

# Informe del Proyecto Práctico ## Administración y Automatización de Base de Datos ###  
"Libros del Rincón"

**Autor:** [Tu Nombre]

**Asignatura:** UF1468 / UF1469 / UF1470

**Fecha de entrega:** [DD/MM/AAAA]

 Logo Librería{width=200px}

## Índice

1. [Introducción](#)
2. [Diseño de la Base de Datos \(UF1468\)](#)
3. [Creación de la Base de Datos y Gestión de Usuarios \(UF1469\)](#)
4. [Introducción de Datos \(UF1469\)](#)
5. [Automatización de Tareas \(UF1470\)](#)
6. [Optimización de Consultas \(UF1470\)](#)
7. [Planificación de Backups \(UF1468\)](#)
8. [Script Python para creación de pedidos](#)
9. [Dificultades y Conclusiones](#)
10. [Anexos](#)

## Introducción {#introduccion}

Este documento recoge el desarrollo completo del proyecto práctico para la librería online **"Libros del Rincón"**. Incluye:

- Diseño del modelo relacional.
- Creación y asignación de usuarios y permisos.
- Inserción de datos de prueba.
- Scripts de automatización (triggers, batch, cron).
- Optimización de consultas críticas.
- Planificación de tareas administrativas (backups e informes).

## Diseño de la Base de Datos (UF1468)

Elegimos **MySQL** por su robustez y capacidad multiusuario. El esquema consta de:

```
CREATE TABLE Libros (  
  id_libro INT PRIMARY KEY AUTO_INCREMENT,  
  titulo VARCHAR(255),  
  autor VARCHAR(255),  
  genero VARCHAR(100),  
  precio DECIMAL(6,2),  
  cantidad_en_stock INT
```

```
);  
-- Tablas Clientes, Pedidos e Items_Pedido definidas de forma análoga
```

Se tomaron decisiones de diseño clave:

- Claves primarias autoincrementales para unicidad automática.
- Claves foráneas entre `Pedidos.id_cliente` → `Clientes.id_cliente` y `Items_Pedido.id_libro` → `Libros.id_libro` para integridad.
- Tipos de datos adecuados (`INT`, `VARCHAR`, `DECIMAL`, `DATE`) para precisión y eficiencia.



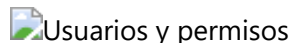
Creación BD

## Creación de la Base de Datos y Gestión de Usuarios (UF1469)

Definimos dos roles:

1. **gerente**: ALL PRIVILEGES sobre `libros_rincon.*`.
2. **agenteventas**:
  - `SELECT` en `Libros` y `Clientes`.
  - `SELECT`, `INSERT` en `Pedidos`.

```
CREATE USER 'gerente'@'localhost' IDENTIFIED BY 'Gerente123';  
GRANT ALL PRIVILEGES ON libros_rincon.* TO 'gerente'@'localhost';  
  
CREATE USER 'agenteventas'@'localhost' IDENTIFIED BY 'Ventas123';  
GRANT SELECT ON libros_rincon.Libros TO 'agenteventas'@'localhost';  
GRANT SELECT ON libros_rincon.Clientes TO 'agenteventas'@'localhost';  
GRANT SELECT, INSERT ON libros_rincon.Pedidos TO  
'agenteventas'@'localhost';
```



Usuarios y permisos





## Introducción de Datos (UF1469)

Cargamos datos de ejemplo en todas las tablas:

```
INSERT INTO Clientes (nombre, apellido, email) VALUES  
( 'Ana', 'García', 'ana@example.com'),  
( 'Luis', 'Pérez', 'luis@example.com'),  
...;  
  
INSERT INTO Libros (titulo, autor, genero, precio, cantidad_en_stock)  
VALUES  
( '1984', 'George Orwell', 'Ciencia Ficción', 9.99, 20),
```

```
('Cien Años de Soledad','Gabriel García Márquez','Novela',14.50,15),  
...;
```

### Capturas de ingesta:

- **Cientes:** Clientes
- **Libros:** Libros
- **Pedidos:** Pedidos
- **Items\_Pedido:** Items Pedido

## Automatización de Tareas (UF1470)

### Trigger de stock

```
DELIMITER //  
CREATE TRIGGER actualizar_stock AFTER INSERT ON Items_Pedido  
FOR EACH ROW  
BEGIN  
    UPDATE Libros  
        SET cantidad_en_stock = cantidad_en_stock - NEW.cantidad  
        WHERE id_libro = NEW.id_libro;  
END;//  
DELIMITER ;
```



Trigger stock

### Informe diario de ventas (Windows)

#### Batch (.bat):

```
@echo off  
for /f "tokens=2 delims==" %%i in ('wmic os get localdatetime /value')  
do set dt=%%i  
set fecha=!dt:~0,4!-!dt:~4,2!-!dt:~6,2!  
mysql -u TU_USUARIO -pTU_CONTRASEÑA libros_rincon <  
scripts/mysql/06_informe_diario_ventas.sql > ventas_%fecha%.txt
```

#### Consulta SQL:

```
SELECT CURDATE() AS fecha_informe, l.genero,  
SUM(ip.cantidad*ip.precio_por_item) AS total_ventas  
FROM Items_Pedido ip  
JOIN Libros l ON ip.id_libro = l.id_libro  
JOIN Pedidos p ON ip.id_pedido = p.id_pedido
```

```
WHERE p.fecha_pedido = CURDATE()  
GROUP BY l.genero;
```

 Resultado informe

## Script Python para creación de pedidos {#script-python-para-creacion-de-pedidos}

```
import mysql.connector  
  
def crear_pedido(id_cliente, id_libro, cantidad):  
    db = mysql.connector.connect(user='TU_USUARIO',  
                                password='TU_CONTRASEÑA',  
                                host='localhost',  
                                database='libros_rincon')  
    cursor = db.cursor()  
    cursor.execute(  
        "INSERT INTO Pedidos (id_cliente, fecha_pedido, monto_total)  
VALUES (%s, CURDATE(), %s)",  
        (id_cliente, cantidad * obtenerPrecio(id_libro, cursor))  
    )  
    pedido_id = cursor.lastrowid  
    precio = obtenerPrecio(id_libro, cursor)  
    cursor.execute(  
        "INSERT INTO Items_Pedido VALUES (%s, %s, %s, %s)",  
        (pedido_id, id_libro, cantidad, precio)  
    )  
    db.commit()  
    cursor.close()  
    db.close()  
  
def obtenerPrecio(id_libro, cursor):  
    cursor.execute("SELECT precio FROM Libros WHERE id_libro=%s",  
        (id_libro,))  
    return cursor.fetchone()[0]
```

## Optimización de Consultas (UF1470)

```
EXPLAIN SELECT DISTINCT c.nombre, c.apellido  
FROM Clientes c  
JOIN Pedidos p ON c.id_cliente = p.id_cliente  
JOIN Items_Pedido ip ON p.id_pedido = ip.id_pedido  
JOIN Libros l ON ip.id_libro = l.id_libro  
WHERE l.genero = 'Ciencia Ficción'  
AND p.fecha_pedido >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH);
```

```
CREATE INDEX idx_genero ON Libros(genero);  
CREATE INDEX idx_fecha ON Pedidos(fecha_pedido);
```

 EXPLAIN y índices

## Planificación de Backups (UF1468)

- **Windows (.bat):**

```
@echo off  
set fecha=%date:~6,4%-%date:~3,2%-%date:~0,2%  
mysqldump -u TU_USUARIO -pTU_CONTRASEÑA libros_rincon >  
backup_%fecha%.sql
```

- **Linux (cron):**

```
0 2 * * * mysqldump -u TU_USUARIO -pTU_CONTRASEÑA libros_rincon >  
/ruta/backup_$(date +%F).sql
```

 Backup Cron

## Dificultades y Conclusiones {#dificultades-y-conclusiones}

1. Formateo de fecha en Windows para nombrar backups.
2. Integridad referencial al insertar datos manualmente.
3. Ajuste de CSS para portada, márgenes y saltos de página en Markdown PDF.

El resultado es un sistema completo, automático y documentado.

## Anexos {#anexos}

- Scripts completos en [scripts/mysql/](#), [scripts/python/](#), [scripts/tareas\\_programadas/](#).
- Capturas originales en [docs/capturas](#).
- [Repositorio GitHub](#).