**TALLER 3 Dia 2**

**OBJETIVO: Implementar desde 0 un proyecto**

* **CONFIGURAR POM.XML**

<dependencies>

<!-- Spring Boot Starter Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- H2 Database -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Starter Test -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

* **IMPLEMENTAR LAS CLASES**

**User.java** (Modelo de usuario)

package com.example.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

// Getters y Setters

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**UserRepository.java** (Interfaz del repositorio)

package com.example.repository;

import com.example.model.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {

}

**Escribir pruebas con @DataJpaTest**

**UserRepositoryTest.java** (Pruebas de integracion)

package com.example.repository;

import com.example.model.User;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTest;

import org.springframework.boot.test.autoconfigure.orm.jpa.TestEntityManager;

import java.util.Optional;

import static org.junit.jupiter.api.Assertions.\*;

@DataJpaTest

class UserRepositoryTest {

@Autowired

private TestEntityManager entityManager;

@Autowired

private UserRepository userRepository;

@Test

void testFindById() {

// Given

User user = new User();

user.setName("John Doe");

entityManager.persist(user);

entityManager.flush();

// When

Optional<User> foundUser = userRepository.findById(user.getId());

// Then

assertTrue(foundUser.isPresent());

assertEquals("John Doe", foundUser.get().getName());

}

@Test

void testSaveUser() {

// Given

User user = new User();

user.setName("Jane Doe");

// When

User savedUser = userRepository.save(user);

// Then

assertNotNull(savedUser.getId());

assertEquals("Jane Doe", savedUser.getName());

}

@Test

void testDeleteUser() {

// Given

User user = new User();

user.setName("John Doe");

entityManager.persist(user);

entityManager.flush();

// When

userRepository.deleteById(user.getId());

// Then

Optional<User> deletedUser = userRepository.findById(user.getId());

assertFalse(deletedUser.isPresent());

}

}

Configurar**application.properties**

Añade la configuración de la base de datos H2 src/main/resources/application.properties

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true