

FORMATO DE GUÍA METODOLÓGICA

1. IDENTIFICACIÓN DE GUÍA METODOLÓGICA

Nombre de la Asignatura: BASE DE DATOS II	Componentes del Aprendizaje	Horas de docencia: 54 Horas de aplicación práctica: 53 Horas de trabajo autónomo: 35
Resultado del Aprendizaje: <ul style="list-style-type: none"> Diseña modelos de datos relacionales. Construye Sistemas de Gestión de Base de Datos Relacionales. Aprende la fundamentación y el manejo de una base de datos Conoce la definición y modelado de una base de datos Utiliza herramientas para obtener y registrar información necesaria y fundamental para el desarrollo de una aplicación en particular. Desarrolla los diferentes artefactos y entregables de cada metodología revisada en el transcurso de este periodo 		
COMPETENCIAS Y OBJETIVOS <ul style="list-style-type: none"> Planificar, implementar y gestionar el uso de las Tecnologías de Información y Comunicación de una organización, a partir del análisis de sus requerimientos, teniendo en cuenta los criterios de calidad, seguridad y ética profesional propiciando el trabajo en equipo. Dominar los conceptos para la creación y manipulación de una Base de Datos considerando aspectos avanzados para el diseño e implementación de un sistema de Base de Datos, como son el control de la concurrencia y las técnicas de recuperación, mediante el análisis de modelos avanzados de bases de datos tales como las bases de datos orientada a objetos y base de datos distribuida. Diseña Bases de Datos Relacionales. Crea Bases de Datos Relacionales. 		
Docente de Implementación:		



Ing. Giovanni Cholca MSc.		Duración: 142 Horas		
Unidades	Competencia	Resultados de Aprendizaje	Actividades	Tiempo de Ejecución
ORGANIZACIÓN FÍSICA DE BASE DE DATOS	Enlistar conceptos básicos de la Organización Y almacenamiento Física de la Base de Datos.	COGNITIVO: Reconoce conceptos relacionados a la organización física de una BDD PROCEDIMENTAL: Implementa modelos de base de datos relacionales con estándares de calidad ACTITUDINAL: Describe las características de los componentes y funciones de una BDD	Exposición visual con diapositivas de conceptos básicos de organización física de las bdd Lluvia de ideas Estimulación multimedios con presentación de metodología de diseño de bases de datos relacionales	8
SISTEMAS DE GESTIÓN DE BASES DE DATOS	Analizar las características y funcionamiento de los sistemas de gestión de bases de datos	COGNITIVO: Distingue mecanismos de reconstrucción de BDD PROCEDIMENTAL: Examina el procesamiento de transacciones y acceso concurrentes en una BDD ACTITUDINAL: Describe las características de los componentes y funciones de una BDD.	Estimulación verbal, diálogo sobre conceptos y funciones de un SGBD Exposición multimedios sobre transacciones, accesos concurrentes y reconstrucción de una BDD Mapa mental Generación modelo físico BDD del caso de estudio	8





TIPOS DE BASES DE DATOS	Describir los diferentes tipos de bases de datos	COGNITIVO: Reconoce fundamentos de los modelos de BDD tradicional y jerárquico PROCEDIMENTAL: Compara funcionamiento e implementación de los modelos de BDD de red y relacional ACTITUDINAL: Analiza consideraciones para implementación de los modelos de BDD avanzados, OO y Declarativo	Estimulación multimedia, presentación visual sobre modelos tradicionales de BDD Argumentación con los estudiantes sobre criterios de selección del modelo de la BDD	8
LABORATORIO DE BASES DE DATOS	Experimentar en laboratorio la implantación de un sistema de base de datos	PROCEDIMENTAL: Gestiona información desde una BDD distribuida ACTITUDINAL: Diseña y construye una Bdd relacional PostgreSQL	Lluvia de ideas Exposición individual BDD NoSql, estudio de casos Observación directa en computador Ejemplificación caso de estudio	8





2. CONOCIMIENTOS PREVIOS Y RELACIONADOS

Co-requisitos

1. El estudiante al finalizar su formación profesional, debe ser capaz de demostrar sus capacidades relacionadas con conocimientos (saber), habilidades y destrezas (saber hacer) y actitudes (saber ser) en relación con las necesidades del contexto, particularmente:
2. Programa aplicaciones de software utilizando lógica de programación y lenguajes específicos, para resolver un problema determinado.
3. Desarrolla software para cualquier ambiente empresarial a partir de la detección de necesidades, levantamiento de información, diseño, análisis, programación, puesta en marcha y entrega funcionando, incluyendo instalación y la capacitación de personal.
4. Administra sistemas operativos (Windows y basados en Unix) de acuerdo a las necesidades de la organización, utilizando los comandos propios del sistema, para el óptimo funcionamiento de los recursos informáticos.
5. Desarrolla software de aplicaciones, con el fin de optimizar el manejo de ésta dentro de cualquier organización, administrar bases de datos, redes de información y páginas Web
6. Desarrolla e integrar soluciones de software en diferentes entornos,
7. Programa sistemas y aplicaciones capaces de codificar, mantener y actualizar los programas asociados a un proyecto de Ingeniería de Software, garantizando eficacia y eficiencia de los mismos.
8. Desarrolla programas de computador en diferentes lenguajes y bajo enfoques metodológicos diversos.
9. Conoce los modelos conceptuales y tecnológicos relacionados con las Bases de Datos, y los utiliza como herramienta en la construcción de software.





3. UNIDADES TEÓRICAS

A. Base Teórica

Introducción

Las bases de datos son un elemento fundamental en el entorno informático hoy en día y tienen aplicación en la práctica totalidad de campos. Concebidas con un propósito general, son de utilidad para toda disciplina o área de aplicación en la que exista una necesidad de gestionar datos, tanto más cuanto más voluminosos sean estos. Además, el trabajo con un SIG presenta una serie de características (uso múltiple de los datos, necesidad de acceso eficiente para análisis, necesidad de indexación, etc.) que hacen que sea recomendable el uso de bases de datos.

¿Por qué interesa usar una base de datos?

En base al ejemplo anterior, podemos analizar algo más sistemáticamente las ventajas de una base de datos frente a una gestión no organizada de los datos. Algunas ventajas que afectan directamente a los datos son las siguientes:

- **Mayor independencia.** Los datos son independientes de las aplicaciones que los usan, así como de los usuarios.
- **Mayor disponibilidad.** Se facilita el acceso a los datos desde contextos, aplicaciones y medios distintos, haciéndolos útiles para un mayor número de usuarios.
- **Mayor seguridad (protección de los datos)** . Por ejemplo, resulta más fácil replicar una base de datos para mantener una copia de seguridad que hacerlo con un conjunto de ficheros almacenados de forma no estructurada. Además, al estar centralizado el acceso a los datos, existe una verdadera sincronización de todo el trabajo que se haya podido hacer sobre estos (modificaciones), con lo que esa copia de seguridad servirá a todos los usuarios.
- **Menor redundancia.** Un mismo dato no se encuentra almacenado en múltiples ficheros o con múltiples esquemas distintos, sino en una única instancia en la base de datos. Esto redundará en menor volumen de datos y mayor rapidez de acceso.
- **Mayor eficiencia en la captura, codificación y entrada de datos.**

Esto tiene una consecuencia directa sobre los resultados que se obtienen de la explotación de la base de datos, presentándose al respecto ventajas como, por ejemplo:





- **Mayor coherencia** . La mayor calidad de los datos que se deriva de su mejor gestión deriva en mayor calidad de los resultados.
- **Mayor eficiencia**. Facilitando el acceso a los datos y haciendo más sencilla su explotación, la obtención de resultados es más eficiente.
- **Mayor valor informativo**. Resulta más sencillo extraer la información que los datos contienen, ya que uno de los cometidos de la base de datos es aumentar el valor de estos como fuente de información.

Por último, los usuarios de la base de datos también obtienen ventajas al trabajar con estas, entre los que cabe citar:

- **Mayor facilidad y sencillez de acceso**. El usuario de la base de datos se debe preocupar únicamente de *usar* los datos, disponiendo para ello de las herramientas adecuadas y de una estructura sólida sobre la que apoyarse.
- **Facilidad para reutilización de datos**. Esto es, facilidad para compartir.

De forma resumida, puede decirse que la principal bondad de una base de datos es la centralización que supone de todos los datos con los que se trabaja en un contexto determinado, con las consecuencias que ello tiene para una mejor gestión, acceso o estructuración de estos.

Modelos de bases de datos

En función de la estructura utilizada para construir una base de datos, existen diversos modelos. El modelo de la base de datos define un paradigma de almacenamiento, estableciendo cómo se estructuran los datos y las relaciones entre estos. Las distintas operaciones sobre la base de datos (eliminación o sustitución de datos, lectura de datos, etc.) vienen condicionadas por esta estructura, y existen notables diferencias entre los principales modelos, cada uno de ellos con sus ventajas e inconvenientes particulares. Algunos de los más habituales son los siguientes:

- **Bases de datos jerárquicas**. Los datos se recogen mediante una estructura basada en nodos interconectados. Cada nodo tiene un único padre, y cero, uno o varios hijos. De este modo, se crea una estructura en forma de árbol invertido en el que todos sus nodos dependen en última instancia de uno denominado *raíz* . Aunque potente, el modelo jerárquico presenta algunas deficiencias, principalmente la escasa independencia de sus registros (el acceso a un registro —un nodo— implica que se ha de pasar por sus padres, restando flexibilidad a la navegación por la base de datos). Otra grave deficiencia de este modelo es la mala gestión de la redundancia de datos, ya que si un registro guarda relación con dos o más, debe almacenarse varias veces, al no estar permitido que el nodo





correspondiente tenga varios padres. Esto tiene consecuencias no solo en el mayor volumen de datos que se almacena, sino también en la integridad y coherencia de los datos. Si se modifica una de las «copias» de ese registro en la base de datos, deben modificarse también las restantes, ya que, aunque no conectadas en la estructura de la base de datos, realmente representan una única realidad y debieran ser idénticas entre sí.

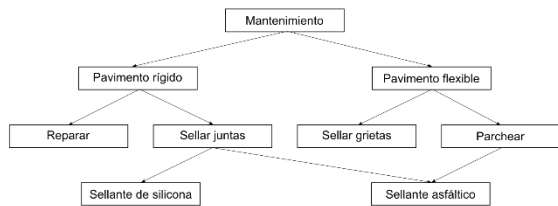
- **Bases de datos en red.** Con objeto de solucionar los problemas de redundancia de las bases de datos jerárquicas, surge el modelo en red. Este modelo permite la aparición de ciclos en la estructura de la base de datos (es decir, no ha de existir un único padre para cada nodo), lo cual permite una mayor eficacia en lo que a la redundancia de datos se refiere. Presenta, no obstante, otros problemas, siendo el más importante de ellos su gran complejidad, lo que hace difícil la administración de la base de datos.
- **Bases de datos relacionales.** Constituyen el modelo de bases de datos más utilizado en la actualidad. Solucionan los problemas asociados a las bases de datos jerárquicas y en red, utilizando para ello un esquema basado en tablas, que resulta a la vez sencillo de comprender y fácil de utilizar para el análisis y la consulta de los datos. Las tablas contienen un número dado de *registros* (equivalentes a las filas en la tabla), así como *campos* (columnas), lo que da lugar a una correcta estructuración y un acceso eficiente.
- **Bases de datos orientadas a objetos.** Se trata de uno de los modelos más actuales, derivado directamente de los paradigmas de la programación orientada a objetos. El modelo extiende las capacidades de las bases de datos relacionales, de tal modo que estas pueden contener objetos, permitiendo así una integración más fácil con la propia arquitectura de los programas empleados para el manejo de la base de datos, en caso de que estos hayan sido desarrollados mediante programación orientada a objetos. Su popularidad crece de forma notable en ciertas áreas en las cuales resultan más ventajosas que el modelo relacional, siendo los SIG una de ellas.

La figura muestra una comparación esquemática de los anteriores modelos de bases de datos.





Red



Relacional

Fecha	Código	Ruta
01/10/01	24	I-95
15/12/01	23	I-495
17/03/02	24	I-66

Clave: 24

Nombre actividad	Fecha	Ruta
Asfaltado	01/10/01	I-95
Asfaltado	17/03/02	I-66

Código	Nombre actividad
23	Parcheado
24	Asfaltado
25	Sellado de grietas

Jerárquico



Orientado a objetos

Fecha	01/12/01
Actividad	24
Ruta	I-95
Producción diaria	2.5
Horas equipamiento	6
Horas labor	6

Objeto 1: Informe de mantenimiento

Instancia del objeto 1

Código actividad
Nombre actividad
Unidad de producción
Producción diaria media

Objeto 2: Actividad de mantenimiento

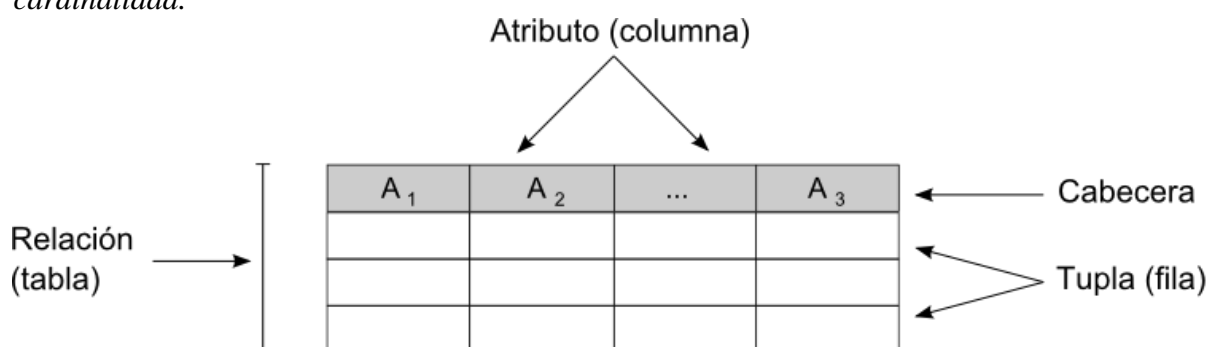
Comparación entre algunos modelos de base de datos más frecuentes

Aunque, como ya hemos visto, existen diversos tipos de bases de datos, las más utilizadas con diferencia en la actualidad son las relacionales, que han demostrado su idoneidad en la mayor parte de situaciones. Estas son también las que encontraremos en el ámbito SIG, y resulta por ello necesario añadir algunas nociones adicionales sobre ellas para la correcta comprensión no solo de este capítulo, sino también de otros posteriores que desarrollan temas relacionados.

Además de las denominaciones habituales de *tabla*, *fila* y *columna*, existe una terminología específica empleada al referirse a las bases de datos relacionales. Así, en el modelo relacional los datos se organizan en tablas bidimensionales, cada una de ellas con información relativa a una determinada entidad. La tabla en sí se conoce como *relación*, ya que recoge la relación existente entre sus elementos, y constituye así el eje central del modelo relacional. Dentro de la tabla, los datos están organizados a su vez en filas y columnas. Las columnas representan los distintos *atributos* asociados a la entidad, mientras que las filas conforman los distintos *registros*. Una fila se forma con un conjunto de n atributos, constituyendo una *tupla*. El *esquema de la relación* está formado por los nombres de los atributos y un *dominio* asociado a estos, que delimita el rango de valores posibles para cada atributo. El dominio especifica el tipo de dato a contener en cada columna. Por ejemplo, si se recoge un nombre el atributo será de tipo alfanumérico, mientras que si el atributo es un conteo deberá ser de tipo entero. Además de los tipos habituales (fechas, cadenas de texto, valores reales). Puede emplearse también la denominación menos formal de *número decimal* o bien *valor de coma flotante*, esta última más



común en el ámbito informático y referida a la forma de almacenamiento de este tipo de valores.}, valores enteros, etc.) pueden emplearse en ciertas bases de datos valores más complejos. Esto es de especial interés en el caso de los SIG, ya que permite utilizar geometrías como un tipo de datos más, con la utilidad que esto tiene a la hora de almacenar datos espaciales. El esquema de la relación se recoge en la primera fila de la tabla, conocida como *cabecera*. El número de filas de la tabla sin contar la cabecera (es decir, el número de tuplas) se conoce como *cardinalidad*.



Elementos del modelo relacional

Las relaciones son, por tanto, un conjunto de tuplas asociadas a un esquema. En una relación, tanto el orden de las filas como el de las columnas son irrelevantes (exceptuando la cabecera, que no es una tupla como tal, sino que define el esquema como hemos visto), pero es importante que cada atributo sea del tipo correspondiente a la columna a la que pertenece. Es decir, que sea coherente con el esquema.

En la figura puede verse un esquema de los elementos fundamentales del modelo relacional.

Una forma abreviada de definir las relaciones que forman parte de una base de datos es mediante su nombre y su esquema expresado como una lista de los atributos que lo constituyen. Por ejemplo, podemos definir una relación denominada PERSONAS como

PERSONAS (DNI, Nombre, Altura, Edad, Ciudad)

Una base de datos contiene normalmente más de una tabla, ya que suelen ser muchos los tipos de datos a almacenar y resulta conveniente dividirlos en distintas tablas. Además de las relaciones que la tabla en sí implica, es necesario definir relaciones entre las distintas tablas, y para ello se emplean los denominados atributos *clave*. Un atributo clave es aquel que tiene valor único para cada tupla, pudiendo servir para representar a esta plenamente. Por ejemplo, en una tabla con nombres de personas e información adicional sobre ellas según el esquema anterior, los nombres no pueden ser la clave primaria, ya que puede haber dos personas con un mismo nombre. El número de su Documento Nacional de Identidad, sin embargo, sí que puede servir como atributo clave. Además de su unicidad, una clave debe ser invariable, identificando la



misma tupla a lo largo del tiempo. Un esquema de relación puede contener varios atributos clave, que se conocen como *claves candidatas*. Normalmente, de estas se elige una como representante principal de las tuplas, y se conoce como *clave primaria*

Por convención, las claves se escriben subrayadas al definir el esquema de la tabla, de tal modo que el de la tabla PERSONAS quedaría de la siguiente forma:

PERSONAS (DNI , Nombre, Altura, Edad, Ciudad)

Si no existe ningún atributo que cumpla los requisitos para ser utilizado como clave, este puede incorporarse al esquema de la relación, añadiendo por ejemplo un nuevo atributo con un código arbitrario. Un ejemplo de esto lo podemos encontrar en el cuadro, donde se incorpora un atributo que hace la función de clave a una tabla con información sobre personas pero que no contiene el DNI de estas entre esa información y, por tanto, carece de un atributo adecuado para servir de clave.

En la definición de clave cabe también la presencia de claves compuestas, es decir, formadas por varios atributos cuya combinación es única para cada tupla. No obstante, la utilización de claves simples es preferible generalmente, ya que simplifica gran parte de las operaciones en las que la presencia de una clave es necesaria.

Tabla a

DNI	Nombre	Altura	Edad	Ciudad
50234561	Juan Gómez	1,85	35	Madrid
13254673	Eduarne Montero	1,60	30	Toledo
46576290	Luis Urrutia	1,75	46	Madrid
38941882	Juan Gómez	1, 71	55	Valencia

Tabla b

ID	Nombre	Altura	Edad	Ciudad
001	Juan Gómez	1,85	35	Madrid
002	Eduarne Montero	1,60	30	Toledo
003	Luis Urrutia	1,75	46	Madrid
004	Juan Gómez	1, 71	55	Valencia

Sistemas gestores de bases de datos

Junto con las bases de datos, el elemento fundamental para el aprovechamiento de estas son los *Sistemas Gestores de Bases de Datos* (SGDB o DBMS, del inglés *DataBase Management*



System). Estos sistemas representan un elemento intermedio entre los propios datos y los programas que van a hacer uso de ellos, facilitando las operaciones a realizar sobre aquellos. En nuestro caso, son el componente que permite unir el SIG con la base de datos en la que se almacenan los datos espaciales con los que este va a trabajar.

Un SGBD es una pieza de software compleja, ya que las situaciones a las que debe responder son diversas y en muchas ocasiones con requerimientos elevados, por ejemplo en lo que a eficiencia y volumen de datos respecta. Piénsese que una base de datos actual puede tener millones de registros y ser utilizada simultáneamente por miles de usuarios, que a su vez pueden utilizar diversos programas, no todos ellos del mismo tipo. Por ejemplo, una base de datos que contenga números de teléfono, nombres de usuarios, direcciones y coordenadas asociadas a cada línea telefónica, puede ser empleada desde un SIG para crear un mapa que muestre la densidad de usuarios o también desde una aplicación que genere un listín telefónico, o bien desde una aplicación en una página Web que permita localizar el número de teléfono de una persona concreta. Cada una de estas aplicaciones realiza un trabajo distinto, pero todas ellas utilizan la misma base de datos. El SGBD debe proporcionar a todos ellos la metodología adecuada para extraer del conjunto de datos completo cuanto sea necesario en cada caso.

Además, el SGBD es la herramienta utilizada no solo por quienes aprovechan los datos, sino también por aquellos que se han de encargar de la propia gestión y mantenimiento de la base de datos. Administrar una base de datos puede suponer una tarea altamente compleja, por lo que el SGBD debe proveer los útiles necesarios para llevar a cabo ese mantenimiento.

Para ser de verdadera utilidad y responder a todas las necesidades que pueden plantearse en relación con la base de datos, un SGBD debe perseguir los siguientes objetivos:

- **Acceso transparente a los datos** . La base de datos ha de poder accederse de forma transparente, sin que sea necesario para el usuario del SGBD preocuparse por aspectos internos relativos a la estructura de esta u otras características. Esto significa que, por ejemplo, si queremos recuperar un registro de la base de datos, debemos poder hacerlo sin necesidad de saber si dicha base de datos está almacenada en un único archivo o varios, o si el registro que pretendemos recuperar está almacenado a su vez de uno u otro modo. Así, el SGBD debe crear una abstracción de los datos que haga el trabajo con estos más sencillo, ocultando aspectos que no sean relevantes para dicho trabajo. Procedimientos como las consultas que veremos en el capítulo [Consultas](#) se realizan a través del SGBD, que es quien se encarga de interpretar dichas consultas, aplicarlas

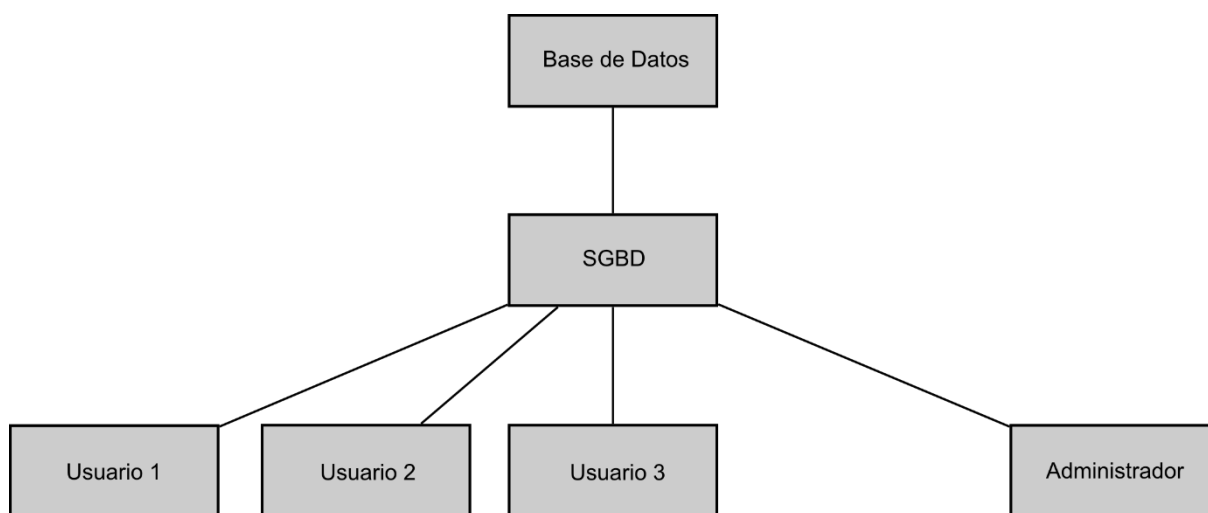




sobre la base de datos y devolver el resultado correspondiente. El SIG no accede a los datos, sino que se comunica con el SGBD y deja en manos de este el proceso de consulta en sí.

- **Protección de los datos** . Si la base de datos almacena información sensible, el SGBD debe controlar el acceso a esta, restringiendo el acceso cuando corresponda (por ejemplo, estableciendo distintos permisos de acceso para distintos tipos de usuarios) e implementando los mecanismos de protección necesarios.
- **Eficiencia** . Acceder a los datos no es suficiente en la mayoría de los casos, sino que se requiere un acceso eficiente. El SGBD debe ser capaz de gestionar de forma fluida grandes volúmenes de datos o de operaciones (por ejemplo, muchos usuarios accediendo simultáneamente), de modo que dé una respuesta rápida a las peticiones de los usuarios de la base de datos.
- **Gestión de transacciones** . Las operaciones sobre la base de datos tales como la adición o borrado de un registro se realizan mediante transacciones. Una transacción es un conjunto de operaciones realizadas por un usuario sobre la base de datos como una única unidad de trabajo, de forma indivisible. El SGBD ha de encargarse de gestionarlas de manera eficiente y segura para que todos los usuarios de la base de datos puedan hacer su trabajo de forma transparente. Aspectos como el acceso concurrente a la base de datos (varias transacciones simultáneas) resultan especialmente importantes, y en su buena gestión se pone gran esfuerzo en el diseño de los SGBD. Se denomina *transaccional* al SGBD capaz de garantizar la integridad de los datos, no permitiendo que las transacciones puedan quedar en un estado intermedio. Esto implica la capacidad de poder volver a un estado anterior en caso de que por cualquier causa (error en el sistema, fallo eléctrico, etc) no haya podido completarse la transacción.





Representación esquemática del papel de un Sistema Gestor de Base de Datos

La figura esquematiza el papel que el SGBD juega en el manejo y empleo de los datos. Tanto los distintos usuarios (en el caso de nuestro supuesto de gestión forestal pueden ser desde el gestor forestal al cartógrafo encargado de actualizar los límites de las unidades inventariables) como el administrador de la base de datos acceden a esta a través del SGBD. No existe acceso directo a la base de datos.

El SGBD tendrá unas u otras características en función del modelo de base de datos subyacente, ya que debe adaptarse a las características de este para ofrecer las funcionalidades correspondientes en el nivel de usuario.

Diseño y creación de una base de datos

Una vez se toma la decisión de emplear una base de datos, el siguiente paso es el diseño y creación de esta. El diseño implica la definición de la estructura que va a tener la base de datos, que se deberá realizar teniendo en cuenta principalmente el tipo de datos que van a almacenarse y el modelo de base de datos elegido. El diseño debe adecuarse al uso previsto de la base de datos, de tal modo que acomode los datos de la mejor forma posible para cumplir los objetivos enunciados anteriormente en este mismo capítulo. Para ello debe conocerse la naturaleza de los datos que van a almacenarse (no necesariamente datos de los que se dispone en el momento de la creación, sino los que se espera pasen a formar parte de la base de datos a lo largo de su ciclo de vida), así como la de los algoritmos y procesos que van a emplearse sobre ellos.

Posteriormente al diseño, debe procederse a la implementación de la base de datos, esto es, a la creación propiamente dicha, incorporando los datos según los esquemas escogidos en la fase de diseño. Por último, y una vez creada la base de datos, debe procurarse un mantenimiento para que esté continuamente en condiciones de ser utilizada.



Más concretamente, pueden distinguirse las siguientes fases en el proceso global de desarrollo de una base de datos:

- **Diseño lógico.** Independiente del SGBD empleado, es un diseño conceptual que pretende modelizar el contenido de la base de datos.
- **Diseño físico.** Es la adaptación del diseño conceptual a las particularidades del SGBD escogido.
- **Implementación.** Introducción de los datos en la base de datos.
- **Mantenimiento.** Monitorización de la actividad sobre la base de datos.

La primera fase en el diseño de una base de datos implica un análisis de los datos que se van a recoger. Como resultado de ese análisis debe surgir un modelo conceptual que exprese la estructura de la información, siendo dicha estructura susceptible de ser empleada como esquema de partida para la base de datos en cuestión. El modelo conceptual ha de definir básicamente los tipos de datos a tratar y las relaciones existentes entre ellos, elementos que serán luego expresados en términos del modelo de base de datos elegido (relacional, orientado a objetos, etc.) una vez se pase a la fase de diseño físico.

El modelo conceptual debe estructurar la información de forma que el usuario de la base de datos comprenda de forma sencilla el contenido y forma de esta. Por tanto, debe desarrollarse teniendo presentes las necesidades de los usuarios y el hecho de que estos no necesariamente han de ser especialistas en bases de datos, sino especialistas en los propios datos en sí. Por otra parte, el modelo debe intentar capturar del mejor modo posible la realidad que se pretende modelizar, por lo que el conjunto de tipos de datos y relaciones debe elaborarse de modo similar a dicha realidad para recoger toda la complejidad del sistema. Y, por supuesto, el modelo debe poder ser implementado posteriormente y utilizado en conjunto con el SGBD escogido, ya que de otro modo no presenta utilidad práctica.

Existen diversas metodologías para desarrollar un modelo conceptual. Una de las más extendidas por su sencillez y potencia es la del *modelo entidad--relación* (abreviadamente, modelo E-R).

Denominamos *entidad* a un objeto o concepto del mundo real acerca del cual se recoge información, y que puede diferenciarse de otros objetos, incluso si son de su misma clase (un ordenador, por ejemplo, es un objeto, y puede diferenciarse de otros ordenadores, incluso si son de idénticas características, ya que no son todos el mismo objeto y ese en particular tendrá



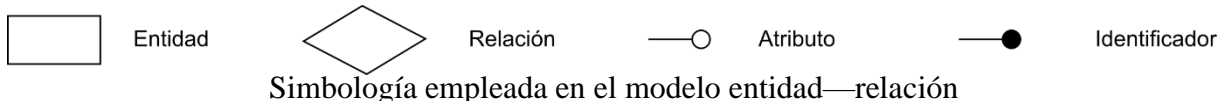


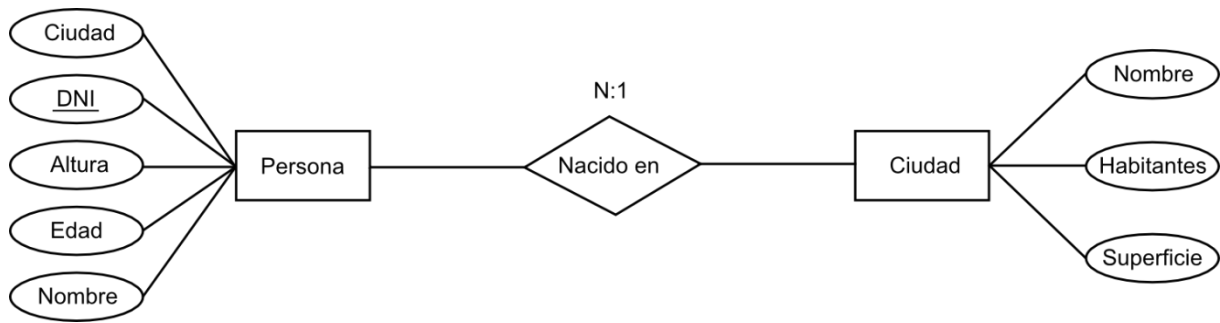
alguna propiedad distinta, como puede ser el número de serie). La entidad puede tener sentido físico o bien ser una idea abstracta, como un tipo de deporte, una clase de música o una palabra. Una entidad se describe mediante una serie de características o atributos, que son las que definen su naturaleza y sus propiedades. Una colección de entidades es un conjunto de entidades distintas (que representan a objetos distintos), las cuales comparten unos atributos comunes. Por ejemplo, un conjunto de ordenadores de los cuales se conocen los atributos *modelo*, *marca* y *procesador*.

Por su parte, una *relación* expresa la dependencia existente entre entidades y permite la asociación de estas. No resulta difícil ver que estos conceptos —entidad, atributos y relación— guardan un notable paralelismo con las ideas del modelo relacional que ya conocemos. Así, y aunque no resulte por completo inmediato, es sencillo traducir un modelo entidad-relación (conceptual) a un modelo relacional, que constituye ya un modelo aplicado a un tipo particular de base de datos. Por ello, el modelo E-R es una herramienta potente para el diseño lógico de la base de datos, especialmente si esta utiliza el modelo relacional.

Para desarrollar el diseño conceptual de una base de datos siguiendo el modelo E-R, estos son los pasos principales:

- Partimos de una descripción textual del problema o sistema que queremos recoger. Esta descripción contiene los requisitos necesarios y ha de formular la pregunta a la que queremos que la base de datos dé respuesta. Para nuestro ejemplo con datos sobre personas y ciudades, el problema podríamos formularlo como «¿qué personas han nacido en cada ciudad?».
- Se toman los verbos y los sustantivos de la descripción textual. Los sustantivos son posibles entidades o atributos, mientras que los verbos son posibles relaciones. En nuestro caso, «persona» y «ciudad» serán entidades y «nacido en» una relación.
- Se analizan las frases y determina la cardinalidad de las relaciones y otros detalles.





Ejemplo de diagrama E-R

El modelo así creado se expresa mediante un diagrama en el que las entidades se representan como cajas rectangulares, las relaciones mediante rombos y los atributos en círculos o elipses, todos ellos con sus correspondientes nombres en el interior. Cuando un atributo es un identificador, se representa con su nombre subrayado, del mismo modo que en la definición de esquemas que ya vimos anteriormente

Bases de datos espaciales

Todo cuanto hemos visto en los puntos anteriores constituye el conjunto de ideas fundamentales sobre las que se asienta la creación y uso de bases de datos de cualquier índole. No obstante, no hemos mencionado a lo largo de los ejemplos presentados ningún dato de carácter espacial, a pesar de que sabemos bien que la información geográfica contiene tanto una componente temática como una espacial. Más aún, algunos de los atributos en los sencillos casos mostrados, como puede ser el atributo CIUDAD, son fácilmente asociables a elementos geográficos (por ejemplo, un punto que señale el centro de la ciudad o un polígono que recoja su contorno).

Evolución del uso de bases de datos en los SIG

Como acabamos de decir, los conceptos que hemos visto en las anteriores secciones representan una gran parte de la realidad actual en cuanto al manejo de datos (espaciales o no) dentro de un SIG. No obstante, el problema del acceso a los datos se ha solucionado de diversas formas a lo largo de la historia de los SIG, y encontramos en las aplicaciones SIG distintos enfoques a lo





largo del tiempo. Para concluir este capítulo veremos con algo más de detalle la evolución que ha seguido esta importante faceta de los SIG.

Primera generación. Ficheros

Los primeros programas, entre los cuales se han de incluir los primeros SIG, se caracterizaban en lo que al almacenamiento de datos respecta por una ausencia completa de cualquier tipo de almacenamiento estructurado. En estas aplicaciones, los datos no se veían como un elemento más dentro de un sistema, sino como una parte del propio *software* o, al menos, como algo asociado únicamente a un producto particular. Así, encontramos en esta época como práctica habitual el uso de ficheros con formatos cerrados, pensados para ser leídos y escritos casi de forma exclusiva por la aplicación particular que ha de consumirlos, limitando así el uso compartido y el alcance de los datos a otros ámbitos distintos.

Integrar en el SIG otros datos distintos a aquellos para los que la aplicación se había diseñado no era sencillo, ya que existía una vinculación muy directa entre software y datos. Asimismo, las funcionalidades del software eran también específicas para esos datos, y todas ellas se implementaban directamente en la aplicación. Al no existir un SGBD que se encargara de gestionar las operaciones, era el propio SIG quien debía ser responsable de las funcionalidades de acceso o edición. Otras funcionalidades típicas de un SGBD, sin embargo, no aparecían en estos primeros SIG, ya que no eran necesarias. Por ejemplo, velar por la integridad de los datos en operaciones concurrentes de varios usuarios no era necesario si la aplicación en sí no estaba diseñada para permitir este acceso múltiple.

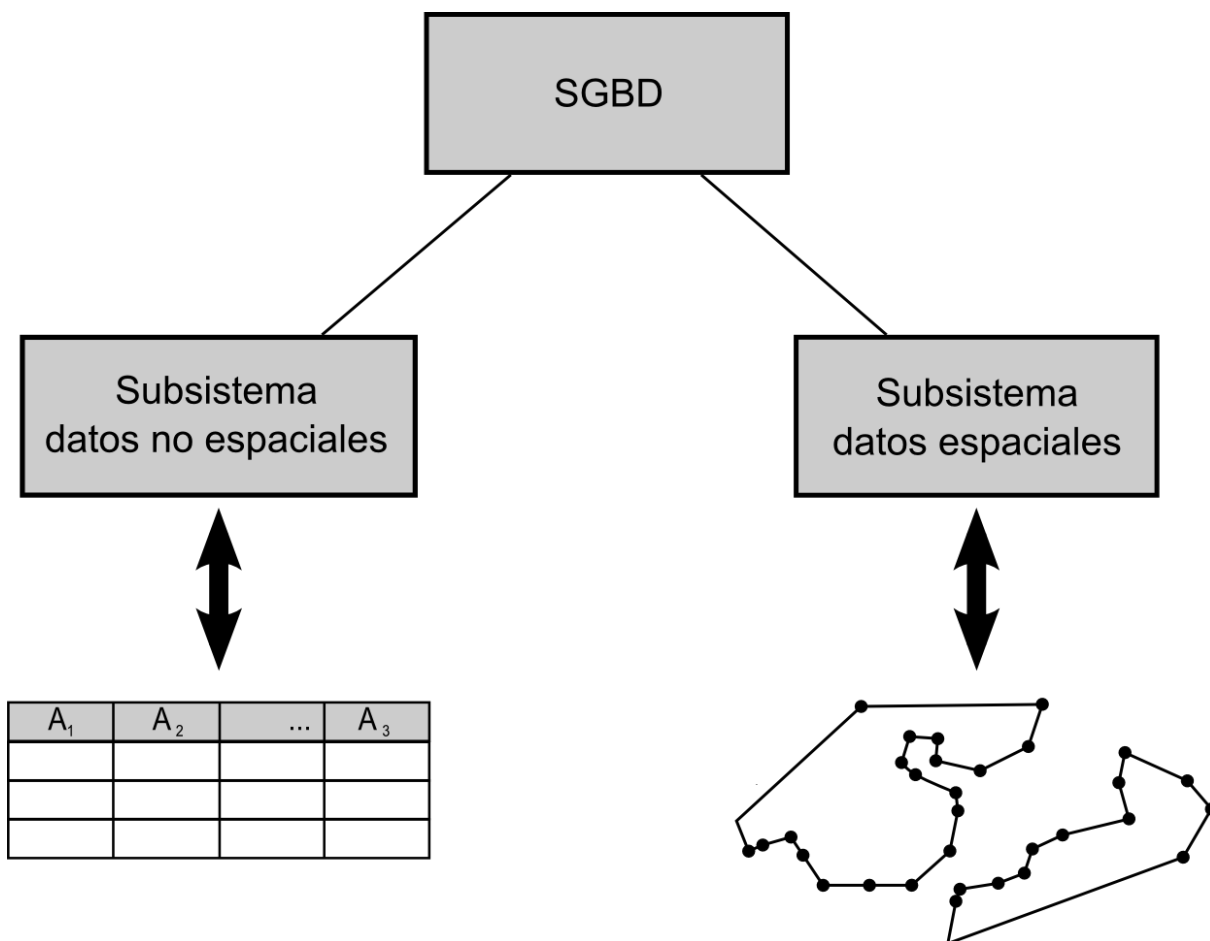
Las únicas ventajas que pueden encontrarse en este enfoque son las relacionadas con el rendimiento, que podía en ciertos casos ser mayor que el esperable en caso de utilizar un SGBD para canalizar el trabajo con los datos. Esto es así debido a que la propia especificidad de la aplicación permitía una optimización «a medida», aunque todo ello a cambio de sacrificar la flexibilidad de la aplicación, su escalabilidad, o la posibilidad de que los datos empleados pudieran ser utilizados de forma sencilla para alimentar otras aplicaciones.

Segunda generación. Bases de datos relacionales

Una vez que las bases de datos comienzan a tomar su papel en el panorama del software, no tardan en encontrar su camino dentro de las aplicaciones SIG. Las bases de datos relacionales, que como ya sabemos son las más empleadas, comienzan a ser utilizadas también para gestionar los datos espaciales con los que se trabaja en un SIG. A partir de esta segunda generación, se empiezan a adaptar las características del modelo relacional y de las bases de datos que lo implementan a las particularidades de los datos espaciales. Las dificultades que aparecen debido



a la inherente complejidad de la componente espacial hacen que surjan diversas alternativas para su manejo. Las más reseñables de entre ellas son el uso de una arquitectura dual en la que únicamente la componente temática se gestiona mediante una base de datos y el uso de una arquitectura en capas en el que se da un pleno almacenamiento de la información espacial en la base de datos.



Arquitectura dual con subsistemas distintos para el manejo de datos espaciales y no espaciales

Arquitectura en capas

La otra forma de aprovechar una base de datos relacional para su uso dentro de un SIG consiste en incorporar toda la información dentro de la base de datos, incluyendo la de corte espacial, buscando la manera más adecuada de llevar esto a cabo pese a las limitaciones que la propia base de datos presenta en este caso. Asumiendo que una base de datos relacional en su concepto tradicional no está diseñada para contener objetos complejos tales como geometrías o imágenes, y que, especialmente, el SGBD correspondiente no presenta las mismas funcionalidades y la



misma potencia en el manejo de este tipo de datos que en el de tipos de dato estándar (valores numéricos, cadenas de texto, fechas, etc.), es posible, sin embargo, plantear soluciones que permitan llevar toda la información de un SIG a una base de datos y poder gestionarla por completo a través de un SGBD, con las ventajas que ello conlleva, y que ya conocemos.

Dos son las alternativas existentes: un almacenamiento *transparente* y un almacenamiento *opaco*. Ambos se distinguen en la forma de almacenar la información y también las operaciones sobre los datos, que vienen condicionadas por la estrategia empleada para el almacenamiento de estos.

Tercera generación. Bases de datos extensibles

En la actualidad, las bases de datos presentan arquitecturas extensibles que permiten adaptarse a la naturaleza de los datos con los que trabajan, de tal forma que enfocan sus funcionalidades hacia la tipología particular que se maneje. Los tipos de datos clásicos conviven con nuevos tipos de datos que pueden ser definidos, y con operaciones específicas para estos.

Un caso particular de estas bases de datos extensibles son las *bases de datos orientadas a objetos*, que ya fueron comentadas al presentar los distintos modelos de bases de datos. A pesar de que este tipo de bases de datos no ocupan una porción significativa en el mercado global de las bases de datos, y son las de tipo relacional las más extendidas, existen algunos sectores en los que han logrado una mayor penetración, entre ellos el del SIG. Por sus características, las bases de datos orientadas a objetos resultan ventajosas para el manejo de datos complejos que no puedan recogerse con facilidad utilizando los tipos de datos clásicos de una base de datos relacional. En este grupo pueden incluirse las primitivas geométricas que utilizamos en un SIG para recoger la componente espacial de un dato espacial, las cuales resulta más adecuado considerar como objetos de un tipo dado (punto, línea o polígono), aprovechando así las ventajas que un enfoque orientado a objetos proporciona.

Los SGBD actuales presentan en su gran mayoría extensiones dedicadas al manejo de datos espaciales, los cuales contienen todo lo necesario para el manejo óptimo de estos, la realización de ciertas operaciones fundamentales y la optimización de las consultas y operaciones. Esta optimización es posible ya que el tipo de datos espacial está plenamente integrado en la base de datos y es considerado de la misma manera que cualquiera de los tipos de datos estándar como puede ser una cadena de texto o un valor numérico. La eficiencia que se obtiene de este modo es muy elevada.





Resumen

En este capítulo hemos visto los conceptos básicos sobre bases de datos. Una base de datos constituye un sistema que permite un manejo adecuado de los datos, garantizando la seguridad e integridad de estos y permitiendo el acceso a distintos usuarios de forma transparente. La base de datos está formada por los datos en sí, organizados de forma estructurada, mientras que las operaciones las provee el sistema gestor de base de datos (SGBD).

Existen diversos modelos para el almacenamiento de datos, siendo el modelo relacional el más habitual en la actualidad. En el modelo relacional la información se organiza en tablas relacionadas entre sí. Cada fila de una base de datos conforma una tupla, que contiene la información correspondiente a una entidad dada.

El diseño de la base de datos es de gran importancia, y conlleva el diseño de un modelo conceptual, el diseño de un modelo físico, la implementación y el mantenimiento. Herramientas como los diagramas E-R son de ayuda en las fases de diseño, cuyo principal objetivo es crear una estructura de la base de datos que facilite la interpretación de la información contenida y permita el máximo aprovechamiento de esta.

En lo que a los SIG respecta, las bases de datos se han ido incorporando paulatinamente a la gestión de los datos espaciales. Partiendo de una situación inicial en la que no se empleaban sistemas gestores de bases de datos, estos han ido integrándose en los SIG de diversas formas. En la actualidad, se emplean bases de datos relacionales adaptadas para poder almacenar datos espaciales y realizar operaciones sobre ellos. Los SGBD extensibles representan la última tendencia, y en ellos puede integrarse plenamente la información geográfica de forma óptima.





Base de Consulta

TÍTULO	AUTOR	EDICIÓN	AÑO	IDIOMA	EDITORIAL

B. Base práctica con ilustraciones

Contenido

- 1 Sobre MySQL
- 2 Crear una base de datos MySQL desde cPanel
 - 2.1 Crear un usuario MySQL
 - 2.2 Añadir privilegios al usuario creado
- 3 Crear una base de datos MySQL desde phpMyadmin
- 4 Crear una base de datos MySQL desde Linux
- 5 Conclusión

Sobre MySQL

Y como siempre, antes de comenzar con el plato principal, una entrada, porque no podemos hablar sobre cómo crear bases de datos en MySQL y asignar usuarios si primero no sabemos qué es MySQL.

Por supuesto es posible que muchos ya lo sepan, pero para quienes no y para quienes no lo tengan fresco en la memoria, no viene nada mal tocar este asunto.

MySQL es y probablemente seguirá siendo durante un largo tiempo uno de los sistemas de bases de datos más populares del mundo del hosting, de hecho hoy en día es el más usado de todos, con una cuota de mercado del 48% según Datanyze.

En este sentido no tiene competidor directo, ya que el segundo lugar lo ocupa el SQL Server de Microsoft con una cuota de apenas un 10%.

Ampliamente usado en desarrollo web, MySQL actualmente está en manos de Oracle (también poseedores de Oracle Database) y cuenta con dos tipos de licencia: una comercial y una pública general, es decir, hay un MySQL de pago y uno gratuito, aunque obviamente este último es el que realmente se lleva toda la popularidad.



Como decíamos, la aplicación principal de MySQL es en sitios web, y muchos CMS populares son compatibles con este gestor de bases de datos, incluyendo por ejemplo a WordPress, Joomla, Drupal, entre otros.

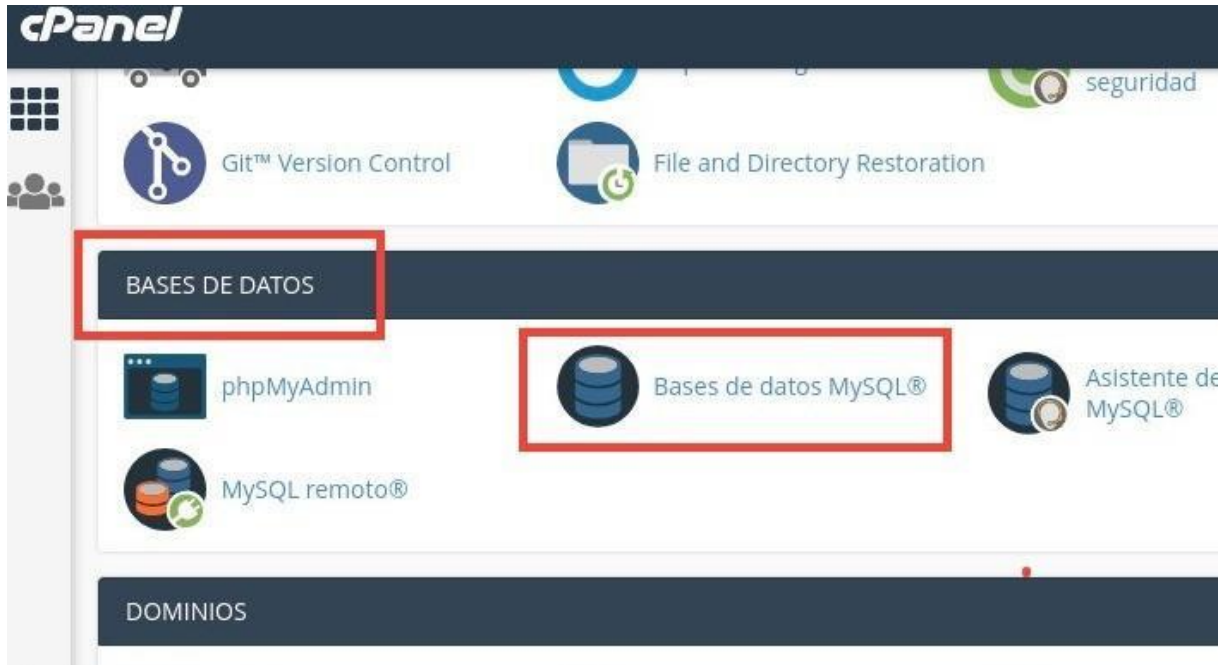
Un gran punto a su favor es el hecho de ser multiplataforma, sin mencionar que se puede utilizar de maravilla en entornos que incluyan Apache o Nginx con PHP o PHP-FPM.

En pocas palabras, MySQL es un sistema que se adapta muy bien a las situaciones, es seguro, rápido y muy confiable, de hecho si no lo fuera es posible que gigantes como Google, Facebook o Twitter dejen de usarlo sin pensarlo dos veces.

Crear una base de datos MySQL desde cPanel

En nuestros planes de hosting en Infranetworking integramos un estupendo panel de control llamado cPanel. Entre varias de sus funciones, cPanel nos brinda la posibilidad de crear bases de datos rápidamente mediante una interfaz gráfica, ideal para quienes no les gusta mucho el trabajo por medio de la consola.

Para empezar con este proceso, lo primero que debemos hacer es, por supuesto, ingresar a nuestro panel de control cPanel. Una vez dentro, la tarea es buscar el apartado de Bases de Datos y allí hacer click en la opción de Bases de Datos MySQL, tal como vemos en la captura de aquí abajo.



Ahora estaremos en un área nueva de nuestro panel, y aquí es donde realizaremos la mayoría de las gestiones referentes a crear/eliminar bases de datos y usuarios de MySQL.

Para añadir nuestra base de datos MySQL debemos comenzar buscando donde dice «Crear una nueva base de datos», y allí colocaremos el nombre que tendrá nuestra bd.



Las bases de datos en cPanel llevan siempre un nombre con esta estructura: usuario de cpanel_bd, siendo «usuariodecpanel» el usuario de cPanel obviamente y «bd» el identificador asignado a la base de datos. En el caso el ejemplo, la base de datos se llama «cuentate_prueba1». Click en el botón de Crear para continuar.



Ahora que nuestra base de datos ha sido creada, toca asignarle un usuario, ya que el modo de operar en MySQL es ese: cualquier base de datos debe tener un usuario asignado con suficientes privilegios para poder usarla. Veamos a continuación cómo es dicho proceso.

Crear un usuario MySQL

Dentro de la sección de Bases de Datos MySQL donde creamos nuestra bd en el paso anterior, vamos a buscar el apartado de «Usuarios MySQL», y allí veremos una opción para añadir un usuario y asignarle una contraseña.

Al igual que en el caso de las bases de datos, la estructura es usuario de cpanel_nombre, es decir que en nuestro ejemplo el usuario de MySQL se llamaría cuentate_usr01.

Aquí mismo tendremos la opción para asignar una contraseña a nuestro usuario de MySQL, y por último solo hay que hacer click en el botón de Crear usuario.





Usuarios MySQL

Añadir nuevo usuario

Nombre de usuario
cuentate_ usr01

Contraseña
.....

Contraseña (nuevamente)
.....

Seguridad ⓘ
Muy segura (100/100)

Generador de contraseñas

Crear usuario

Ahora que ya tenemos creada una base de datos MySQL y un usuario, es el turno de la última parte de este tutorial: asignar los privilegios necesarios al usuario para que pueda interactuar con nuestra base de datos.

Añadir privilegios al usuario creado

Desde el lugar de siempre (cPanel > Bases de Datos MySQL), vamos a buscar donde diga «Añadir usuario a la base de datos», lo cual básicamente es una opción para dar privilegios al usuario sobre la bd. Simplemente debemos seleccionar el usuario primero y después la base de datos y hacemos click en Añadir.

Añadir usuario a la base de datos

Usuario
cuentate_usr01

Base de datos
cuentate_prueba1

Añadir

Al añadir en el paso anterior estaremos en una nueva área donde dice «Administrar privilegios de usuario», y se indicará el usuario y la base que hemos seleccionado. Aquí tenemos la opción de indicar al sistema los privilegios que tendrá el usuario sobre la base de datos.





Los privilegios que se deben asignar dependen en gran parte del rol que tendrá el usuario en la bd, y a veces también del rol de la base de datos en sí. ¿Y quién define eso? Un programador por supuesto, así que si tienes dudas sobre cuáles privilegios asignar te recomendamos consultar con el programador de tu sitio web.

Aun así, y hablando en términos generales, en el 90% de los casos la opción elegida es simplemente «Todos los privilegios», lo cual dotará al usuario de MySQL de todos los privilegios necesarios para hacer todo lo que se desee sobre la bd en cuestión. Para terminar simplemente hacemos clic en «Hacer cambios».



Y ya está, ahora lo único que resta es comenzar a trabajar con tu nueva base de datos MySQL.

Por cierto, si quieres ver un listado de las bases de datos y los usuarios que están asignados a cada una, simplemente ve a la sección de siempre (Bases de Datos MySQL) y busca donde diga «Bases de datos actuales», allí podrás ver el listado en detalle e incluso se pueden ejecutar un par de acciones como cambiar el nombre de la db o eliminarla.





Crear una base de datos MySQL desde phpMyadmin

Si necesitamos crear una base de datos MySQL por medio de phpMyAdmin, lo primero que debemos hacer es ingresar al phpMyAdmin que tengamos disponible. Cabe mencionar que este método no es usado en servidores cPanel, en un servidor cPanel se recomienda usar el método descrito anteriormente.

Dentro de phpMyAdmin, primero que nada hacemos click en donde dice Databases o Bases de Datos.



A continuación, veremos una opción que dice **Create Database** o **Crear Base de Datos**, y debajo un campo de texto. En dichos campos de texto debemos introducir el nombre de la base de datos que queremos crear, seleccionar la Colación y posteriormente le damos al botón Create. Para la Colación es importante coordinar con el programador del sitio cuál seleccionar, por supuesto en caso de que no seamos el programador.



Ahora que está creada nuestra BD vamos a **Users > Add User**. Desde aquí añadiremos un nuevo usuario, para lo cual debemos rellenar varios campos como en la captura de abajo.





Add user

Login Information

User name: Use text field: minuevousuario

Host: Use text field: localhost

Password: Use text field:

Re-type:

Generate password: Generate

En la misma sección, pero más abajo tenemos la posibilidad de seleccionar los privilegios que tendrá nuestro usuario. Finalmente hacemos click en «Go» y el usuario será creado.

Global privileges (Check All /Uncheck All)

Note: MySQL privilege names are expressed in English

Data

- ☒ SELECT
- ☐ INSERT
- ☐ UPDATE
- ☒ DELETE
- ☐ FILE

Structure

- ☐ CREATE
- ☐ ALTER
- ☐ INDEX
- ☐ DROP
- ☐ CREATE TEMPORARY TABLES
- ☒ SHOW VIEW
- ☐ CREATE ROUTINE
- ☐ ALTER ROUTINE
- ☒ EXECUTE
- ☐ CREATE VIEW
- ☐ EVENT
- ☐ TRIGGER

Administration

- ☐ GRANT
- ☐ SUPER
- ☒ PROCESS
- ☐ RELOAD
- ☐ SHUTDOWN
- ☒ SHOW DATABASES
- ☐ LOCK TABLES
- ☐ REFERENCES
- ☒ REPLICATION CLIENT
- ☐ REPLICATION SLAVE
- ☐ CREATE USER

Resource limits

Note: Setting these options to 0 (zero) removes the limit.

MAX QUERIES PER HOUR 0

MAX UPDATES PER HOUR 0

MAX CONNECTIONS PER HOUR 0

MAX USER_CONNECTIONS 0

Para terminar vamos a darle permisos sobre la bd que deseamos. Vamos a Usuarios, buscamos nuestro usuario y click en **Editar Privilegios**. Aquí veremos una opción que dice «Database-specific privileges», y es allí donde debemos colocar la base de datos asociada a nuestro usuario. Completamos el campo y hacemos click en Go para terminar la operación.





Crear una base de datos MySQL desde Linux

Esta es la forma más rápida de crear una base de datos y asignarle un usuario, pues solo necesitamos correr tres líneas desde la consola de MySQL.

Primero conectarnos como root:

```
mysql -u root -p
```

Luego de eso debemos correr lo siguiente:

```
create database nombre_bd;
```

```
GRANT ALL PRIVILEGES ON nombre_bd.* TO 'usuario'@'localhost' IDENTIFIED BY 'contraseña';
```

```
flush privileges;
```

Primero se crea la base de datos, después el usuario y los privilegios y finalmente se refrescan los privilegios. Tres líneas es todo lo necesario. Por supuesto en dichas líneas debemos reemplazar algunos datos que hemos colocado a modo de ejemplo:

- nombre_bd: será el nombre real de la base de datos.
- usuario: será el usuario asociado a la bd.
- localhost: si el host es local entonces lo dejamos como localhost, de lo contrario indicamos un hostname o una IP.
- contraseña: lo reemplazamos por la contraseña del usuario de nuestra bd. Recuerda usar una contraseña fuerte.





4. ESTRATEGIAS DE APRENDIZAJE

ESTRATEGIA DE APRENDIZAJE 1: Análisis y Planeación

Descripción:

Se aplicará una metodología eminentemente práctica y activa, seleccionando métodos, procesos, técnicas y estrategias innovadoras entre el docente y estudiante, a través del trabajo colaborativo, cooperativo, la investigación de campo en sinergia y bibliográfica, la técnica bajo la orientación, guía y se utilizará el Aprendizaje Basado en Problemas como metodología activa.

Para el desarrollo de la Unidad Didáctica se aplicará los siguientes procedimientos didácticos:

- Clase teóricas: con exposición del docente y la participación activa de los alumnos
- Prácticas: se irán resolviendo aplicaciones mediante desarrollo de prácticas dirigidas y calificadas
- Intervenciones orales: con presentaciones y exposiciones de los alumnos en equipo e individual
- Proyecto: los alumnos formarán equipos de trabajo; y con la asesoría del docente desarrollarán su Proyecto de Investigación correspondiente a esta unidad didáctica, el cual será sustentado en la semana 17

Ambiente(s) requerido:

Plataforma Moddle

Plataforma Teams

Procesador de texto

Videos

Material (es) requerido:

Aplicaciones: NetBeans, Eclipse

Sistemas Operativos: Windows y Linux

Acceso a Internet

Docente:

Ingeniero en Sistemas Informáticos y Magíster en Informática Empresarial, Experto en desarrollo de aplicaciones Orientadas a Objetos

5. ACTIVIDADES

- Las Actividades se basarán en Clases Magistrales, Organizadores gráficos, Investigación y análisis de información mediante las TICs
- Análisis y comprensión del material de estudio





- Consulta bibliográfica Preparar
- Exposición Ejemplificación Realizar tarea

6. EVIDENCIAS Y EVALUACIÓN

Tipo de Evidencia	Descripción
De conocimiento:	Ensayo expositivo grupal de lecturas Definición del tema de investigación
Desempeño:	Trabajo grupal presentación del trabajo sobre la implementación de aplicaciones Orientadas a Objetos
De Producto:	Se realizará una producción técnica relacionada con el diseño y la implementación de una aplicación móvil para la línea de investigación de Gerencia Empresarial y Pública ya que existe limitada visión empresarial del mercado global y de estrategias de marketing digital la misma que estará enmarcada en la educación como estrategia para el fortalecimiento de las empresas de economía popular y solidaria.
Criterios de Evaluación (Mínimo 5 Actividades por asignatura)	La clase está ponderada sobre 10 puntos, y su distribución es de la siguiente manera: Nota 1 (30%) 1 exposición 1 debate Nota 2 (30 %) Exposición de ejercicios de los recursos académicos Nota 3 (40 %) 3 evaluación por parcial

Compilado por: Ing. Giovanni Cholca MSc.	Revisado por: (Dirección investigación)	Aprobado por: (Rectorado)

