datos ni ros campos no mostrados en un formulario o informe.**

**Formato se omite durante esta**

**operación.**

**Para los campos de búsqueda, se exportan solo ros valores del identificador de búsqueda.**

**Para campos de hipervfnculo, el contenido se exporta como una columna de texto que muestra los vínculos en el formato textoParaMostrar#dirección#.**

**El Asistente respeta la configuración de la propiedad Formato.**

**Para campos de búsqueda, se exportan los valores de búsqueda.**

**Para campos de hipervrnculo, ros**

**valores se exportan como hipervfnculos.**

**Para campos de texto enriquecido, se exporta el texto pero no el formato.**

El)ja el libro de destino y el formato de archivo. Tenga en cuenta que los informes solo se pueden exportar en el antiguo formato de archivo \* .xls, no en el nuevo \* .xlsx.

Durante la operación de exportación, Access le pide que especifique el nombre del libro de destino. En la tabla siguiente se resume cuándo se crea un libro (si no existe aún) y cuándo se sobrescribe (si ya existe) .

•

'"¡('

"

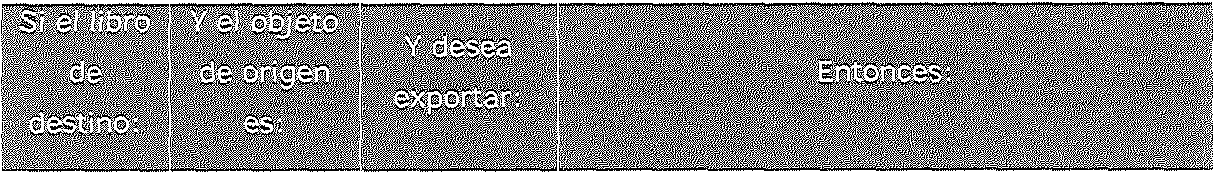
"o

**consulta,**

**formulario o**

**Los datos con**

**formato o sin El libro se crea durante la operación de exportación.**



Una tabla,

**formato**

z **informe**

**El libro no se sobrescribe.- se agrega una nueva hoja**

**de cálculo al libro y recibe el nombre del objeto del que**

·"";¡'

.",

>

Una tabla o

**consulta**

Los datos, **se exportan los datos. Si ya existe una hoja de cálculo pero no el con ese nombre en el libro, Access le pide que formato reemplace el contenido de la hqja de cálculo**

**correspondiente o especifique otro nombre para la**

**nueva hoja.**

**El libro se sobrescribe con los datos exportados. Se**

Una tabla, **eliminan todas las hojas de cálculo existentes y se crea**

';"('

"

".,

**consulta,**

**formulario o**

Los datos,

**incluido e\**

**formato**

**una nueva hoja de cálculo con el mismo nombre que el**

objeto exportado. Los datos de la hoja de cálculo de

> **informe Excel heredan la configuración de formato del objeto**

**de origen.**

Los datos siempre se agregan en una hoja de cálculo nueva. No puede anexar los datos en una hqja de cálculo existente o en un rango con nombre.

* + 1. **Ejecutar la operación de exportación**
       1. Si el libro de destino de Excel está abierto, ciérrelo antes de continuar.
       2. En el panel de navegación de la base de datos de origen, seleccione el objeto que desea exportar.
       3. En el grupo Exportar de la ficha Datos externos, haga clic en Excel.
       4. En el cuadro de diálogo Exportar a una hoja de cálculo de Excel, compruebe el nombre de archivo propuesto para el libro de Excel (Access usa el nombre del objeto de origen) . Si lo desea, puede modificar el nombre de archivo.

desee.

* + - 1. En el cuadro Formato de archivo, seleccione el formato de archivo que
      2. Si está exportando una tabla o consulta y desea exportar datos con formato, seleccione Exportar datos con formato y diseño. Para obtener más información, vea la sección sobre cómo prepararse para la operación de exportación.
      3. Para ver el libro de destino de Excel cuando se complete la operación de exportación, active la casilla Abrir el archivo de destino al finalizar la operación de exportación.
      4. Si el oQjeto de origen está abierto y selecciona uno o varios registros en la vista antes de iniciar la operación de exportación, puede seleccionar Exportar solo los registros seleccionados. Para exportar todos los registros presentados en la vista, daje sin activar esta casilla.
      5. Haga clic en Aceptar.

Si se produce un error en la operación de exportación, Access muestra un mensaje en el que se describe la causa del error. De lo contrario, Access exporta los datos y, según su selección en el paso 7, abre el libro de destino en Excel. A continuación, Access muestra un cuadro de diálogo en el que puede crear una especificación con la información de la operación de exportación.

1. **Arquitecturas de informes, elementos de informes**

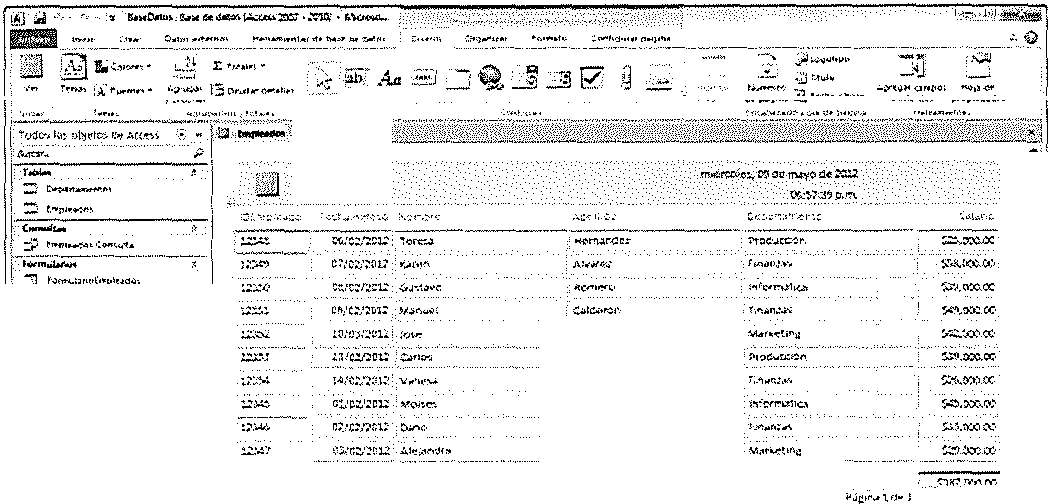
Los informes ofrecen el major método para imprimir la información y distribuirla, y nos dan un gran control sobre el diseño general. Gracias a los informes podemos controlar fácilmente los estilos y tamaños de los tipos de letra, efectuar de forma sencilla los cálculos con los datos subyacentes, dar formato a los datos para que se ajusten a formularios ya diseñados e impresos, incluir gráficos, imágenes y otros elementos, o agrupar y organizar los datos para que resulte más fácil entender el informe.

En Access hay cuatro tipos de informes básicos: tabulares, en columnas, para gráficos y para etiquetas. Los informes tabulares muestran los datos en filas y columnas, y son parecidos a las tablas. Sin embargo, los informes tabulares se diferencian de las tablas en que agrupan sus datos según los valores de uno o más

## •

campos. Los informes tabulares pueden, además, contener otros elementos, como totales, fechas y subtotales. Se suelen usar para calcular y mostrar los subtotales de campos numéricos de grupos del informe.

Los informes distribuidos en columnas muestran los datos en un formato vertical, con uno o más registros por página. Estos informes muestran los datos de una forma muy parecida a la de los formularios de introducción de datos, pero se usan para ver e imprimir la información, no para introducir datos.



**El:npl(;>ados**

'""'"",.,,

'"'",,.:;,



**Elementos del informe**

En Access, el diseño de los informes se divide en secciones. En una base de datos cliente, se puede ver el informe en vista Diseño para mirar sus secciones. En vista Presentación, las secciones no son tan obvias, pero siguen ahí y se pueden seleccionar mediante el uso de la lista desplegable en el grupo Selección de la ficha Formato. Para crear informes útiles, debe comprender cómo funciona cada sección. Por ajemplo, la sección en la que eligió colocar un control calculado determina cómo calcula Access los resultados. En la lista siguiente se muestra un resumen de los tipos de sección y sus usos:

## •

* + **Encabezado del informe.** Esta sección se imprime una vez al principio del informe. El encabezado del informe se usa para ofrecer información que normalmente aparecería en una página de portada, como un logotipo o un título y una fecha. Cuando se coloca un control calculado que usa la función de agregado de suma en el encabezado del informe, el resultado de la suma se calcula para todo el informe. El encabezado del informe se imprime antes del encabezado de página.
  + **Encabezado de página.** Esta sección se imprime al principio de cada página. Por ajemplo, para repetir el título del informe en todas las páginas se utiliza el encabezado de página.
  + **Encabezado de grupo.** Esta sección se imprime al principio de cada nuevo grupo de registros. Use el encabezado de grupo para imprimir el nombre del grupo. Por ajemplo, en un informe que se agrupa por producto, use el encabezado de grupo para imprimir el nombre del producto. Cuando se coloca un control calculado que usa la función de agregado de suma en el encabezado de grupo, la suma se calcula para el grupo actual. Puede tener varias secciones de encabezado de grupo en un informe, en función de cuántos niveles de agrupación haya agregado.
  + **Detalle.** Esta sección se imprime una vez por cada fila del origen de registros. En ella se colocan los controles que constituyen el cuerpo principal del informe.
  + **Pie del grupo.** Esta sección se imprime al final de cada grupo de registros. Los pies de grupo se usan para imprimir información de resumen para un grupo. Puede tener varias secciones de pie de grupo en un informe, en función de cuántos niveles de agrupación haya agregado.
  + **Pie de página.** Esta sección se imprime al final de cada página. Los pies de página se utilizan para imprimir números de página o información sobre cada página.
  + **Pie del informe.** Esta sección se imprime una vez al final del informe. Los pies del informe se utilizan para imprimir totales de los informes u otra información de resumen de todo el informe .

## •

* 1. **Especificaciones para el diseño de informes**

Para diseñar y crear un informe con Access, debemos seguir una serie de pasos que garanticen su correcto desarrollo:

* + 1. **Diseñara la apariencia, funcionamiento y orígenes de datos del informe.** Los informes muestran los datos en pantalla o sobre el papel. Antes de diseñar el informe, tenemos que saber qué información queremos que muestre y cuál será el origen de esa información. Por ajemplo, si toda la información que queremos mostrar en el informe está en una única tabla. entonces usaremos esa tabla como base del informe. Si la información está en varias tablas, es majar crear una consulta que combine las tablas y basar el informe en esa consulta. Diseñar la tabla o consulta es tan importante como diseñar el informe. Si no hay una forma sencilla de que le informe reciba datos válidos. no será útil.
    2. **Crear un nuevo informe.** Una vez que se han determinado los datos que queremos que muestre el informe, podemos crear un informe en blanco. como se comentó en el tema de los formularios. los controles pueden ser dependientes. independientes o calculados. El origen de los datos que muestra un control ayuda a determinar qué tipo de control vamos a necesitar. Los controles dependientes son adecuados cuando queremos que muestren datos de un campo de la tabla o consulta subyacente. Los controles independientes son adecuados cuando el control muestra información que no se encuentra en otro lugar de la base de datos. En un control calculado se usa una expresión en lugar de un campo para determinar los datos a mostrar. Una vez creado el informe en blanco, añadimos controles usando los comandos análogos a los vistos en los formularios.
    3. **Ordenar y agrupar.** Ordenando los datos determinamos cómo los mostrará el informe. Agrupándolos conseguimos que los datos relacionados aparezcan juntos en el informe. Esto permite determinar de forma sencilla las relaciones entre los grupos. También proporciona una majar lectura y comprensión de los datos en la tabla o consulta subyacente, pero al realizarlo en el informe no necesitamos modificar el orden en la tabla o consulta .

# •

* + 1. **Añadir efectos especiales.** Podemos majorar la apariencia de los informes usando gráficos y diferentes efectos especiales. Un informe que contenga solamente texto puede ser muy aburrido. Los elementos gráficos majoran la legibilidad y contribuyen también a que el informe resulte más agradable a la vista. Utilizando las reglas que recorren la Vista Diseño del informe y la cuadrícula, podemos colocar los controles y elementos gráficos de manera que resulte más fácil y agradable leerlo.

**3. Herramientas**

**para**

**la creación de**

**formularios e informes**

A continuación vamos a ver las diferentes herramientas que nos aporta Microsoft Office Access 201O para la creación de formularios e informes.

**Crear un formulario**

Considere los formularios como ventanas por las que los usuarios ven y alcanzan las bases de datos. Un formulario eficaz acelera el uso de las bases de datos, ya que los usuarios no tienen que buscar lo que necesitan. Si un formulario es visualmente atractivo, resultará más agradable y más eficaz trabajar con una base de datos, además de ayudar a evitar que se introduzcan datos incorrectos. **Microsoft Office Access 2010** incluye nuevas herramientas que ayudan a crear formularios con rapidez, así como nuevos tipos de formulario y nuevas características que majoran el uso de las bases de datos.

**Herramienta Formulario**

Puede usar la herramienta **Formulario** para crear un formulario con un solo clic. Cuando usa esta herramienta, todos los campos de origen de datos subyacente están colocados en el formulario. Puede comenzar a usar inmediatamente el nuevo formulario, o bien, puede modificarlo en la vista Presentación o Diseño para ajustarlo a sus necesidades.

1. En el panel de navegación, haga clic en la tabla o consulta que contiene los datos que desee ver en el formulario.
2. En la ficha Crear, en el grupo Formularios, haga clic en Formulario.

•

'

=

Formularii:i Diseño del Formulario

1. istente pa ra f orm ula rías

iQ Navegación ..,.

formulari.:i

en blanco

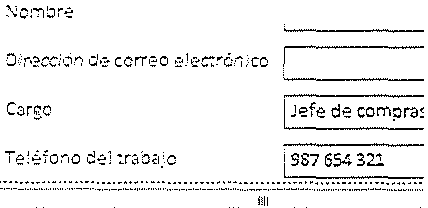
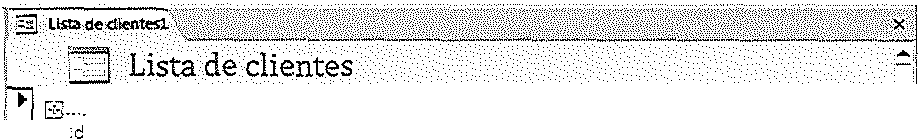
Formr.ilarl·:is

.·las formularioJs ...

1 Herramienta formulario

Herramienta Formulario dividido

Un formulario dividido proporciona dos vistas de los datos al mismo tiempo: una vista Formulario y una vista Hoja de datos. Las dos vistas están conectadas al mismo origen de datos y están en todo momento sincronizadas entre ellas. Si se selecciona un campo en una parte del formulario, se selecciona al mismo tiempo en la otra parte de éste. Se pueden agregar, editar o eliminar datos de ambas partes (siempre que el origen sea actualizable y el formulario no esté configurado para evitar estas acciones).



10

Alverca ' Luís



Compañi'aJ Alverca Luís

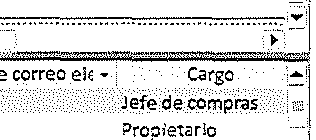
l Compañia A Benito Almudena Cornpai'ífa B Bermejo Antonio compañia K O:emando Compañi<i T Caejón Fra;iciscc Javie

""'º

Compañia O c:iaves Ferreir Pedro

Compañia L De Camargo Gustavo 14 Compañia N Estrada Modesto

Jefe de compras Jefe de compras Jefe de compras Jefe de compras Representante de



Prople'tario



Formulario dividido



Para crear un formulario dividido con la herramienta Formulario dividido:

* 1. En el panel de navegación, haga clic en la tabla o consulta que contiene los datos que desee incluir en el formulario. O bien, abra la tabla o consulta en la vista Hoja de datos.
  2. En el grupo Formulario de la ficha Crear, haga clic en Más formularios y, a continuación, haga clic en Formulario dividido '"";:ij .

**Herramienta varios elementos**

Cuando se crea un formulario mediante Ja herramienta Formulario, ese formulario muestra un solo registro a la *vez.* Si se desea disponer de un formulario que muestre varios registros pero más personalizable que una hqja de datos, se puede usar la herramienta Varios elementos.

1. En el panel de navegación, haga clic en la tabla o consulta que contiene los datos que desee *ver* en el formulario.
2. En el grupo Formularios de la ficha Crear, haga clic en Más formularios y,

===>

a continuación, haga clic en Varios elementos 2J

Cuando se usa la herramienta Varios elementos, el formulario creado por Access se parece a una hqja de datos. Los datos vienen organizados en filas y columnas y se ve más de un registro a la *vez.* Sin embargo, un formulario de varios elementos proporciona más opciones de personalización que una hqja de datos, como la posibilidad de agregar elementos gráficos, botones y otros controles.

**Asistente para formularios**

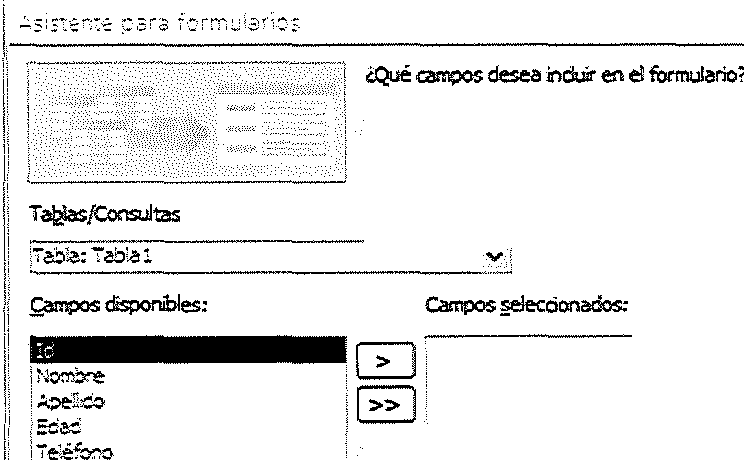
Para seleccionar con mayor criterio los campos que van a aparecer en un formulario, puede usar Asistente para formularios en vez de diversas herramientas de creación de formularios.

1. En Ja ficha Crear, en el grupo Formularios, haga clic en Asistente para formularios l'SJ . Aparecerá un cuadro de diálogo.
2. En el cuadro combinado Tablas y consultas, haga clic en el nombre de la consulta que desee usar como origen de registros del formulario.
3. En el cuadro de lista Campos disponibles, haga doble clic en cada uno de los campos que desee usar. Al hacer doble clic en un campo, éste se agrega al cuadro de lista Campos seleccionados.
4. Haga clic en **Siguiente** o en **Finalizar** cuando termine de agregar campos.

**Herramienta Formulario en blanco**

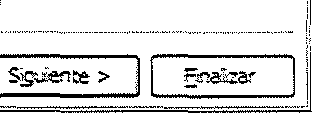
Si el asistente o las herramientas de creación de formulario no se ajustan a sus necesidades, puede usar la herramienta Formulario en blanco para crear un formulario. Puede ser una forma muy rápida de crear un formulario, especialmente si está pensando en incluir sólo unos pocos campos.

1. En la ficha Crear, en el grupo Formularios, haga clic en Formulario en blanco·-·



Puede elegi' de mós de una consulta o tabla.

<



**Asistente formularios**



Access abre un formulario en blanco en la vista Presentación y muestra el panel Lista de campos.

i Lista dec:;ampos ·. . · .. • X ! i W' Mostrar solamente los campos del origen de registros actual ¡ Campos disponibles en otras tablas: ¡

·a Clientes Editar tabla .a

Id ,

Com pañia !'

Apel!idos .

J I

1 NDoirmecbcrióen de correo e!ectr6nko .l

Cargo

1 Teléfono del tra bajo Teléfono particular Teléf ono móvil



1

1

¡ 1

1

1

1

Número de fax

1 i Dirección

' 1

1 Ciudad

Estado o provincia

C Postal

País o región

j Pagina Vieb

J N•Jtas

¡ 1 EB Datos adjuntos

EB Cuerdas

1 1±1 Detalles de pedido

l±l Detalles de pedidos de compra

1 El Empleados

¡ 1 EB Estado de detalle de pedidos

J 1 l±l Estado de impuestos de pedidos

l iEB Estado de pedidos

1 [ EB Estado de pedidos do: compra

1 @: Facturas

Editar tabla Editar tabla Editar tabla

Editar tabla

Editar tabla ··· l

Edítar tabla Editar tabla

1

Editar tabla 1

Editar tab!a 1

1

1±1 Informes de ventas

I±J Pedidos

Panel lista de campos

Editar tabla l

Editar tabla ·-;:

2, En el panel Lista de campos, haga clic en el signo más (+) situadojunto a la tabla o las tablas que contiene los campos que desee ver en el formulario,

3, Para agregar un campo al formulario, haga doble clic en él o arrástrelo hasta el formulario, Para agregar varios campos a la vez, mantenga presionada la tecla Control y haga clic en varios campos, A continuación, arrástrelos todosjuntos hasta el formulario,

Crear un Informe

Para crear un informe podemos utilizar las opciones del grupo Informes, en la pestaña Crear:

[] Asístente- para i nf ormes

Etiquetas

Inf orme Diseño de Inf orme­

i nf orme en bla nco

Infc1 rme-s

* + **Informe** consiste en crear automáticamente un nuevo informe que contiene todos los datos de la tabla o consulta seleccionada en el Panel de Navegación.
  + **Diseño de informe** abre un informe en blanco en la vista diseño y tenemos que ir incorporando los distintos objetos que queremos aparezcan en él. Este método no se suele utilizar ya que en la mayoría de los casos es más cómodo y rápido crear un autoinforme o utilizar el asistente y después sobre el resultado modificar el diseño para ajustar el informe a nuestras necesidades.
  + **Informe en blanco** abre un informe en blanco en vista Presentación.
  + **Asistente para informes** utiliza un asistente que nos va guiando paso por paso en la creación del informe.
  1. **Desarrollo de un generador de informes para la obtención de información**

El **Generador de informes** es un entorno de creación de informes destinado a los usuarios empresariales que prefieren trabajar en el entorno de Office Microsoft. Cuando se diseña un informe, se especifica de dónde se obtienen los datos, qué tipo de datos se obtienen y cómo se muestran. Al ajecutar el informe, el procesador de informes toma toda la información especificada, recupera los datos y los combina con el diseño del informe para generar el informe. Puede obtener una vista previa de los informes en el Generador de informes o publicar el informe en un servidor de informes o en un servidor de informes en el modo integrado de SharePoint, donde otros usuarios podrán ajecutarlo.

El Generador de informes permite hacer lo siguiente:

* + - Usar la cinta de opciones del Generador de informes para agregar elementos rápidamente a los informes, iniciar los asistentes para tabla, gráfico y mapa, y dar formato a los datos del informe.
    - Agregar datos de los proveedores de datos integrados mediante los diseñadores de consultas que le ayudan a especificar los datos que se van a incluir en el informe.
* Crear y usar los parámetros de informe y otras características interactivas que permiten a los lectores del informe para personalizar los datos y variar la presentación del informe.
* Crear expresiones de campos integrados, colecciones, operadores y

funciones.

* Abrir informes directamente desde un servidor de informes.
* Obtener una vista previa de los informes que usan orígenes de datos compartidos locales o publicados y conjuntos de datos compartidos.
* Obtener una vista previa de los informes en formato HTML o para impresión.
* Exportar los informes a otros formatos de archivo como Microsoft Excel.
* Guardar el informe y los elementos relacionados en una biblioteca de SharePoint, en un servidor de informes o en el equipo local.

**RECUERDA**

* Los formularios se conocen a veces como "pantallas de entrada de datos". Son las interfaces que se utilizan para trabajar con los datos y, a menudo, contienen botones de comando que ajecutan diversos comandos. Se puede crear una base de datos sin usar formularios, editando los datos de las hojas de las tablas. No obstante, casi todos los usuarios de bases de datos prefieren usar formularios para ver, escribir y editar datos en las tablas.
* Los informes sirven para resumir y presentar los datos de las tablas. Normalmente, un informe responde a una pregunta específica, como "¿Cuánto dinero se ha facturado por cliente este año?" o "¿En qué ciudades están nuestros clientes?" Cada informe se puede diseñar para presentar la información de la majar manera posible.
* Generalmente la herramienta más utilizada para trabajar con formularios e informes es **Microsoft Access 2010.**
* Mediante el **Asistente para exportación,** puede exportar datos de una base de datos de Access en un formato de archivo que Microsoft Excel puede leer.
* Los informes ofrecen el majar método para imprimir la información y distribuirla, y nos dan un gran control sobre el diseño general. Gracias a los informes podemos controlar fácilmente los estilos y tamaños de los tipos de letra, efectuar de forma sencilla los cálculos con los datos subyacentes, dar formato a los datos para que se ajusten a formularios ya diseñados e impresos, incluir gráficos, imágenes y otros elementos, o agrupar y organizar los datos para que resulte más fácil entender el informe.
* El **Generador de informes** es un entorno de creación de informes destinado a los usuarios empresariales que prefieren trabajar en el entorno de Office Microsoft. Cuando se diseña un informe, se especifica de dónde se obtienen los datos, qué tipo de datos se obtienen y cómo se muestran.

## •

**Preguntas de Autoeval uación**

1. **¿Con qué elemento podemos añadir datos a la base de datos?**
2. Con los informes.
3. Con las consultas.
4. Con los formularios.
5. **Los formularios se conocen a veces como:**
6. Pantallas de entrada de datos.
7. Pantallas de salida de datos.
8. Pantallas gráficas.
9. **Confirma si la siguiente afirmación es verdadera o falsa:**

*"Los formularios y los informes son totalmente iguales y cumplen las mismas funciones".*

1. Verdadero.
2. Falso.
3. **Si el libro de destino no existe y el objeto de origen es una tabla, consulta, formulario o informes, y desea exportar los datos con formato o sin formato; entonces:**
4. El libro se crea durante la operación de exportación.
5. El libro no se sobrescribe.
6. El libro se sobrescribe con los datos exportados.
7. **Confirma si la siguiente afirmación es verdadera o falsa:**

*..El Generador de informes es un entorno de creación de informes destinado a los usuarios empresariales que prefieren trabqjar en el entorno de Office Microsoft".*

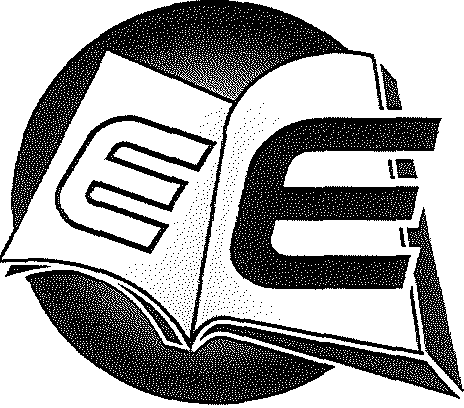
1. Verdadero.
2. Falso.

## •

Accesos a la

UD8

**información**



UF1888 Ope:rac;iones de Mantenimiento y

C- o--

n- . s,,..u-- - l- ta·- -- '

1. e-

Da.·. tos...

* 1. **Accesos mediante dispositivos ODBC**

Como introducción diremos que **ODBC** (Conectividad Abierta de Bases de Datos) es un protocolo que especifica un diseño de interfaz que todos Jos fabricantes utilizan para facilitar una comunicación confiable entre los programas de computadora y es diferentes bases de datos.

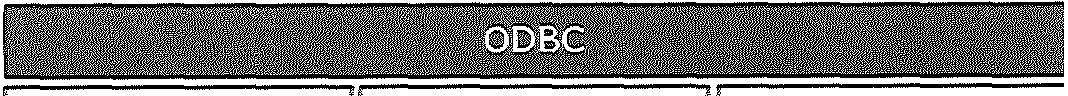
* + 1. **Definición del protocolo ODBC**

**Open Database Connectivity (ODBC)** es un estándar de acceso a Bases de Datos desarrollado por SQL Access Group en 1992.

El oQjetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS, el propósito de esta capa es traducir las consultas de datos de Ja aplicación en comandos que el DBMS entienda. Para que esto funcione tanto la aplicación como el DBMS deben ser compatibles con ODBC, esto es que la aplicación debe ser capaz de producir comandos ODBC y el DBMS debe ser capaz de responder a ellos. Desde la versión 2.0 el estándar soporta SAG y SQL.

El software funciona de dos modos, con un software manajador en el cliente, o una filosofía cliente-servidor. En el primer modo, el driver interpreta las conexiones y llamadas SQL y las traduce desde el APJ ODBC hacia el DBMS. En el segundo modo para conectarse a Ja base de datos se crea una DSN dentro del ODBC que define los parámetros, ruta y características de la conexión según los datos que solicite el creador o fabricante.

Java Database Connectivity (JDBC) es un derivado inspirado en el mismo, una interfaz de programación de aplicaciones que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ajecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.



Controlador SQL

Controlador Access

Otros controladores

1

1

Cliente SQL Server

Motor otras BD

Motor SQL Server

1

.

·

**Bases de** \_,.

r

\ **datos**

'

.

·

**Bases de datos**

1

**Bases de**

**datos**

**1.1.1. Ventajas del uso del acceso indirecto a través del ODBC**

La mayor ventaja es que el acceso a datos a través del ODBC permite el poder gestionar un amplio rango de datos con una sola interface. Desde que los más populares DBMS ofrecen drivers ODBC, muchas son las aplicaciones que lo incluyen como drivers de acceso.

Otra ventaja es que al poderse variar la fuente de datos, el cambio de una DBMS a otra es simplista. Por ajemplo, si se diseña un informe usando Oracle Data source, y más tarde, la empresa decide cambiar a Microsoft SQL Server, simplemente con cambiar el ODBC data source usado por el informe, se accede a la nueva DBMS. El único requerimiento es que la nueva fuente de datos ha de tener la misma estructura de tablas y campos que la fuente de datos original.

Además, el SQL que se envía al ODBC puede ser revisado y corregido antes de su envío, y por tanto se controla exactamente los datos que se quieren recuperar de la fuente.

## •

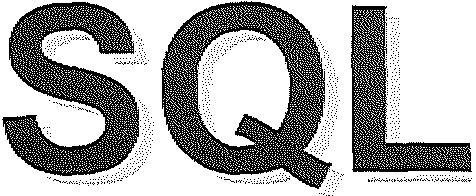
**1.1.2. Desventajas del ODBC**

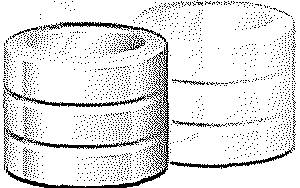
La principal desventaja son las capas a través de las cuales tiene que pasar la consulta, esto hace que el tiempo de respuesta hasta que se obtienen los datos se incremente. El proceso es que la petición ha de "traducirse" a ODBC, para que éste entienda la consulta. ODBC determina que fuente de datos contiene los datos que se piden y transmite la petición a la siguiente capa que es la fuente de datos ODBC (ODBC data source). La fuente de datos analiza la petición y "traduce" de nuevo la consulta a un formato que pueda ser "comprendido" por la DBMS. Este complajo proceso puede verse alterado por cualquier fallo en cualquiera de sus fases y por tanto la consulta no tendría éxito.

El OBDC tiene en cuenta dos factores claves para la empresa : El coste de la inversión Hardware y los costes asociados de Explotación. En este sentido el OBDC optimiza la inversión en hardware, ya que permite combinar diferentes tecnologías, consiguiendo no solo optimizar el coste por megabyte almacenado sino también disponer de un sistema abierto a cualquier tecnología presente y futura. Además, reduce los costes de gestión y explotación del sistema, descargando a los administradores de trabajo y reduciendo el coste del personal de mantenimiento.

**1.2. Utilización de consultas en lenguaje SQL**

Junto con las primeras bases de datos relacionales, a comienzos de la década de 1970 la empresa IBM desarrolló un lenguaje llamado Seque! para efectuar diversas operaciones con ellas. La evolución de ese lenguaje dio origen al denominado **Lenguaje Estructurado de Consultas** *( Structured Query Language, SQL) .* Su primera especificación estándar fue publicada en 1986.





El SQL es un lenguaje de naturaleza declarativa : le especifica a la base de datos la operación que debe efectuar, con una sintaxis (reglas de formación de las consultas SQL) y una semántica (nombres de las instrucciones) próximos a las de la lengua inglesa. Ese carácter lo hace muy sencillo de aprender y utilizar. A título de ajemplo, una sencilla declaración de SQL se escribirá así:

SELECT nombre, apellidos FROM clientes WHERE ciudad-"Madrid\*

Esta consulta o *query* SQL mostrará el nombre y los apellidos de los registros de la tabla clientes cuyo campo ciudad sea igual a «Madrid». Notemos que la sintaxis y los nombres de los operadores son relativamente intuitivos y sencillos de aprender con solo unas nociones básicas de lengua inglesa. Los nombres de los operadores en SQL se suelen escribir en mayúsculas para distinguirlos de los nombres de las tablas y campos, para los que habitualmente se recomienda el uso de minúsculas. En ocasiones, para evitar las posibles ambigüedades, el nombre de los campos se precede del nombre de la tabla a la que estos pertenecen, separándose ambos con un punto. De este modo la consulta anterior podría expresarse también así:

SELECT clientes.nombre, clientes.apellidos FROM clientes

WHERE clientes .ciudad="Madrid"

Notemos que es perfectamente posible introducir saltos de línea en la consulta para hacerla más legible.

* + 1. **Componentes del SQL**

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

**Comandos**

Existen dos tipos de comandos SQL:

* + - * **DDL** que permiten crear y definir nuevas bases de datos, campos e índices.
        + CREATE: utilizado para crear nuevas tablas, campos e índices. o DROP: empleado para eliminar tablas e índices.
        + ALTER: utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.
* **DML** que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.
  + SELECT: utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.
  + INSERT: utilizado para cargar lotes de datos en la base de datos en una única operación.
  + UPDATE: utilizado para modificar los valores de los campos y

registros especificados.

* + DELETE: utilizado para eliminar registros de una tabla de una base de datos.

**Cláusulas**

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

* FROM: utilizada para especificar la tabla de la cual se van a seleccionar los registros.
* WHERE: utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar.
* GROUP BY: utilizada para separar los registros seleccionados en grupos específicos.
* HAVING: utilizada para expresar la condición que debe satisfacer cada grupo.
* ORDER BY: utilizada para ordenar los registros seleccionados de acuerdo con un orden específico.

**Operadores**

Un operador es un símbolo que especifica una acción que se realiza en una o más expresiones.

* **Operadores lógicos:** comprueban la veracidad de alguna condición.
  + ANO: Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
  + OR: Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
  + NOT: Negación lógica. Devuelve el valor contrario de la expresión.
* **Operadores de comparación:** comprueban si dos expresiones son iguales. Se pueden utilizar en todas las expresiones excepto en las de los tipos de datos text, ntext o image.

o = (Igual a)

* + > (Mayor que)
  + < (Menor que)
  + >= (Mayor o igual que) o < = (Menor o igual que) o < > (No igual a)
  + != (No es igual a)
  + !< (No menor que) o !> (No mayor que)

**Funciones de agregado**

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

* AVG: utilizada para calcular el promedio de los valores de un campo determinado.
* COUNT: utilizada para devolver el número de registros de la selección.
* SUM: utilizada para devolver la suma de todos los valores de un campo determinado.

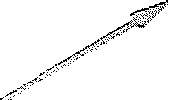
|  |  |  |  |
| --- | --- | --- | --- |
| * MAX: utilizada para devolver el valor más alto | de | un | campo |
| especificado. |  |  |  |
| * MIN: utilizada para devolver el valor más bajo especificado. | de | un | campo |

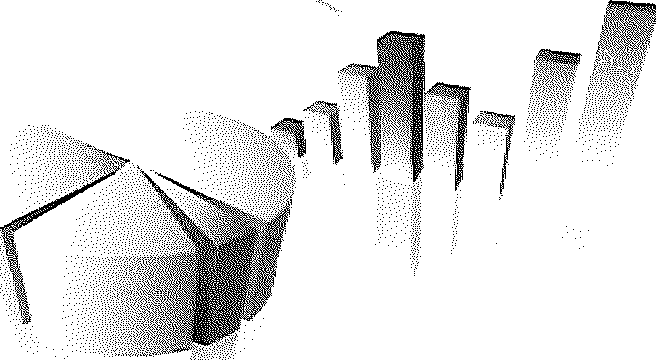
* + 1. **Consultas SQL**

Las consultas de SQL se clasifican de la siguiente forma :

* + - * De selección.
      * De acción.
      * De unión internas.
      * De unión externas.
      * De referencias cruzadas.
  1. **Generación de gráficos**

Un **gráfico** es la representación de datos, generalmente numéricos, mediante líneas, superficies o símbolos, para ver la relación que esos datos guardan entre sí y facilitar su interpretación.

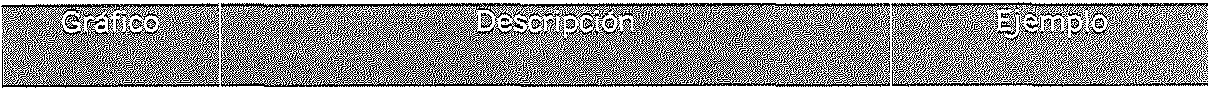




Los gráficos sirven para resumir en un dibl!\_jo toda una serie de datos que, presentados solos, resultan fríos y poco expresivos. Normalmente es un apartado que está íntimamente ligado a los conceptos de hqja electrónica y base de datos.

Pensemos en un fabricante al que le interesa saber el efecto sobre las ventas de un incremento del presupuesto de publicidad. Las cifras pueden calcularse mediante la hqja electrónica y puede tener sobre el papel, pero están presentadas de manera que son difíciles de comprender de comprender. Por esta razón, el fabricante puede optar por obtener estos datos de una manera mucho más explicita y fácil de asimilar, o sea, a través de una representación gráfica.

**Tipos de gráficos**



**Este gráfico sirve para comparar datos entre diferentes** I

**Barras/columnas**

segmentos (sectores, empresas, periodos de tiempo...). l.lL

'

**Líneas**

**Tartas**

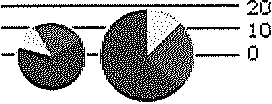
**Radar**

**Ayudan a ver la evolución de los datos. Por lo general se usan para mostrar un mismo tipo de dato y su evolución (valor de la acción y el tiempo, numero de ventas y** precio).

**Aquí podemos ver la contribución de cada parte a un total. Este grafico se puede utilizar de forma creativa comparando el tamaño de tas tartas entre si y el contenido de tas mismas**

**En el radar podemos ver la superficie creada por varias variables y asr poder comparar entidades (dos productos que presentan varias características pueden ser comparados en su totalidad usando esta grafica).**

..\



*f*

,-"'·'t.:..,. /

' ·.

Stocks

**Aquí se representan datos con 4 variables (tiempo, máximo, mínimo y cierre).**

r ,

Aquí el grid (líneas de división del eje) suele ser una

variable por si misma, haciendo que la disposición de las 9

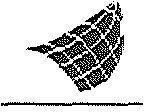
Burbt¡jas burbujas represente otras variablesjunto al propio tamaño de la burbyja. Este tipo de graficas permite concentrar mucha información en poco espacio.

•

Superficies

Este grafico se suele usar para ver la evolución de un dato sujeto a 3 variables. Por ejemplo la dureza de un material dependiendo de Ja temperatura, densidad y

volumen.



Representación gráfica del modelo Entidad-Relación

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en Jo que se ha denominado Modelo Entidad-Relación extendido. Los elementos básicos de representación del modelo son los siguientes:

1 c=r°

compuesto

Entídad Atrrbuto

11 11

Ent¡dad débH

* Relación

-o

Jerarquía de

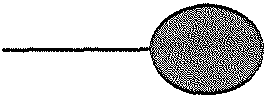
. generalización

Atributo / '

/ / ""''

'

Identificador



## •

1 1

* + **Diagrama entidad-relación:**
    - **Entidad.** Las entidades son el fundamento del modelo entidad relación. Podemos adoptar como definición de entidad cualquier cosa o parte del mundo que es distinguible del resto. Por ajemplo, en un sistema bancario, las personas y las cuentas bancarias se podrían interpretar como entidades. Las entidades pueden representar entes concretos, como una persona o un avión, o abstractas, como por ajemplo un préstamo o una reserva. Se representan por medio de un rectángulo. que pueden ser de tipo: maestras, transaccionales, históricas y temporales.
    - **Atributos.** Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta. Por motivos de legibilidad, los atributos suelen no aparecer representados en el diagrama entidad-relación, sino descritos textualmente en otros documentos aqjuntos.
    - **Relaciones.** Se representan mediante un rombo etiquetado en su interior con un verbo. Este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona, para así saber cuál es la relación que lleva cada uno.
  + **Diagramas extendidos:**
    - **Entidades fuertes y débiles.** Cuando una entidad participa en una relación puede adquirir un papel *fuerte* o *débil.* Una **entidad débil** es aquella que no puede existir sin participar en la relación; es decir, aquella que no puede ser unívocamente identificada solamente por sus atributos. **Una entidad fuerte** (también conocida como entidad regular) es aquella que sí puede ser identificada unívocamente. En los casos en que se requiera, se puede dar que una entidad fuerte "preste" algunos de sus atributos a una entidad débil para que esta última se pueda identificar. Las entidades débiles se representan- mediante un doble rectángulo; es decir, un rectángulo con doble línea.

## •

* + - * **Cardinalidad de las relaciones.** El tipo de cardinalidad se representa mediante una etiqueta en el exterior de la relación, respectivamente: "1:1", "1:N" y "N:M", aunque la notación depende del lenguaje utilizado, la que más se usa actualmente es el unificado. Otra forma de expresar la cardinalidad es situando un símbolo cerca de la línea que conecta una entidad con una relación:
        + "O" si cada instancia de la entidad no está obligada a participar en la relación.
* "1" si toda instancia de la entidad está obligada a participar en la relación y, además, solamente participa una vez.
* "N", "M", ó "\* " si cada instancia de la entidad no está obligada a participar en la relación y puede hacerlo cualquier número de veces.
  + - * **Atributos en las relaciones.** Las relaciones también pueden tener atributos asociados. Se representan igual que los atributos de las entidades. Un ajemplo tfpico son las relaciones de tipo "histórico" donde debe constar una fecha o una hora. Por ajemplo, supongamos que es necesario hacer constar la fecha de emisión de una factura a un cliente, y que es posible emitir duplicados de la factura (con distinta fecha). En tal caso, el atributo "Fecha de emisión" de la factura debería colocarse en la relación "se emite".
      * **Herencia.** Es un intento de adaptación de estos diagramas al paradigma orientado a objetos. La herencia es un tipo de relación entre una entidad "padre" y una entidad "hijo". La entidad "hijo" hereda todos los atributos y relaciones de la entidad "padre". Por tanto, no necesitan ser representadas dos veces en el diagrama. La relación de herencia se representa mediante un triángulo interconectado por líneas a las entidades. La entidad conectada por el vértice superior del triángulo es la entidad "padre". Solamente puede existir una entidad "padre" (herencia simple) . Las entidades "hijo" se conectan por la base del triángulo.

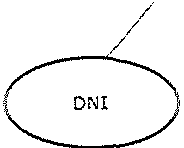
o **Agregación.** Es una abstracción a través de Ja cual las relaciones se tratan como entidades de un nivel más alto. Se utiliza para expresar relaciones entre relaciones o entre entidades y relaciones. Se representa englobando Ja relación abstraída y las entidades que participan en ella en un rectángulo. En la figura se muestra un ajemplo de agregación en el que se representa la situación en la que un profesor, cuando está impartiendo una clase, puede poner una incidencia ocurrida a lo largo de ésta (se fue la luz, falta Ja configuración de un determinado software, etc.) .

A continuación se muestra un ajemplo de cómo sería un diagrama del modelo entidad-relación.

/'

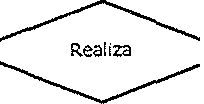
G

**Cliente**



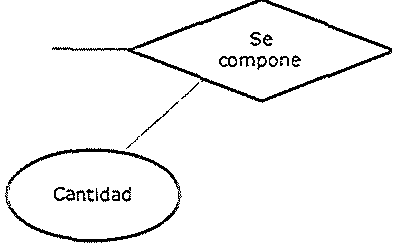
/

• /



**Ped1do**

**N:M**



•

.--------, .

**Articulo**

**Número**

**Serte**

**RECU ERDA**

* **Open Database Connectivity (ODBC)** es un estándar de acceso a Bases de Datos desarrollado por SQL Access Group en 1992.
* El oetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manajador de Bases de Datos, entre la aplicación y el DBMS, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda.
* La mayor **ventaja** es que el acceso a datos a través del ODBC permite el poder gestionar un amplio rango de datos con una sola interface. Desde que los más populares DBMS ofrecen drivers ODBC, muchas son las aplicaciones que lo incluyen como drivers de acceso.
* La principal desventaja son las capas a través de las cuales tiene que pasar la consulta, esto hace que el tiempo de respuesta hasta que se obtienen los datos se incremente. El proceso es que la petición ha de "traducirse" a ODBC, para que éste entienda la consulta.
* Junto con las primeras bases de datos relacionales, a comienzos de la década de 1970 la empresa 1BM desarrolló un lenguaje llamado Seque!

para efectuar

diversas operaciones con ellas. La evolución de ese

lenguaje dio

**Consultas**

origen al (*Structured*

denominado **Lenguaje Estructurado de**

*Query Language, SQL) .* Su primera

especificación estándar fue publicada en 1986.

* El lenguaje SQL está compuesto por **comandos, cláusulas, operadores y funciones de agregado.** Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.
* Un **gráfico** es la representación de datos, generalmente numéricos, mediante líneas, superficies o símbolos, para ver la relación que esos datos guardan entre sí y facilitar su interpretación.
* Los gráficos sirven para resumir en un dibujo toda una serie de datos que, presentados solos, resultan fríos y poco expresivos. Normalmente es un apartado que está íntimamente ligado a los conceptos de hoja

##### •

electrónica y base de datos.

* Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y lasjerarquías de generalización, en lo que se ha denominado **Modelo Entidad-Relación extendido.**

**Preguntas de Autoeval uación**

1. **Open Database Connectivity (ODBC) es un estándar de acceso a Bases de Datos desarrollado por SQL Access Group en:**

|  |  |
| --- | --- |
| a) | 1992 |
| b) | 1994 |
| c) | 1997 |

1. **¿Cuál es el objetivo de ODBC?**
2. Administrar todos los oQjetos de una Base de datos.
3. Hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos.
4. Representar de datos, generalmente numéricos, mediante líneas, superficies o símbolos, para ver la relación que esos datos guardan entre sí y facilitar su interpretación.
5. **Completa la siguiente afirmación con la opción correcta:**

"El lenguaje SQL está compuesto por comandos, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos".

1. OQjetos.
2. Entidades.
3. Cláusulas.
4. **Confirma si la siguiente afirmación es verdadera o falsa:**

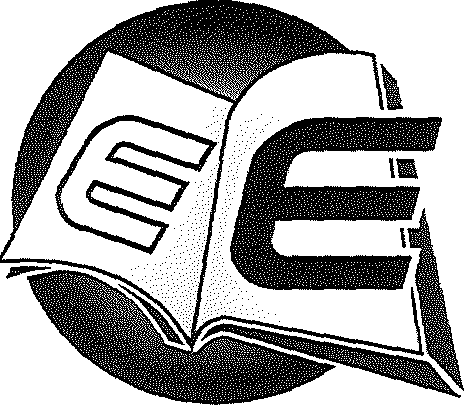
*"Un operador son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular".*

1. Verdadero.
2. Falso.
3. **Confirma si la siguiente afirmación es verdadera o falsa:**

*"Los gráficos sirven para resumir en un dibi¿jo toda una serie de datos que, presentados solos, resultan fríos y poco expresivos. Normalmente es un apartado que está íntimamente ligado a los conceptos de hQja electrónica y base de datos".*

1. Verdadero.
2. Falso.

**Actividades Prácticas**



UF1888 Operaciones de Mantenimiento y Consulta de Datos

UF1888 OPERACIONES DE MANTENIMIENTO Y CONSULTA DE DATOS

**Actividad Práctica RP1**

Para la elaboración de esta actividad práctica se atenderá a lo indicado en el Real Decreto 1531/2011, de 31 de octubre, en lo correspondiente a la UC1215\_3 (Realizar y mantener componentes software en un sistema de planificación de recursos empresariales y de gestión de relaciones con clientes).

El ajercicio que se expone a continuación, se corresponde con la realización profesional RP1 (Realizar operaciones de mantenimiento del diccionario de datos y de consulta en sistemas de planificación de recursos empresariales y de gestión de relaciones con clientes (sistemas ERP-CRM) para la extracción y posterior tratamiento de informaciones, siguiendo especificaciones de diseño establecidas)

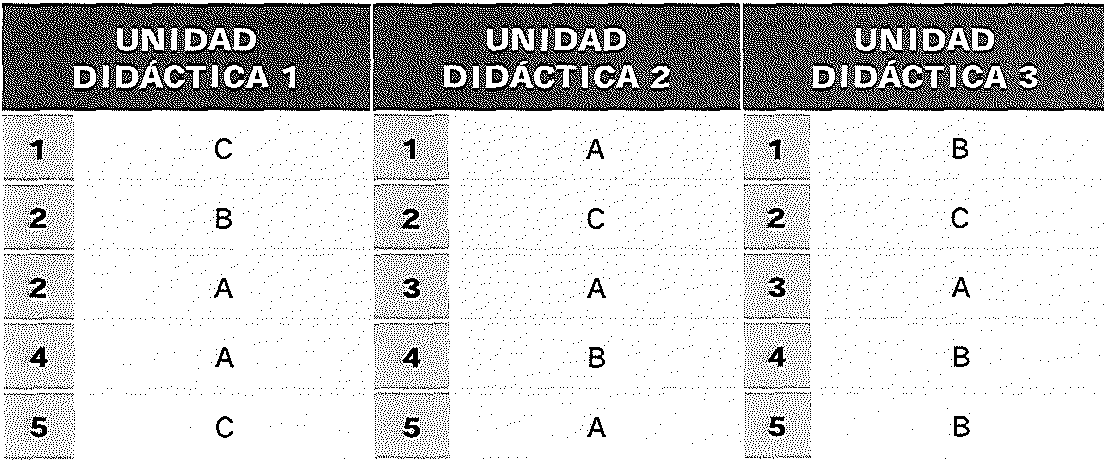
**Ejercicio**

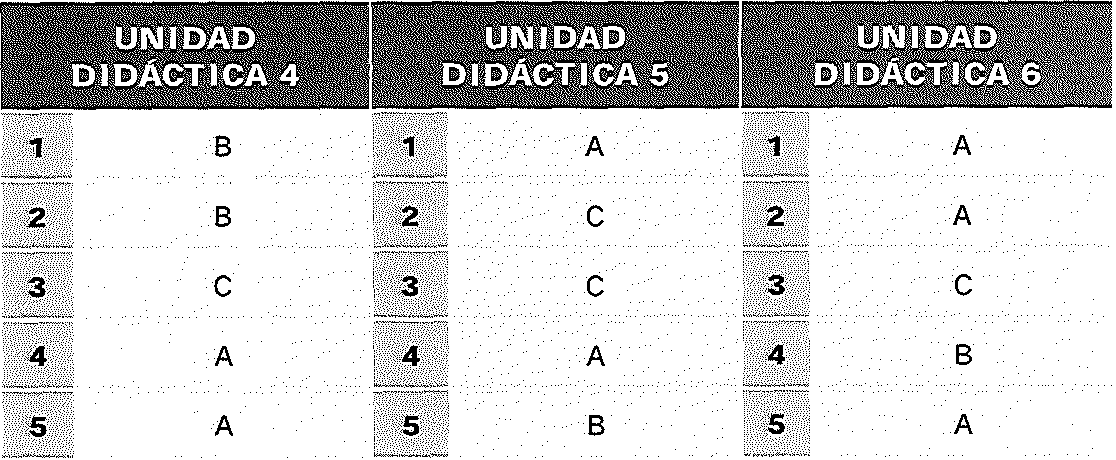
Se propone al alumnado que investigue sobre diversas herramientas para el mantenimiento del diccionario de datos y de consulta en sistemas de planificación de recursos empresariales y de gestión de relaciones con clientes (sistemas ERP­ CRM). Una vez que hayas recabado toda la información necesaria, expón la información encontrada de la forma más resumida posible y con la mayor coherencia posible.

En esta actividad se valorará la capacidad de síntesis para realizar dicho documento, así como las ideas expuestas y la coherencia de las mismas.

UF1888 OPERACIONES DE MANTENIMIENTO Y CONSULTA DE DATOS

**Respuestas a las Preguntas de Autoeva luación**





UF1888 OPERACIONES DE MANTENIMIENTO Y CONSULTA DE DATOS

