Base dados H2

A base de dados H2 é uma base de dados relacional escrita em Java, que pode ser usada tanto em modo embutido quanto em modo cliente-servidor. Quando usada em memória, ela é particularmente útil para testes e desenvolvimento, pois os dados são armazenados na RAM e não são persistidos no disco, o que torna as operações muito rápidas.

Principais Características do H2 em Memória

- **Volatilidade**: Os dados são armazenados na memória RAM, o que significa que eles são perdidos quando a aplicação é encerrada. -** Velocidade**: Como os dados não são escritos no disco, as operações de leitura e escrita são muito rápidas.
- **Facilidade de Configuração**: É fácil de configurar e integrar em aplicações Java, especialmente para testes unitários e de integração.
- **Console Web**: O H2 oferece uma consola web que permite visualizar e manipular os dados diretamente no browser.

Passo 1 - Configurar o Maven

```
<dependencies>
    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
        <version>2.1.214</version>
        <scope>runtime</scope>
    </dependency>
</dependencies>
```

Passo 2 - Java: Criar Connection

A base de dados H2 tem um SQL compativel com diversas bases de dados (embora não suporta todas as funcionalidades proprietárias)

- PostgreSQL
- MySQL
- Oracle
- SQL Server
- DB2

A forma de escolher a base de dados modelo:

```
String url = "jdbc:h2:mem:testdb;MODE=PostgreSQL";
Connection conn = DriverManager.getConnection(url, "sa", "");
```

Um exemplo de uma classe que uso particularmente nos exemplos e exercícios:

```
public class H2BaseDados {
    public static Connection obtemConnection() {
        try {
            return DriverManager.getConnection("jdbc:h2:mem:utilizadores", "sa",
"");
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }

    public static void inicializa(Connection con) {
        try {
            con.createStatement().execute("CREATE TABLE utilizadores (USERNAME
VARCHAR(50), PASSWORD VARCHAR(50), DATA DATE);");
            con.createStatement().execute("INSERT INTO utilizadores (username,
password, data) values ('admin','21232f297a57a5a743894a0e4a801fc3','2024-01-
01');");
        } catch(Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Passo 3 - Usar a Connection numa classe DAO

```
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class UserDao {
    public void createTable() throws SQLException {
        String sql = "CREATE TABLE IF NOT EXISTS users (id BIGINT AUTO_INCREMENT
PRIMARY KEY, name VARCHAR(255))";
        try (Connection conn = DatabaseConnection.getConnection();
             Statement stmt = conn.createStatement()) {
            stmt.execute(sql);
        }
    }

    public void insertUser(User user) throws SQLException {
        String sql = "INSERT INTO users (name) VALUES (?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, user.getName());
            pstmt.executeUpdate();
        }
    }

    public User getUserById(Long id) throws SQLException {
        String sql = "SELECT * FROM users WHERE id = ?";
```

```
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setLong(1, id);
            try (ResultSet rs = pstmt.executeQuery()) {
                if (rs.next()) {
                    User user = new User();
                    user.setId(rs.getLong("id"));
                    user.setName(rs.getString("name"));
                    return user;
                }
            }
        }
        return null;
    }

    public List<User> getAllUsers() throws SQLException {
        List<User> users = new ArrayList<>();
        String sql = "SELECT * FROM users";
        try (Connection conn = DatabaseConnection.getConnection();
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(sql)) {
            while (rs.next()) {
                User user = new User();
                user.setId(rs.getLong("id"));
                user.setName(rs.getString("name"));
                users.add(user);
            }
        }
        return users;
    }

    public void updateUser(User user) throws SQLException {
        String sql = "UPDATE users SET name = ? WHERE id = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, user.getName());
            pstmt.setLong(2, user.getId());
            pstmt.executeUpdate();
        }
    }

    public void deleteUser(Long id) throws SQLException {
        String sql = "DELETE FROM users WHERE id = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setLong(1, id);
            pstmt.executeUpdate();
        }
    }
}
```