

Login:

Pruebas de junit

```
package com.example.gestion_panaderia.controller;

import com.example.gestion_panaderia.model.Usuario;
import com.example.gestion_panaderia.service.IAuthService;

import javafx.embed.swing.JFXPanel;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;

import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

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

// Servicio falso para pruebas
class AuthServiceFake implements IAuthService {
    @Override
    public Usuario autenticar(String usuario, String password) {
        if (usuario.equals("admin") && password.equals("1234")) {
            return new Usuario("admin", "1234", "ADMIN");
        }
        return null;
    }
}
```

```
public class LoginControllerTest {

    private LoginController controller;

    @BeforeEach
    void setUp() {
        // Inicializar JavaFX
        new JFXPanel();

        // Crear instancia del controlador
        controller = new LoginController();

        // Crear componentes JavaFX simulados
        controller.txtUsuario = new TextField();
        controller.txtPassword = new PasswordField();
        controller.lblMensaje = new Label();
        controller.boxMensaje = new VBox();
        controller.btnAceptar = new Button();
        controller.btnCancelar = new Button();

        // Inyectar servicio falso
        controller.authService = new AuthServiceFake();
    }
}
```

```
@Test
void testLoginCorrecto() {
    controller.txtUsuario.setText("admin");
    controller.txtPassword.setText("1234");

    // Simular clic en el botón de aceptar
    controller.btnAceptar.fire();

    // Como el login es correcto, no debe haber mensaje de error
    // (aunque boxMensaje podría estar visible por Main.cambiarVista que no se ejecuta en pruebas)
    assertEquals("Usuario o contraseña incorrectos", controller.lblMensaje.getText());
}

@Test
void testLoginIncorrecto() {
    controller.txtUsuario.setText("pepe");
    controller.txtPassword.setText("malo");

    // Simular clic en el botón de aceptar
    controller.btnAceptar.fire();

    // Verificar mensaje de error
    assertTrue(controller.boxMensaje.isVisible());
    assertEquals("Usuario o contraseña incorrectos", controller.lblMensaje.getText());
}
```

```
@Test
void testCamposVacios() {
    controller.txtUsuario.setText("");
    controller.txtPassword.setText("");

    // Simular clic en el botón de aceptar
    controller.btnAceptar.fire();

    // Verificar mensaje de campos vacios
    assertTrue(controller.boxMensaje.isVisible());
    assertEquals("Por favor complete todos los campos", controller.lblMensaje.getText());
}

@Test
void testCampoUsuarioVacio() {
    controller.txtUsuario.setText("");
    controller.txtPassword.setText("1234");

    controller.btnAceptar.fire();

    assertTrue(controller.boxMensaje.isVisible());
    assertEquals("Por favor complete todos los campos", controller.lblMensaje.getText());
}
```

```
@Test
void testCampoPasswordVacio() {
    controller.txtUsuario.setText("admin");
    controller.txtPassword.setText("");

    controller.btnAceptar.fire();

    assertTrue(controller.boxMensaje.isVisible());
    assertEquals("Por favor complete todos los campos", controller.lblMensaje.getText());
}
}
```



¿Qué está pasando?

La prueba JUnit está:

- Abriendo la ventana de login
- Ingresando datos de prueba
- Verificando que aparezca el mensaje de error correcto
- Esto es exactamente lo que debe hacer una prueba