1. JDBC: Introducción a las bases de datos relacionales

SISTEMAS GESTORES DB

1.1. SQLite

SQLite es uno de los sistemas gestores de bases de datos (SGBD) más usados en dispositivos móviles. Es un motor autónomo, sin servidor, sin configuración y transaccional. Es ideal para proyectos que requieren portabilidad.

Ventajas:

- Orientado a archivos, no requiere instalación de servicios.
- Compatible con bases de datos en memoria y en archivo.
- Muy usado en dispositivos móviles y computadoras.

Enlaces útiles:

- SQLite Sitio Oficial
- SQLite Browser

▼ 1.1. Dependencias Maven

Para usar SQLite en Java se requiere la dependencia del driver JDBC en el archivo `pom.xml`:

```
<dependencies>
 <dependency>
  <groupId>org.xerial</groupId>
  <artifactId>sqlite-jdbc</artifactId>
  <version>3.47.0.0</version>
 </dependency>
</dependencies>
```

Última versión disponible: **3.47.0.0**

Descarga directa: Maven Repository.

▼ 1.2. Bases de datos en memoria

Son útiles para pruebas o datos temporales:

```
Connection conex = DriverManager.getConnection("jdbc:sqlite::memory:");
```

▼ 1.3. Bases de datos en archivo

Para datos persistentes:

```
Connection conex = DriverManager.getConnection("jdbc:sqlite:rutaArchivo.sqlite3");
```

▼ 1.4. Recursos adicionales

- Documentación de Java para manejo de tablas con Swing.
- Tutoriales de configuración en IDEs como NetBeans e IntelliJ IDEA.

1.2. DAO (Data Access Object)

Patrón de diseño que separa la lógica de acceso a datos del resto de la aplicación. Es compatible con JDBC y proporciona una interfaz consistente para manejar operaciones CRUD.

Recursos recomendados:

▼ Tutoriales de Oracle y YouTube sobre DAO.

YouTube - Tutorial DAO

YouTube - JDBC DAO Tutorial

- Artículo en Acódigo Blog.
- Oracle - Data Access Object

1.3. Creación de una Base de Datos en H2

H2 es otro SGBD embebido, ideal para entornos de prueba o proyectos ligeros.

▼ 1.3.1. Dependencias Maven

Versión 2.x:

```
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <version>2.3.232</version>
</dependency>
```

Versión 1.x:

```
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <version>1.4.200</version>
</dependency>
```

▼ 1.3.2. Configuración

- Crear una nueva conexión en el IDE o usando herramientas como **DBeaver**.
- Definir la URL de conexión, por ejemplo:

```
jdbc:h2:RutaABaseDatos\nomeBD
```

▼ 1.3.3. Ejemplo de base de datos

Definición de tablas y relaciones:

```
CREATE TABLE PUBLIC.Debuxo (
    idDebuxo INTEGER NOT NULL AUTO_INCREMENT,
    nome CHARACTER VARYING(64) NOT NULL,
    CONSTRAINT DEBUXO_PK PRIMARY KEY (idDebuxo)
);

CREATE TABLE PUBLIC.Shape (
    idDebuxo INTEGER NOT NULL,
    shape BINARY LARGE OBJECT,
    CONSTRAINT SHAPE_FK FOREIGN KEY (idDebuxo) REFERENCES PU
```

```
BLIC.Debuxo(idDebuxo) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Recursos útiles:

- DBeaver para gestionar bases de datos.
- Sitio oficial de H2: H2 Database.

INTRODUCCIÓN A LAS BASES DE DATOS RELACIONALES

2.1. Introducción a Bases de Datos Relacionales y SQL

- **Base de datos:** Es una colección organizada de datos.
- **Base de datos relacional:** Organizada en tablas compuestas por filas y columnas.
- **Acceso desde Java:**
  1. **JDBC (Java Database Connectivity):** Manipula datos como filas y columnas.
  2. **JPA (Java Persistence API):** Maneja datos a través de objetos (ORM, explicado con Hibernate).
- **SQL (Structured Query Language):** Lenguaje estándar para interactuar con bases de datos relacionales.
- **Tipos de bases de datos:**
  - **Relacionales:** Tablas (ejemplo: MySQL, PostgreSQL).
  - **NoSQL:** Clave-valor, documentos, gráficos (explicado en futuras unidades).

▼ 2.1.1. Derby

- Pequeño SGBD en memoria; solo requiere un archivo JAR.
- Alternativas a Derby para trabajar:

- **SQLite y H2:** Uso en ejemplos.
- **MySQL, MariaDB, PostgreSQL:** Motores completos y populares.

2.2. Ejemplo de una base de datos relacional

- **Relaciones:** Un animal pertenece a una especie.
- **Tablas de ejemplo:**
  1. **Especie**

idEspecie (PK)	nome (VARCHAR)	area (DECIMAL)
1	Elefante Africano	9.5
2	Cebra	3.1

  2. **Animal**

idAnimal (PK)	idEspecie (FK)	nome (VARCHAR)	dataNacemento (TIMESTAMP)
1	1	Pepa	2001-05-06 02:15:00
2	2	Lola	2012-08-15 09:12:00

▼ 2.2.1 Código para configurar la base de datos (Derby)

- **Configuración inicial:**
  - Agregar `<PATH TO DERBY>/derby.jar` al classpath o dependencia Maven.
  - Crear tablas y datos de ejemplo con **JDBC**:

```
java
Copiar código
public class SetupDerbyDatabase {
    public static void main(String[] args) throws Exception {
        String url = "jdbc:derby:zoo;create=true";
        try (Connection conexion = DriverManager.getConnection(url)) {
            executar(conexion, "CREATE TABLE Especie (idEspecie INTEGER PI
            executar(conexion, "CREATE TABLE Animal (idAnimal INTEGER PRI
            executar(conexion, "INSERT INTO Especie VALUES (1, 'Elefante Afri
            executar(conexion, "INSERT INTO Especie VALUES (2, 'Cebra', 3.1)"
            executar(conexion, "INSERT INTO Animal VALUES (1, 1, 'Pepa', '200
        }
```

```
    }
    private static void executar(Connection conexion, String sql) throws SQL
        try (PreparedStatement ps = conexion.prepareStatement(sql)) {
            ps.executeUpdate();
        }
    }
}
```

- **Ejecución del programa:**
  - Linux/MacOS:

    ```
    java -cp "<path_to_derby>/derby.jar:." SetupDerbyDatabase
    ```

  - Windows:

    ```
    java -cp "c:\program files\jdk\db\lib\derby.jar;." SetupDerbyData
    base
    ```

2.3. Repaso de declaraciones SQL básicas

> 💡 El SQL básico es común en la mayoría de los SGBD relacionales. SQL avanzado puede variar según el sistema utilizado.

- **CRUD (Create, Read, Update, Delete):** Operaciones esenciales para datos en bases relacionales.

Operación	Palabra clave SQL	Descripción
Crear	INSERT	Añade filas a una tabla.
Leer	SELECT	Recupera datos de la tabla.
Actualizar	UPDATE	Modifica datos existentes.
Eliminar	DELETE	Borra filas de la tabla.

▼ Ejemplos básicos:

- **INSERT:**

```
INSERT INTO Especie VALUES (3, 'Elefante Asiático', 10.8);
```

- **SELECT:**

```
SELECT nome, area FROM Especie WHERE idEspecie = 3
```

- **UPDATE:**

```
UPDATE Especie SET area = area + 0.5 WHERE nome = 'Elefante Asi
ático';
```

- **DELETE:**

```
DELETE FROM Especie WHERE nome = 'Elefante Asiático';
```