

Guía #2

JDBC y Patrones de Diseño

Objetivo

El objetivo de este trabajo práctico es que los alumnos aprendan a:

1. Conectarse a una base de datos MySQL desde Java usando JDBC.
2. Hacer uso del patrón de diseño Singleton para manejar la conexión a la base de datos.
3. Insertar, consultar, actualizar y eliminar registros desde Java.
4. Manejar excepciones y cerrar correctamente la conexión a la base de datos.

Base de Datos

Se provee un archivo `alumnos.sql` que contiene la siguiente estructura de tabla:

```
CREATE TABLE alumnos (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    nombre TEXT NOT NULL,  
    apellido TEXT NOT NULL,  
    edad INTEGER NOT NULL,  
    email TEXT UNIQUE NOT NULL  
);  
  
CREATE TABLE direcciones (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    calle TEXT NOT NULL,  
    altura INTEGER NOT NULL,  
    alumno_id INTEGER NOT NULL,  
    FOREIGN KEY (alumno_id) REFERENCES alumnos(id) ON DELETE CASCADE  
);
```

Datos Iniciales en la Base de Datos

La base de datos ya tiene algunos registros precargados:

id	nombre	apellido	edad	email
1	Pedro	Gutierrez	21	pedro@email.com
2	Matias	Lopez	22	matias@email.com

3	Sofia	Martinez	20	sofia@email.com
4	Martin	Gabilan	19	martin@email.com
5	Camila	Rossi	23	camila@email.com

id	calle	altura	alumno_id
1	Av. Siempre Viva	742	1
2	Calle Falsa	123	2
3	Boulevard Central	456	3
4	Av. Libertador	789	4
5	San Martín	1011	5

Actividades

Los alumnos deben escribir un programa en Java que realice las siguientes operaciones sobre la base de datos usando JDBC:

1. Conectarse a la Base de Datos

- Crear una clase `ConexionMySQL` que maneje la conexión a la base de datos MySQL.
- Usar el controlador JDBC de MySQL.

2. Insertar un Nuevo Alumno

- Pedir por consola el nombre, edad y email de un nuevo alumno.
- Insertarlo en la base de datos.

3. Insertar una nueva dirección

- Pedir por consola los datos e insertarlo en la base de datos.
- Manejar correctamente el caso de que el alumno al que corresponde la dirección no exista.

4. Listar Todos los Alumnos

- Consultar y mostrar todos los registros de la tabla `alumnos`.
- Permitir consultar y mostrar todos los alumnos junto a sus direcciones. (Tener en cuenta que puede haber mas de una dirección por alumno)

5. Listar todas las direcciones

- Permitir dado un alumno, poder visualizar todas sus direcciones

6. Actualizar la Edad de un Alumno.

- Pedir el ID de un alumno y su nueva edad.
- Actualizar la edad en la base de datos.

7. Eliminar un Alumno

- Pedir el ID de un alumno y eliminarlo de la base de datos.

8. Eliminar una dirección

- Pedir el ID de una dirección y eliminarla de la base de datos.