



GUÍA TESTING MENU LOGIN

ADMINISTRACIÓN DE INVENTARIO DE
CLÍNICA

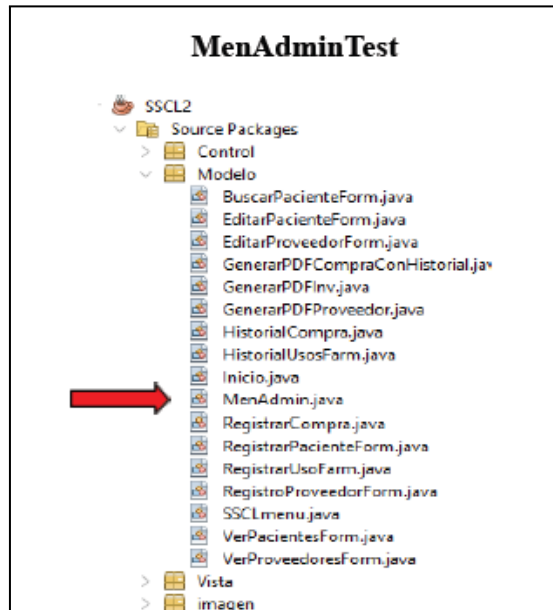
ALDO SAULA ESTEBAN LARCO



Guía Paso a Paso para Pruebas Unitarias con JUnit4

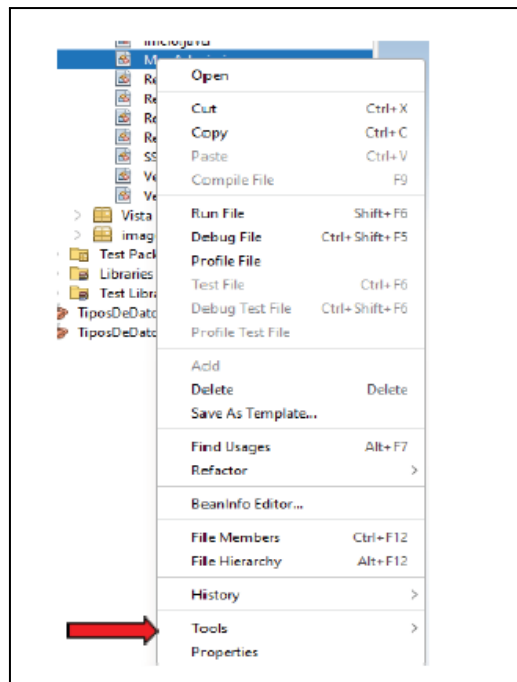
1. Identificación de la Clase a Probar

- Abre tu entorno de desarrollo integrado (IDE).
- Localiza la clase MenAdmin a la que se le realizarán las pruebas unitarias.



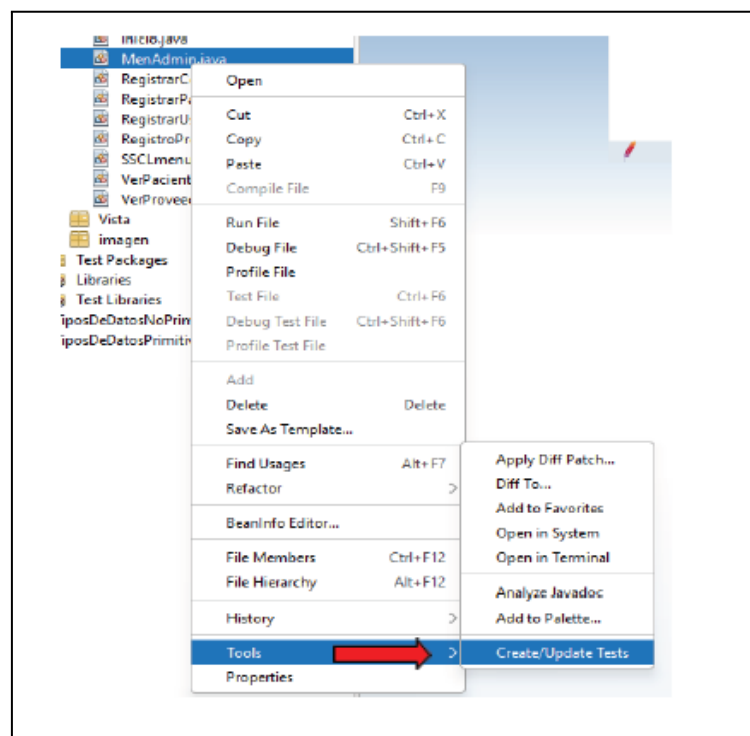
2. Acceso a las Herramientas de Pruebas Unitarias

- Haz clic derecho sobre la clase MenAdmin.
- En el menú desplegable, busca y selecciona la opción "Tools" (Herramientas).



3. Creación de Pruebas Unitarias

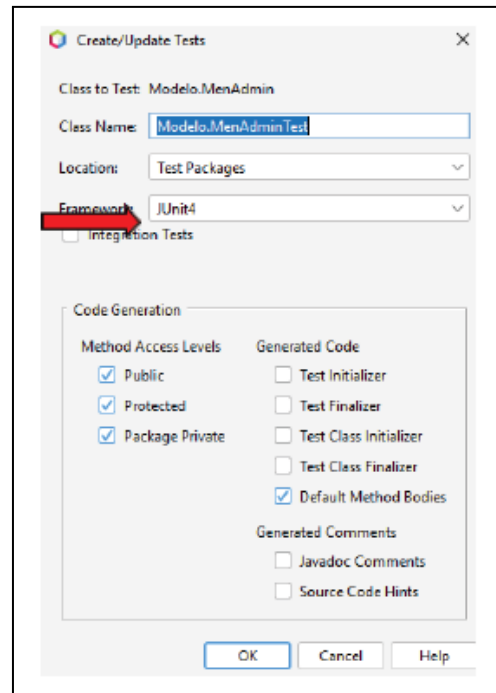
- Dentro del menú "Tools", selecciona la opción "CreateUpdateTests" para iniciar la creación de las pruebas unitarias.



4. Configuración del Formulario de Pruebas

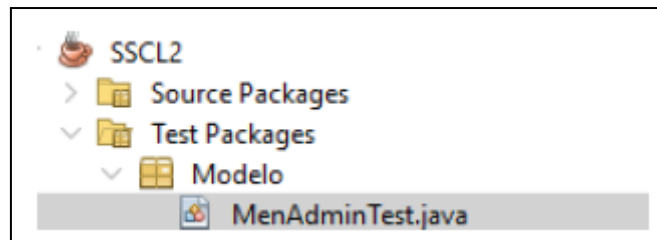
- Selecciona el framework "JUnit4" en el formulario que se despliega.
- Marca las casillas que se muestran en la captura de pantalla proporcionada.

- Confirma la configuración haciendo clic en "OK".



5. Creación de Paquete y Clase de Pruebas

- Después de confirmar la configuración, se creará automáticamente un paquete y una clase de pruebas para realizar el testeo.
- Asegúrate de que la clase creada tenga el nombre "MenAdminTest".



6. Importación de Anotaciones de Prueba

- En la clase MenAdminTest, importa la clase necesaria para utilizar las anotaciones de prueba de JUnit. Debería ser algo como: `import org.junit.Test;`

7. Implementación de Métodos de Prueba

- Define los métodos de prueba dentro de la clase MenAdminTest.
- Cada método de prueba debe ser anotado con `@Test` para que JUnit los reconozca como pruebas unitarias.

8. Implementación de los Métodos de Prueba

- Implementa los métodos de prueba con los escenarios adecuados para evaluar la funcionalidad de la clase MenAdmin.
- Asegúrate de cubrir diferentes casos de entrada y salida esperada.

```
package Modelo;

import org.junit.Test;
import static org.junit.Assert.*;

public class MenAdminTest {

    @Test
    public void testVerificarCredenciales_CorrectCredentials_ReturnsTrue() {
        // Se crea una instancia de la clase MenAdmin para realizar las pruebas.
        MenAdmin menAdmin = new MenAdmin();

        // Se llama al método verificarCredenciales con credenciales correctas y se guarda el resultado.
        boolean result = menAdmin.verificarCredenciales("JavierCevallos", "129062023");

        // Se verifica si el resultado es verdadero.
        assertTrue(result);
    }

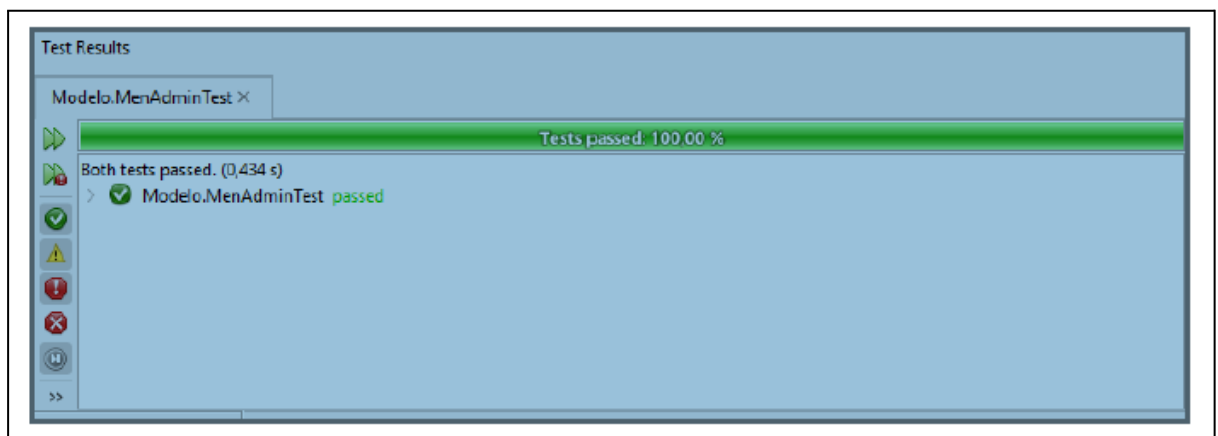
    @Test
    public void testVerificarCredenciales_IncorrectCredentials_ReturnsFalse() {
        // Se crea una instancia de la clase MenAdmin para realizar las pruebas.
        MenAdmin menAdmin = new MenAdmin();

        // Se llama al método verificarCredenciales con credenciales incorrectas y se guarda el resultado.
        boolean result = menAdmin.verificarCredenciales("AldoSaula", "12345");

        // Se verifica si el resultado es falso.
        assertFalse(result);
    }
}
```

9. Ejecución de las Pruebas

- Ejecuta las pruebas unitarias utilizando la funcionalidad proporcionada por tu IDE o mediante la línea de comandos.
- Verifica que todas las pruebas se ejecuten correctamente y que los resultados coincidan con lo esperado.



10. Análisis de Resultados

- Revisa los resultados de las pruebas para asegurarte de que todas hayan pasado con éxito.
- Analiza cualquier error o fallo que se haya producido y realiza los ajustes necesarios en el código de la clase MenAdmin si es necesario.

11. Mantenimiento de las Pruebas

- A medida que se realicen cambios en la clase MenAdmin, actualiza y ajusta las pruebas unitarias correspondientes para garantizar que sigan siendo válidas y efectivas.

12. Documentación

- Documenta adecuadamente las pruebas unitarias realizadas, incluyendo descripciones claras de los escenarios de prueba y los resultados obtenidos.

Conclusión

Al seguir esta guía paso a paso, podrás llevar a cabo pruebas unitarias efectivas utilizando JUnit4 para la clase MenAdmin, lo que te ayudará a identificar y corregir errores de manera proactiva, asegurando la calidad y la fiabilidad del código.