

Unidad Académica: FACULTAD DE INGENIERÍA
Asignatura: BASE DE DATOS APLICADA
Año académico: 2025 **Régimen:** SEMESTRAL
Horas Totales: 60 **Carácter:** TEÓRICA/PRÁCTICA

Guía de Trabajos Prácticos

UNIDAD 1

Unidad 1: Instalación y configuración de Sistemas de Gestión de Bases de Datos (SGBD).

Sistemas de Gestión de Base de Datos. Selección. Requisitos del Sistema. Configuración inicial. Seguridad. Bases de datos de Usuario. Bases de Datos del Sistema. Monitorización y mantenimiento. Pruebas y ajustes. Arquitectura Cliente Servidor. Conexión al Sistema de gestión de Base de Datos. Conexión de aplicaciones a base de datos locales y en la nube.

Trabajo Práctico 1: Instalación y Configuración de Sistemas de Gestión de Base de Datos

Objetivo: Comprender y aplicar los conceptos fundamentales de los Sistemas de Gestión de Bases de Datos (SGBD) con enfoque en MySQL

A. Introducción a los SGBD y Configuración Inicial de MySQL

Objetivos:

- Comprender el rol y la importancia de un SGBD.
- Conocer los criterios para la selección de un SGBD.
- Realizar una instalación básica de MySQL y su configuración inicial.
- Familiarizarse con las herramientas de administración de MySQL.

1. Definición y Rol de un SGBD:

- ¿Qué es un Sistema de Gestión de Bases de Datos (SGBD) y cuál es su función principal en un sistema informático?
- Mencionar al menos 3 ventajas de utilizar un SGBD en lugar de almacenar datos en archivos planos.

2. Selección de un SGBD:

- Enumere y describa brevemente al menos 5 criterios clave a considerar al seleccionar un SGBD para un proyecto.

3. Requisitos del Sistema:

- Investigar los requisitos mínimos de hardware (RAM, CPU, espacio en disco) y software (Sistema Operativo) para instalar MySQL Server en un entorno de desarrollo.

4. Instalación de MySQL:

- Instalar MySQL Community Server en su equipo.
- Configurar usuario `root` y su contraseña.
- Acceder a la línea de comandos de MySQL (mysql client) utilizando el siguiente comando:
- `mysql -u root -p`
- Desde la línea de comandos, verificar la versión de MySQL usando:
`SELECT VERSION();`

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

Salir de la línea de comandos (**exit**).

2. Acceso y Configuración Inicial:

- Instalar MySQL Workbench y crear una nueva conexión a la instancia local de MySQL utilizando el usuario **root** y la contraseña establecida.
- Explique brevemente qué es y para qué sirve MySQL Workbench.
- En MySQL Workbench, "Server Status", identificar la versión de MySQL instalada.

B. Seguridad en MySQL y Bases de Datos del Sistema/Usuario

Objetivos:

- Comprender la importancia de la seguridad en un SGBD.
- Administrar usuarios y privilegios en MySQL.
- Diferenciar entre bases de datos del sistema y de usuario.
- Explorar las bases de datos del sistema en MySQL.

1. Seguridad en Bases de Datos:

- Justificar por qué es importante implementar medidas de seguridad en un SGBD. Mencionar al menos 3 riesgos de seguridad comunes en bases de datos.
- Explicar el concepto de "privilegios" en MySQL.
- ¿Por qué es una buena práctica asignar el mínimo privilegio necesario a los usuarios?

2. Bases de Datos del Sistema vs. Bases de Datos de Usuario:

- ¿Qué son las bases de datos del sistema en MySQL? ¿Cuál es su propósito principal? Nombrar algunas de las bases de datos del sistema.
- Definir qué son las bases de datos de usuario.

1. Gestión de Usuarios y Privilegios:

- Conectarse a MySQL utilizando algún cliente (mysql client, Workbench como **root**.)
- Crear un nuevo usuario llamado **admin_db** con una contraseña.
- Asignar al usuario creado privilegios para crear, modificar y eliminar bases de datos (**CREATE, ALTER, DROP** on *.*).
- **CREATE USER 'admin_db'@'localhost' IDENTIFIED BY 'su_contraseña_segura';**
- **GRANT CREATE, ALTER, DROP ON *.* TO 'admin_db'@'localhost';**
- **FLUSH PRIVILEGES;**
- Crear otro usuario llamado **app_user** con una contraseña. Inicialmente, no asignar ningún privilegio a este usuario.
- **CREATE USER 'app_user'@'localhost' IDENTIFIED BY 'otra_contraseña_segura';**
- **FLUSH PRIVILEGES;**
- Intentar acceder a MySQL Workbench con el usuario **app_user** y verificar:
- ¿Qué bases de datos puede ver?



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen: SEMESTRAL

Horas Totales: 60

Carácter: TEÓRICA/PRÁCTICA

- ¿Puede crear una nueva base de datos? Explicar.
- Conectarse nuevamente como **root**.
- Otorgar al usuario **app_user** privilegios para seleccionar, insertar, actualizar y eliminar datos (**SELECT, INSERT, UPDATE, DELETE**) en **cualquier tabla de una base de datos específica (ej: mi_db)**
- **GRANT SELECT, INSERT, UPDATE, DELETE ON mi_db.* TO 'app_user'@'localhost';**
- **FLUSH PRIVILEGES;**
- Si **mi_db** no existe, ¿el comando anterior da error?
- Intente nuevamente acceder a MySQL Workbench con el usuario **app_user**.
- ¿Puede este usuario crear la base de datos **mi_db**? ¿Por qué sí o por qué no?

2. Exploración de Bases de Datos del Sistema:

- Conectarse a MySQL Workbench como **root**.
- Explorar las bases de datos **mysql, information_schema** y **performance_schema**.
- En la base de datos **mysql**, buscar la tabla **user**. ¿Qué información contiene esta tabla?
- En la base de datos **information_schema**, explorar la tabla **SCHEMATA**. ¿Qué información proporciona sobre las bases de datos existentes en su sistema MySQL?
- En la base de datos **performance_schema**, explorar algunas tablas **threads** y **events_statements_summary_by_digest**. ¿Qué tipo de información se almacena en estas tablas?

C. Monitorización y Mantenimiento de MySQL

Objetivos:

- Comprender la importancia de la monitorización y el mantenimiento regular de un SGBD.
- Familiarizarse con herramientas y comandos para la monitorización básica de MySQL.
- Realizar tareas básicas de mantenimiento como backups y optimización.

1. Monitorización:

- ¿Por qué es fundamental monitorizar el rendimiento y el estado de un servidor MySQL? Mencionar al menos 3 métricas importantes que deberían ser monitoreadas.
- ¿Qué herramientas o comandos nativos de MySQL se pueden utilizar para obtener información sobre el estado del servidor?

2. Mantenimiento:

- Explicar la importancia de los backups regulares en un entorno de base de datos. ¿Qué estrategias de backup existen?
- ¿Qué es la optimización de bases de datos y por qué es importante para el rendimiento? Mencionar 2 técnicas de optimización.



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

1. Monitorización Básica:

- Conectarse a la línea de comandos de MySQL como **root**.
- Ejecutar el comando **SHOW PROCESSLIST**; ¿Qué información muestra y cuál es su utilidad?
- Ejecutar **SHOW STATUS LIKE 'Threads_connected%'**; ¿Qué representa este valor?
- Ejecutar **SHOW GLOBAL STATUS LIKE 'Com_select%'**; ¿Qué muestra este comando?
- En MySQL Workbench, "Server Status", observar la información en tiempo real. Identificar algunos de los parámetros visualizados en la línea de comandos.

2. Mantenimiento: Backup y Restauración (Simulado):

- Crear una nueva base de datos llamada **db_backup_test**.
- Crear una tabla dentro de **db_backup_test** con algunos datos de prueba.
- **CREATE TABLE db_backup_test.productos (**
- **id INT PRIMARY KEY AUTO_INCREMENT,**
- **nombre VARCHAR(100),**
- **precio DECIMAL(10, 2)**
- **);**
- **INSERT INTO db_backup_test.productos (nombre, precio) VALUES ('Laptop', 1200.00), ('Teclado', 75.50);**
- Desde la línea de comandos realizar un backup de la base de datos **db_backup_test** utilizando **mysqldump**.
- **mysqldump -u root -p db_backup_test > db_backup_test_backup.sql**
- Simula situación de "desastre": eliminar la base de datos **db_backup_test**.
- **DROP DATABASE db_backup_test;**
- Restaurar la base de datos desde el archivo de backup utilizando el siguiente comando:
- **mysql -u root -p < db_backup_test_backup.sql**
- Verificar que la base de datos **db_backup_test** ha sido restaurada correctamente.

3. Mantenimiento: Optimización (Conceptual):

- Investigar el comando **OPTIMIZE TABLE** en MySQL. ¿Para qué se utiliza y en qué situaciones sería beneficioso?
- ¿Por qué es importante el uso de índices en la optimización de consultas? Explicar brevemente.

D. Pruebas y Ajustes, Arquitectura Cliente-Servidor y Conexión de Aplicaciones

Objetivos:

- Comprender la importancia de las pruebas y ajustes en bases de datos.
- Entender la arquitectura cliente-servidor en el contexto de bases de datos.
- Conectar una aplicación simple a una base de datos MySQL local.
- Discutir los desafíos y consideraciones de conexión a bases de datos en la nube.



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA
Asignatura: BASE DE DATOS APLICADA
Año académico: 2025 **Régimen:** SEMESTRAL
Horas Totales: 60 **Carácter:** TEÓRICA/PRÁCTICA

1. Pruebas y Ajustes:

- ¿Por qué son importantes las fases de pruebas y ajustes en el ciclo de vida de una base de datos?
- Mencionar al menos 3 tipos de pruebas que se pueden realizar en una base de datos.

2. Arquitectura Cliente-Servidor:

- Explicar el concepto de arquitectura Cliente-Servidor en el contexto de los SGBD. Identificar el "cliente" y el "servidor" en esta arquitectura.
- ¿Cuáles son las ventajas de una arquitectura cliente-servidor para las bases de datos?

3. Conexión a Bases de Datos:

- ¿Qué es una cadena de conexión (connection string) y qué información suele contener para conectarse a una base de datos MySQL?
- ¿Cuáles son las principales diferencias y consideraciones al conectar una aplicación a una base de datos MySQL local versus una base de datos MySQL en la nube (ej: Amazon RDS, Google Cloud SQL)?

1. Simulación de Pruebas de Rendimiento:

- Suponiendo una tabla con millones de registros. ¿Cómo se podría simular una "prueba de rendimiento" para evaluar la velocidad de una consulta **SELECT** compleja? ¿Qué métricas interesan?
- ¿Qué ajustes podría considerar para mejorar el rendimiento de esa consulta?

2. Conexión de Aplicación a MySQL Local

○ Crear una Base de Datos y Tabla de Prueba:

- Crear una nueva base de datos llamada **mi_db**.
- Cree una tabla llamada **usuarios** con los siguientes campos

id (PRIMARY KEY, AUTO_INCREMENT),
nombre (VARCHAR),
email (VARCHAR único).

```
CREATE DATABASE mi_app_db;  
USE mi_app_db;  
CREATE TABLE usuarios (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nombre VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE NOT NULL  
);
```

- **Escribir un Script de Conexión:**



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

- Escriba un script que realice las siguientes operaciones:
 1. **Conectarse** a la base de datos **mi_db** utilizando el usuario **app_user** o **root**.
 2. **Insertar** al menos 2 registros en la tabla **usuarios**.
 3. **Consultar** todos los registros de la tabla **usuarios** y mostrarlos por consola.
 4. **Cerrar** la conexión a la base de datos.

3. Conexión a la Nube (Investigación y Debate):

- Investigar al menos un servicio de base de datos MySQL gestionado en la nube (ej: Amazon RDS for MySQL, Google Cloud SQL for MySQL, Azure Database for MySQL).
- Enumerar al menos 3 ventajas y 2 desventajas de utilizar una base de datos en la nube en comparación con una base de datos local.
- ¿Qué consideraciones de seguridad adicionales se deben tener en cuenta al conectar una aplicación a una base de datos en la nube?

Unidad Académica: FACULTAD DE INGENIERÍA
Asignatura: BASE DE DATOS APLICADA
Año académico: 2025 **Régimen:** SEMESTRAL
Horas Totales: 60 **Carácter:** TEÓRICA/PRÁCTICA

UNIDAD 2

Unidad 2: Administración y gestión de bases de datos.

Creación y configuración de Base de datos. Gestión de Usuarios y permisos. Monitorización de rendimiento. Mantenimiento de Base de Datos. Auditoría y registro de eventos. Seguridad en base de datos. Backup. Restauración. Estrategias. Estrategias de Respaldo y Restauración. Plan de respaldo. Pruebas de restauración. Registro de respaldo. Plan de recuperación ante incidentes. Medios de almacenamiento. Redundancia. Redundancia de datos. Redundancia de hardware. Estrategias de caché en bases de datos. Caché de datos. Caché de consulta.

Trabajo Práctico 2: Administración y gestión de bases de datos

A. Creación, Configuración y Gestión de Usuarios y Permisos

Objetivos:

- Crear y configurar bases de datos en MySQL.
- Administrar usuarios y asignar/revocar privilegios.
- Comprender el modelo de seguridad basado en privilegios de MySQL.

1. Creación y Configuración de Bases de Datos:

- ¿Cuál es la diferencia entre una "base de datos" y un "esquema" en MySQL?
- Mencione y explique brevemente al menos tres parámetros de configuración importantes que se pueden ajustar en el archivo `my.cnf` (o `my.ini`) de MySQL para influir en el rendimiento o comportamiento del servidor.

2. Gestión de Usuarios y Permisos:

- Describir la diferencia entre los privilegios `GRANT ALL PRIVILEGES` y `GRANT SELECT, INSERT, UPDATE, DELETE` sobre una base de datos específica. ¿Cuándo usaría cada uno?

1. Creación y Configuración Básica:

- Conectarse al servidor MySQL local como `root`.
- Crear una base de datos llamada `tp_db`.
- `CREATE DATABASE tp_aplicadas_db;`
- Dentro de `tp_db`, crear una tabla llamada `productos` con las siguientes columnas:
 - `id_producto` (INT, PRIMARY KEY, AUTO_INCREMENT)
 - `nombre` (VARCHAR(100), NOT NULL)
 - `precio` (DECIMAL(10, 2), NOT NULL)
 - `stock` (INT, DEFAULT 0)
- `USE tp_aplicadas_db;`
- `CREATE TABLE productos (`
- `id_producto INT PRIMARY KEY AUTO_INCREMENT,`
- `nombre VARCHAR(100) NOT NULL,`
- `precio DECIMAL(10, 2) NOT NULL,`



Unidad Académica: FACULTAD DE INGENIERÍA
Asignatura: BASE DE DATOS APLICADA
Año académico: 2025 **Régimen:** SEMESTRAL
Horas Totales: 60 **Carácter:** TEÓRICA/PRÁCTICA

- `stock INT DEFAULT 0`
- `);`
- Inserta 3 registros de ejemplo en la tabla `productos`.
- `INSERT INTO productos (nombre, precio, stock) VALUES`
- `('Laptop Gamer', 1500.00, 10),`
- `('Mouse Inalámbrico', 25.99, 50),`
- `('Teclado Mecánico', 89.99, 20);`
-
- **Gestión de Usuarios y Permisos:**
- Crear un nuevo usuario llamado `dev_user` que solo pueda conectarse desde `localhost`.
- `CREATE USER 'dev_user'@'localhost' IDENTIFIED BY 'DevPass123!';`
- Asignar a `dev_user` los privilegios `SELECT`, `INSERT`, `UPDATE`, `DELETE` sobre la base de datos `tp_db`.
- `GRANT SELECT, INSERT, UPDATE, DELETE ON tp_db.* TO 'dev_user'@'localhost';`
- `FLUSH PRIVILEGES;`
- Crear otro usuario llamado `report_user` que también sólo pueda conectarse desde `localhost`.
- `CREATE USER 'report_user'@'localhost' IDENTIFIED BY 'ReportPass!';`
- Asignar a `report_user` únicamente el privilegio `SELECT` sobre la tabla `productos` dentro de `tp_db`.
- **Prueba de Permisos:**
 - Intentar conectarse a MySQL como `dev_user`.
 - `INSERTAR` un nuevo producto.
 - `ELIMINAR` un producto.
 - `CREAR` una nueva tabla en `tp_db`. Documentar resultados.
 - Intente conectarse a MySQL como `report_user`.
 - `SELECCIONAR` datos de `tp_db.productos`.
 - `INSERTAR` un nuevo producto en `tp_db.productos`.
 - `SELECCIONAR` datos de otra tabla (si existe) en `tp_db` o de una base de datos del sistema como `mysql.user`. Documentar sus resultados.

B. Monitorización de Rendimiento y Mantenimiento de Base de Datos

Objetivos:

- Identificar métricas clave de rendimiento en MySQL.
- Utilizar comandos y herramientas para monitorizar el estado del servidor.
- Realizar tareas básicas de mantenimiento para optimizar el rendimiento.

1. Monitorización de Rendimiento:

- Mencionar al menos 4 métricas de rendimiento importantes que un administrador de bases de datos debería monitorizar en MySQL y explique brevemente por qué son relevantes.



Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen: SEMESTRAL

Horas Totales: 60

Carácter: TEÓRICA/PRÁCTICA

- ¿Qué es el **EXPLAIN** de una consulta SQL y cómo ayuda a la optimización del rendimiento?

2. Mantenimiento de Base de Datos:

- ¿Por qué es importante realizar tareas de mantenimiento regular en una base de datos?
- Explicar la diferencia entre **OPTIMIZE TABLE** y **ANALYZE TABLE** en MySQL.
¿Cuándo se aplica cada uno de los comandos?

1. Monitorización Básica:

- Conectarse a la línea de comandos de MySQL como **root**.
- Ejecutar **SHOW PROCESSLIST**; Verificar información y cómo podría usarla para identificar problemas.
- Ejecutar los **SHOW STATUS LIKE 'Threads_connected'**; y **SHOW STATUS LIKE 'Questions'**; Explique qué representan estos valores.
- Ejecutar **SHOW GLOBAL STATUS LIKE 'Innodb_buffer_pool_reads'**; y **SHOW GLOBAL STATUS LIKE 'Innodb_buffer_pool_read_requests'**; ¿Qué indican estos dos valores sobre el uso del buffer pool de InnoDB?
- En MySQL Workbench, explorar la sección "Performance Dashboard" o "Server Status". Identifique al menos 3 métricas que considere útiles para la monitorización en tiempo real.

2. Análisis y Optimización de Consultas:

- En **tp_db.productos**, insertar registros de prueba (10,000 a 50,000) con datos aleatorios para **nombre**, **precio** y **stock**.
- Ejecutar la siguiente consulta y observe su tiempo de ejecución:
SELECT * FROM productos WHERE nombre LIKE 'Producto%';
- Ejecutar **EXPLAIN** para la misma consulta:
EXPLAIN SELECT * FROM productos WHERE nombre LIKE 'Producto%';
- Analizar la salida de **EXPLAIN**. Verificar información de cómo MySQL ejecuta la consulta.
- Crear un índice en la columna **nombre**:
CREATE INDEX idx_nombre ON productos (nombre);
- Ejecutar la consulta
SELECT * FROM productos WHERE nombre LIKE 'Producto%';
- Ejecutar **EXPLAIN** para la misma consulta.
- Comparar los tiempos de ejecución y la salida de **EXPLAIN** antes y después de crear el índice. ¿Qué diferencias se observan y por qué?

3. Mantenimiento de Tablas:

- Ejecutar **OPTIMIZE TABLE productos**; en **tp_db**. ¿Qué mensaje obtiene y cuándo es útil este comando?
- Ejecutar **ANALYZE TABLE productos**; ¿Cuál es el propósito de este comando?

C. Auditoría, Registro de Eventos y Seguridad en Base de Datos



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

Objetivos:

- Comprender la importancia de la auditoría y el registro de eventos en bases de datos.
- Familiarizarse con los diferentes tipos de logs de MySQL.
- Aplicar conceptos de seguridad avanzada.

1. Auditoría y Registro de Eventos:

- Verificar la diferencia entre
- **General Query Log**, **Error Log** y **Slow Query Log** en MySQL
- ¿Qué es el **Binary Log** (Binlog) y cuál es su importancia principal en la administración de bases de datos MySQL?

2. Seguridad en Base de Datos:

- Mencione y explique al menos 3 prácticas de seguridad adicionales (más allá de la gestión de usuarios y privilegios) que se deben implementar para proteger una base de datos MySQL.
- ¿Qué es la inyección SQL y cómo se puede prevenir en el desarrollo de aplicaciones?

1. Configuración y Exploración de Logs:

- Importante: Modificar el archivo de configuración de MySQL puede requerir reiniciar el servicio. Asegúrese de tener permisos y de entender los pasos.
- Localizar el archivo de configuración de MySQL (**my.cnf** en Linux/macOS, **my.ini** en Windows).
- Habilitación por archivo:
 - Habilitar **General Query Log** agregando o modificando las siguientes líneas en la sección **[mysqld]**:
 - **general_log = 1**
 - **general_log_file = /var/log/mysql/mysql.log** # O la ruta adecuada para su SO
- Habilitar en tiempo de ejecución:
- Habilitar con
 - **SET GLOBAL general_log = 'ON';**
 - **SET GLOBAL general_log_file = '/path/to/mysql.log';**

¿Cuál es la diferencia entre ambos métodos?

- Reiniciar el servicio MySQL para tomar modificaciones del archivo de configuración.
- Conectarse a MySQL y realizar algunas operaciones.
- Localizar el archivo **general_log_file** y abrir con un editor de texto. Verificar las entradas generadas por las operaciones realizadas en el punto anterior.
- Deshabilitar **General Query Log**.
- Verificar la ubicación de **Error Log** y **Slow Query Log** en la instalación de MySQL.

2. Seguridad Avanzada (Conceptual):

- Investigar cómo se configura el uso de conexiones SSL/TLS para MySQL. ¿Por qué es importante para la seguridad de la comunicación entre cliente y servidor?

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen: SEMESTRAL

Horas Totales: 60

Carácter: TEÓRICA/PRÁCTICA

- Describa cómo implementaría una política de contraseñas robusta para los usuarios de MySQL.

D. Backup, Restauración y Estrategias de Respaldo

Objetivos:

- Comprender la importancia de los backups y las diferentes estrategias.
- Realizar backups lógicos de bases de datos utilizando `mysqldump`.
- Restaurar bases de datos a partir de backups.
- Diseñar un plan de respaldo básico.

1. Backup y Restauración:

- Explicar la diferencia entre un backup lógico y un backup físico en MySQL. Ejemplo de herramienta para cada uno.
- Describa en qué consiste un backup completo, un backup incremental y un backup diferencial. ¿en qué se diferencian?

2. Estrategias y Plan de Respaldo:

- ¿Cuáles son los componentes clave que debe incluir un "Plan de Respaldo" para una base de datos crítica?
- ¿Por qué es importante realizar "Pruebas de Restauración" de forma regular?

1. Backup Completo con `mysqldump`:

- Desde la línea de comandos de su sistema operativo realizar un backup completo de la base de datos `tp_db` en un archivo SQL.
- `mysqldump -u root -p tp_aplicadas_db > tp_db_full_backup.sql`
- Abrir el archivo `tp_db_full_backup.sql` con un editor de texto.
- Verificar contenido. ¿Qué tipo de sentencias SQL contiene?

2. Simulación de Desastre y Restauración:

- Eliminar la base de datos `tp_db` del servidor MySQL.
- `DROP DATABASE tp_aplicadas_db;`
- Restaurar la base de datos `tp_db` desde el archivo de backup creado.
- `mysql -u root -p < tp_aplicadas_db_full_backup.sql`
- Verificar que la base de datos `tp_db` y sus datos han sido restaurados correctamente.

3. Estrategias de Respaldo (Conceptual):

- Describir una estrategia de respaldo para una base de datos de producción que maneja transacciones 24/7, considerando backups completos semanales y backups incrementales diarios. ¿Qué herramientas o características de MySQL usarías?
- Diseñar un "Registro de Respaldo" simple. ¿Qué información mínima debería contener cada entrada de este registro?

E. Plan de Recuperación ante Incidentes, Redundancia y Medios de Almacenamiento

Objetivos:

- Comprender la importancia de un plan de recuperación ante incidentes.
- Explorar conceptos de redundancia de datos y hardware.



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen: SEMESTRAL

Horas Totales: 60

Carácter: TEÓRICA/PRÁCTICA

- Identificar diferentes medios de almacenamiento para bases de datos.

1. Plan de Recuperación ante Incidentes (DRP):

- ¿Qué es un Plan de Recuperación ante Incidentes (DRP) y por qué es esencial para la continuidad del negocio?
- Defina RTO (Recovery Time Objective) y RPO (Recovery Point Objective). ¿Cómo influyen estos en el diseño de un DRP?

2. Redundancia de Datos y Hardware:

- Explique el concepto de "Redundancia de Datos" en bases de datos. Mencione al menos dos formas de lograrla en MySQL.
- ¿Qué es la "Redundancia de Hardware" en el contexto de un servidor de bases de datos?

3. Medios de Almacenamiento:

- Compare y contraste los discos HDD (Hard Disk Drive) y SSD (Solid State Drive) como medios de almacenamiento para bases de datos en términos de rendimiento, costo y durabilidad.
- ¿Qué es una SAN (Storage Area Network) o una NAS (Network Attached Storage) y cómo pueden contribuir a la redundancia y escalabilidad del almacenamiento de bases de datos?

1. Diseño de un Plan de Recuperación ante Incidentes (Esquemático):

- Elaborar un esquema de un Plan de Recuperación ante Incidentes para la base de datos `tp_db`, asumiendo que es crítica para una pequeña aplicación web. Incluya al menos los siguientes puntos:
 - Detección del incidente.
 - Roles y responsabilidades del equipo de recuperación.
 - Procedimientos de restauración.
 - Pruebas del plan.
 - Comunicación.
 - RTO y RPO deseados para este escenario.

2. Redundancia de Datos (Conceptual):

- Investigar el concepto de "Replicación Master-Slave" en MySQL. Explicar sus beneficios para la redundancia y escalabilidad de lectura.
- ¿Cómo podría la replicación de MySQL ayudar a cumplir con un RPO bajo?

3. Redundancia de Hardware (Conceptual):

- Investigue y explique brevemente al menos dos niveles de RAID (ej: RAID 1, RAID 5, RAID 10) y cómo contribuyen a la redundancia de hardware para el almacenamiento de datos.
- ¿Qué otros componentes de hardware en un servidor de bases de datos deberían ser redundantes para asegurar alta disponibilidad?



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

F. Estrategias de Caché en Bases de Datos

Objetivos:

- Comprender la importancia del caché para el rendimiento de las bases de datos.
- Diferenciar entre caché de datos y caché de consultas.
- Identificar las principales estrategias de caché en MySQL.

1. Estrategias de Caché:

- ¿Cuál es el propósito principal de utilizar caché en un sistema de bases de datos?
- Explicar la diferencia fundamental entre "Caché de Datos" y "Caché de Consultas".

2. Caché en MySQL:

- Describir el rol del **InnoDB Buffer Pool** en MySQL. ¿Qué tipo de información almacena y por qué es importante para el rendimiento de las bases de datos InnoDB?
- Investigar sobre el **Query Cache** de MySQL. Verificar alternativas en versiones más recientes de MySQL (MySQL 8.0 en adelante)

1. Configuración del InnoDB Buffer Pool (Conceptual y Verificación):

- Conectarse a la línea de comandos de MySQL como **root**.
- Ejecutar **SHOW VARIABLES LIKE 'innodb_buffer_pool_size';**.
- Verificar que valor tiene y que representa.
- Investigar cómo se ajustaría este parámetro en el archivo **my.cnf** (o **my.ini**).
- ¿Qué consideraciones se deben tomar para decidir un tamaño adecuado para un servidor de producción?
- Ejecutar
- **SHOW GLOBAL STATUS LIKE 'Innodb_buffer_pool_read_requests';**
- **SHOW GLOBAL STATUS LIKE 'Innodb_buffer_pool_reads';**.
- Explicar cómo la relación entre estos dos números puede indicar la eficiencia del buffer pool.

2. Caché de Consultas (Verificación y Debate):

- Ejecutar:
- **SHOW VARIABLES LIKE 'query_cache%';**
- **SHOW STATUS LIKE 'Qcache%';**.
- Verificar salida de los comandos.
- ¿Qué otras estrategias de optimización de consultas son más relevantes? (Tener en cuenta que en nuevas versiones de MySQL, el Cache de consulta fue eliminado)



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

UNIDAD 3

Unidad 3: Programación Procedural.

SQL procedimental. Características. Procedimientos almacenados. Funciones almacenadas. Triggers (Disparadores). Variables locales y sentencias de control. Gestión de errores. Transacciones. Performance, índices, parámetros de configuración. Índices. Identificación de campos clave. Tipos de índices. Actualizaciones. Mantenimiento. Estadísticas. Plan de ejecución de consultas. Particionamiento de tablas.

Trabajo Práctico 3: Programación Procedural

1. Considere una transferencia entre dos cuentas. Para lograr esto tienes que escribir sentencias SQL que hagan lo siguiente:
 - Verifique la disponibilidad del monto solicitado en la primera cuenta.
 - Deducir cantidad solicitada de la primera cuenta
 - Depositarlo en segunda cuenta
 - Si se produce una falla en este proceso, el conjunto debe revertirse a su estado anterior.
2. Se tiene la siguiente tabla de datos:

Datos (id, apellido, nombre, saldo, mail)

- Crear un procedimiento almacenado que permita insertar datos.
- Crear un procedimiento almacenado que permita actualizar los datos, apellido, nombre o saldo.
- Crear un procedimiento almacenado que permita borrar datos.
- Cada operación debe quedar registrada en la la tabla log_datos
(id_cambio, id, apellido, nombre, saldo, mail, operación)
- En operación debe quedar registrado el tipo de operación que generó el registro. (I, U, D)

3. Crear la siguiente tabla:

socio(id, apellido, nombre, fechaingreso, actividad, estado)

- actividad: Fútbol, Tenis, Natación
- estado: Activo, Pasivo
- Crear un procedimiento almacenado que permita insertar datos del socio.



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen: SEMESTRAL

Horas Totales: 60

Carácter: TEÓRICA/PRÁCTICA

- Cargar en forma automática las 12 cuotas a pagar por el socio en la tabla cuotas con la siguiente estructura:

cuotas(id_cuota, cuota, monto)

Suponer cuota: 1 - 12
monto: 100

- Crear un procedimiento almacenado que permita actualizar los datos, apellido, nombre.
4. Crear un procedimiento que permita actualizar el estado de socio. Si pasa a Pasivo, cancelar todas las cuotas relacionadas.
Crear un procedimiento que permita actualizar cuota como pagada y descontar del saldo del socio.
5. Dada la siguiente tabla:
Competidores(id, nombre, puntos, penalización, premio)

- Crear un procedimiento que permita calcular el premio de cada competidor en función de los puntos acumulados menos la penalización recibida.
Premio = (Puntos acumulados – Penalización) * valor
Si puntos acumulados – penalización es menor que 100, el premio es directamente 0.
- Calcular el premio mediante una función almacenada. Considerar valor como constante.

6. Realizar los cambios necesarios en el modelo para el problema 5, considerando que el valor del premio depende de la competencia en la que haya participado el competidor.

Ejemplo:

Tenis: \$200

Futbol: \$300

Bochas: \$150

7. Una financiera desea crear una base de datos que permita registrar los datos de los clientes y sus movimientos.

Los datos que se necesitan son los siguientes:

Tabla Clientes:

IdCliente

Apellido

Nombres

Saldo

Tabla Movimientos

IdMovimiento

Fecha



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

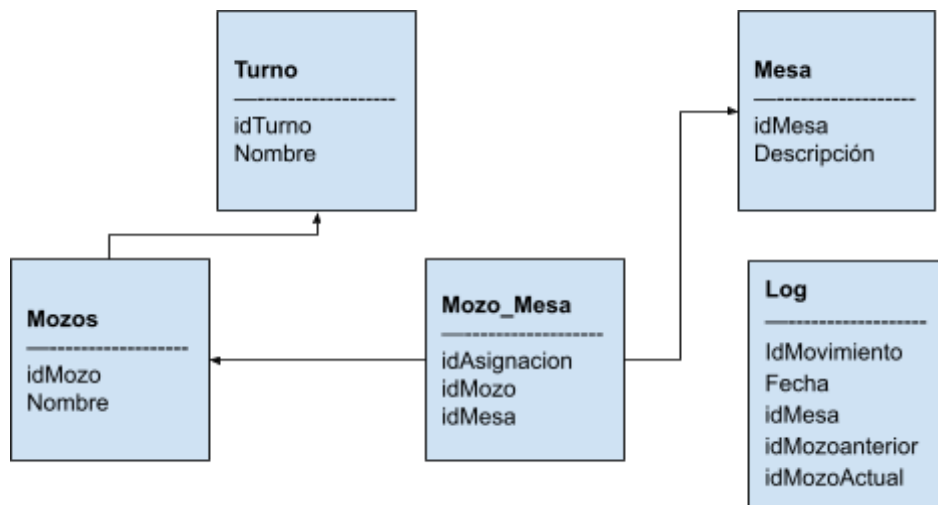
Tipo Movimiento (I - E)

Monto

Se pide:

- Crear una base de datos con el nombre Saldos
- Crear las tablas Clientes y Movimientos en la base de datos Saldos (Tener en cuenta todas las consideraciones necesarias para crear el modelo en la base. Ej: Claves)
- Crear un procedimiento almacenado que permita ingresar un nuevo movimiento en la tabla Movimientos para un determinado cliente.
- Por trigger, actualizar el saldo del Cliente al que se le generó el movimiento.

8. Procesos S.A., realiza el desarrollo de un software para empresas gastronómicas. El producto consta de varios módulos, que pueden ser implementados en forma independiente. Uno de los módulos más implementados es el que permite realizar el proceso de pedidos y cobros en el local de los restaurantes que lo implementan. El modelo de datos implementado es el siguiente:



Dentro de las reglas de gestión establecidas, se pueden destacar las siguientes:

- Un mozo puede atender varias mesas.
- Cada vez que se ocupa una mesa, se carga la misma, para luego se asigne mozo.
- Las mesas pueden ser reasignadas a un mozo determinado. Este proceso de reasignación, debe ser controlado y para ello se debe registrar en una tabla de log, los datos de los cambios de asignación que se producen.

Se pide:

- Crear una base de datos con el nombre BD
- Crear las tablas indicadas en el modelo. (Tener en cuenta todas las consideraciones necesarias para crear el modelo en la base. Ej: Claves, tipo de datos, etc)



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen: SEMESTRAL

Horas Totales: 60

Carácter: TEÓRICA/PRÁCTICA

- Crear un procedimiento almacenado que permita asignar una mesa a un mozo.
- Crear una función que cuente la cantidad de mesas sin mozo asignado.
- Por trigger, registrar en la tabla Log, los datos del cambio de asignación



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

UNIDAD 4

Unidad 4: Almacén de Datos

Sistema Soporte a la Toma de Decisiones. Introducción a la Inteligencia de Negocios. Implementación de un almacén de datos. Datawarehouse. DataMart. Integración de Datos. Extracción. Transformación. Carga de datos. Modelo de datos de implementación. Modelo en Estrella, Modelo Copo de Nieve, Modelo Constelación.

Procesamiento Analítico en Línea (OLAP). Implementaciones. Base de Datos Multidimensional. Medidas. Dimensiones. Hechos. Jerarquías. Dimensión Tiempo. Cubos OLAP.

Trabajo Práctico 4: Almacén de Datos

1. Se cuenta con una base de datos que contiene información correspondiente al uso de transporte.

Los archivos de información son los siguientes:

dat-ab-usos-2024.csv: Contiene el conjunto de datos.

Construir un almacén de datos que contenga la totalidad de los datos con el objetivo de poder armar un conjunto de consultas e indicadores sobre el tema.

Se pide desarrollar las siguientes tareas:

- Crear una base de datos denominada USO, para contener los datos del programa. Cargar los datos del archivo en la base de datos USO.
- Realizar la evaluación de los datos de origen y determinar:
 - ¿Qué tipo de información relevante se puede obtener?
 - ¿Cuáles son las perspectivas de análisis que se pueden considerar?
- En base a la evaluación realizada del origen de datos:
 - Crear el modelo conceptual del caso.
- Determinar los indicadores definidos y forma de cálculo de los mismos.
- Elaborar un modelo de datos para el almacén de datos. (Utilizar esquema estrella)
- Realizar el proceso ETL con SQL, mediante procedimientos almacenados.
- Crear una visualización basada en los datos del almacén de datos construido. (10 puntos)

2. Proyecto Final (A definir)

Unidad Académica: FACULTAD DE INGENIERÍA
Asignatura: BASE DE DATOS APLICADA
Año académico: 2025 **Régimen:** SEMESTRAL
Horas Totales: 60 **Carácter:** TEÓRICA/PRÁCTICA

UNIDAD 5

Unidad 5: Bases de Datos no estructuradas

Concepto. Datos no estructurados. Características. Flexibilidad. Diversidad de Formatos. Eficiencia en almacenamiento. Búsqueda y recuperación de datos. Mecanismos de búsquedas. Aplicaciones. Big Data. Concepto. Tecnologías. Motores de búsqueda. Sistemas de almacenamiento de objetos en la nube. DataMining. Concepto.

Trabajo Práctico 5: Bases de Datos no estructuradas

1. Ejemplo: Desarrollar una base de datos para almacenar los productos que vende una empresa de ventas online.

Base: **Prueba**

Colección: **Producto**

Campos:

- **nombre**
 - **precio**
 - **stock**
 - **categoría**
- Crea tabla producto agregando un registro
 - Ingreso de registros múltiples
2. Tomando la base de datos creada en el punto anterior, desarrollar las siguientes consultas.
 - Mostrar todos los Productos
 - Mostrar todos los productos con precio mayor a 1000
 - Mostrar los productos con nombre Notebook
 - Mostrar los productos de categoría Accesorios y stock < 100
 - Realizar una consulta que permita mostrar los productos que tengan nombre Notebook o con precio menor a 30.
 - Mostrar nombre y precio de un producto específico.
 - Mostrar los primeros dos registros de datos de la tabla Producto (Limitar el número de resultados)
 - Mostrar la lista de productos ordenados por precio de forma ascendente
 - Mostrar la lista de productos ordenados por stock de forma descendente
 - Actualizar el precio y stock de un producto (Por ejemplo Notebook)
 - Actualizar el stock de todos los productos de categoría Accesorio
 - Borra el registro correspondiente al producto Mouse
 - Eliminar los productos con stock menor a 20 unidades
 - Borrar todos los Productos
 - Eliminar la colección Productos



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA
Asignatura: BASE DE DATOS APLICADA
Año académico: 2025 **Régimen:** SEMESTRAL
Horas Totales: 60 **Carácter:** TEÓRICA/PRÁCTICA

3. Se desea realizar una base de datos para implementar un blog. La base de datos debe almacenar las publicaciones y comentarios asociados a cada publicación.

Se pide:

- Crear una base de datos en MongoDB llamada Blog
- Crear las colecciones posts y comentarios
- Poblar ambas colecciones (2 posts y 2 comentarios asociados a cada post)

Columnas posts

- _id
- titulo
- contenido
- autor
- categoria
- publicado

Columnas comentarios

- _id
- postId
- autor
- email
- contenido



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Unidad Académica: FACULTAD DE INGENIERÍA

Asignatura: BASE DE DATOS APLICADA

Año académico: 2025

Régimen:

SEMESTRAL

Horas Totales: 60

Carácter:

TEÓRICA/PRÁCTICA

1.BIBLIOGRAFÍA

1.1.Bibliografía obligatoria

- MySQL "La Biblia" - Ian Gilfillan - Anaya Multimedia [2013]
- Hefesto 3.0 - Darío Bernabeu y Mariano García Mattío [2017]
- Pentaho Data Integration Cookbook Second Edition 2013/Packt Publishing - Alex Meadows, Adrián Sergio Pulvirenti y María Carina Roldán

1.2.Bibliografía complementaria

- Pentaho 5.0 Reporting By Example Beginner's Guide - 2013/Packt Publishing - Darío Bernabeu & Mariano García Mattío
- Big Data Análisis de grandes volúmenes de datos en organizaciones - 2013/Alfaomega Luis Joyanes Aguilar
- Pentaho Business Analytics Cookbook 2014/ Packt Publishing - Sergio Ramazzina

1.3.Otros recursos web y multimediales

- <https://dev.mysql.com/doc/>