

**Universidad ORT Uruguay**  
**Facultad de Ingeniería**  
**Escuela de Tecnología**

**OBLIGATORIO 1**



Juan Andrés Olivera – N° Estudiante 348856



Andrés Álvarez Juncal – N° Estudiante 295384

Grupo N3B

Materia: Algoritmos y Estructuras de Datos

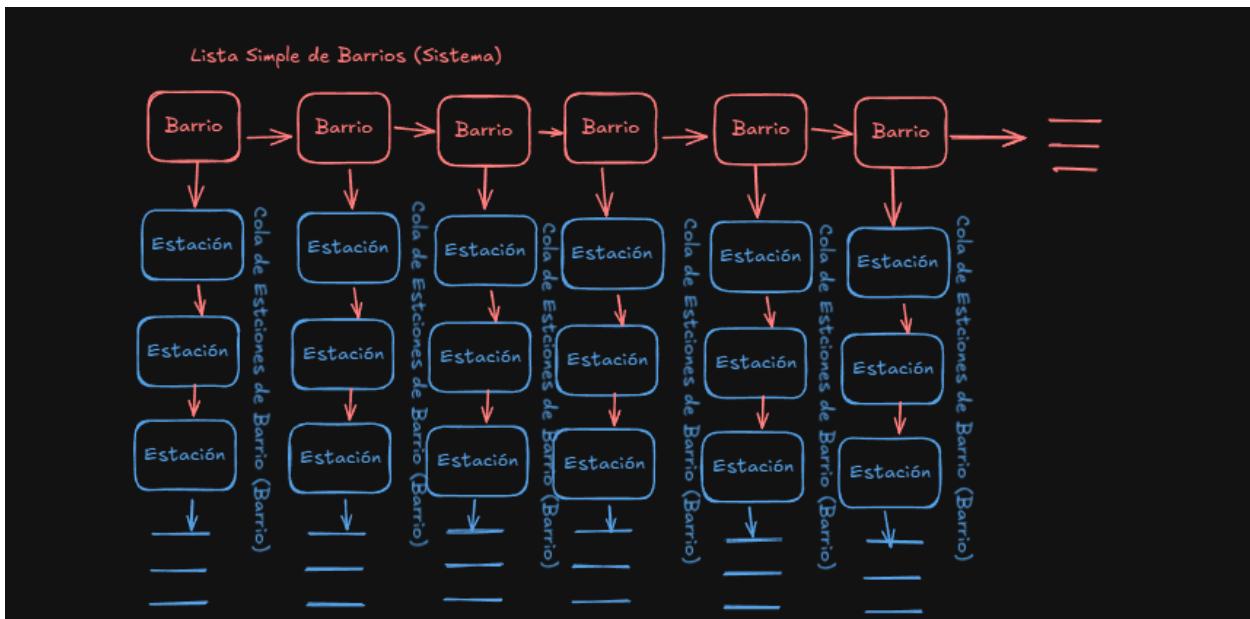
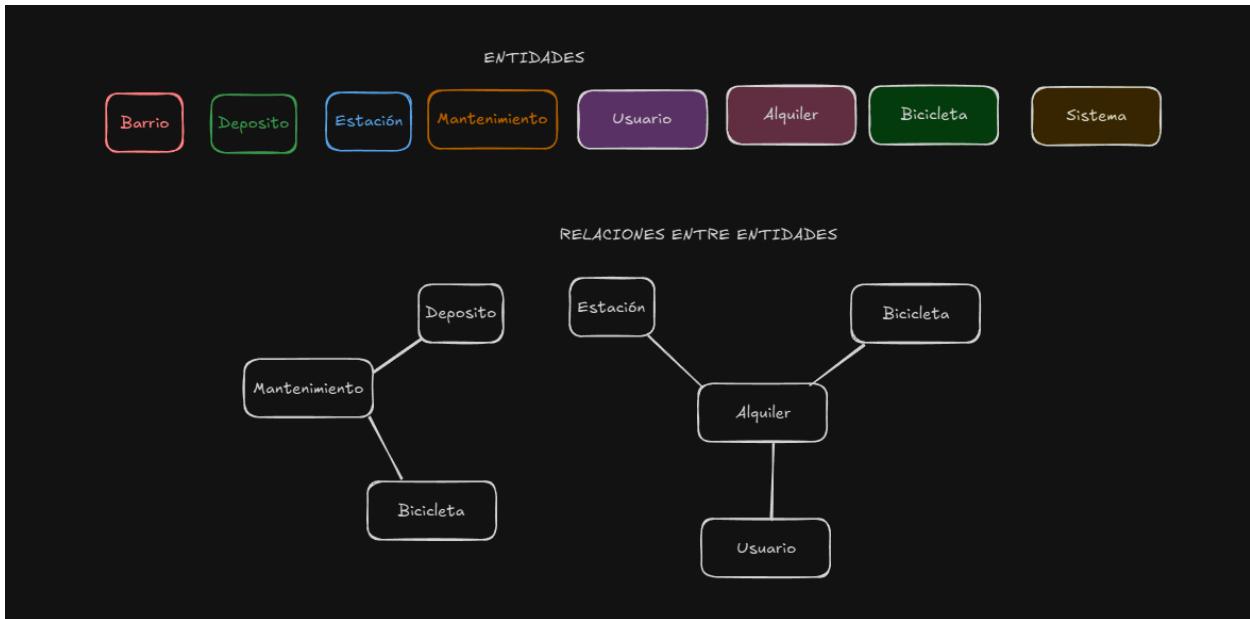
Profesor: Hugo Cepeda

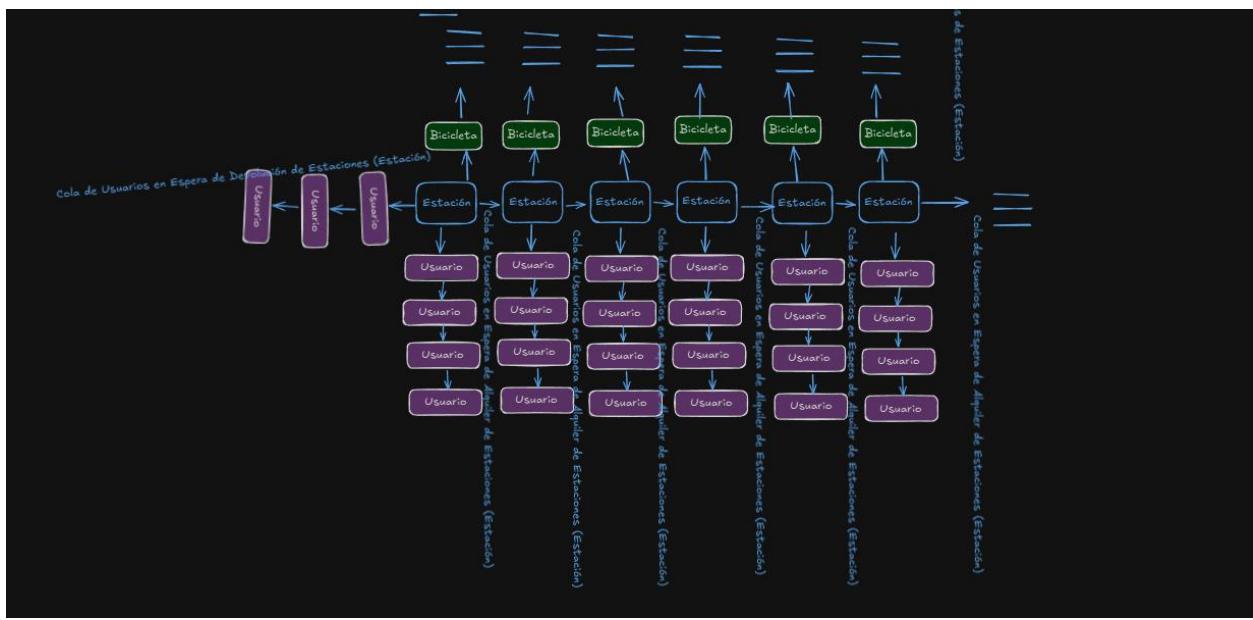
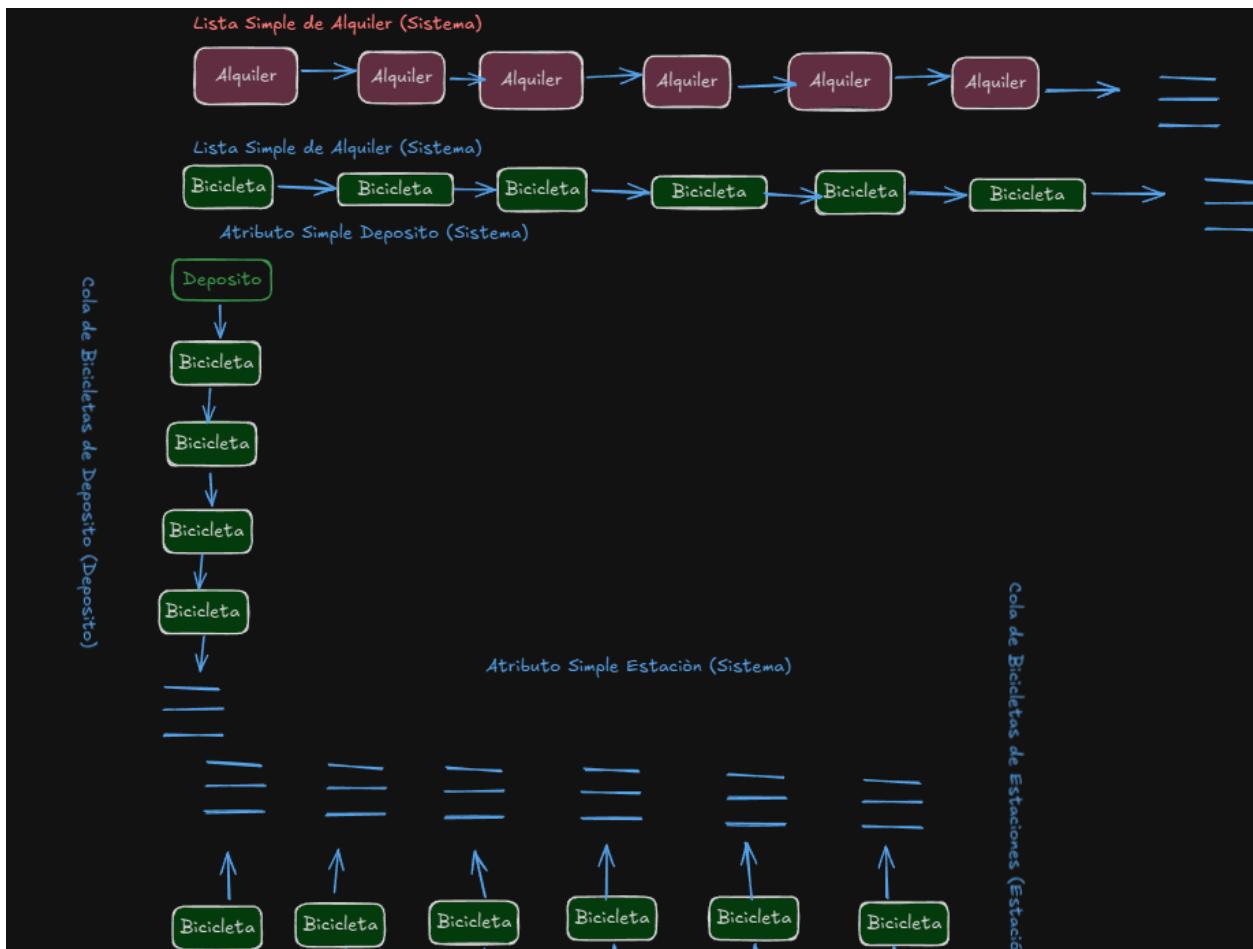
## Contenido

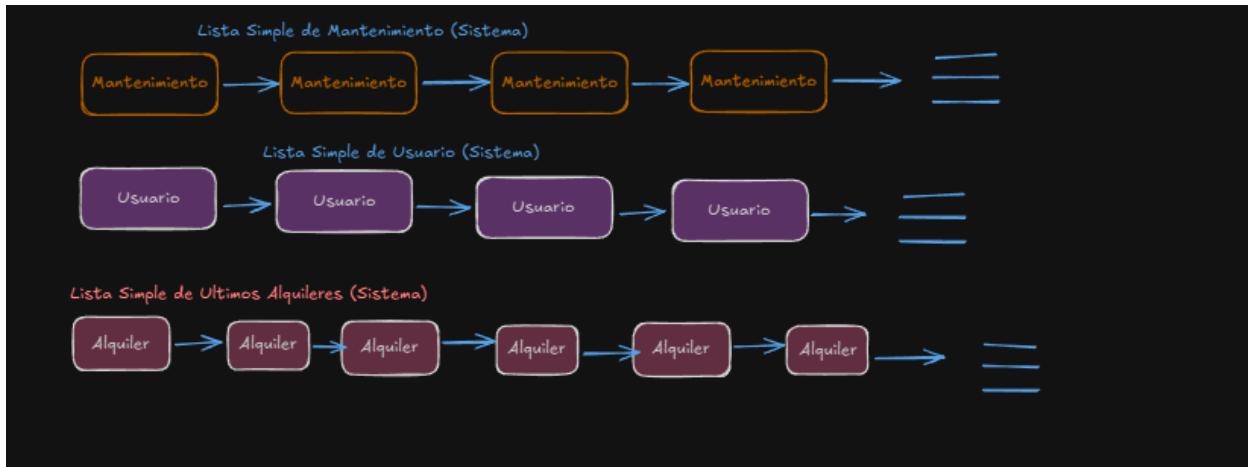
<b>OBLIGATORIO 1 .....</b>	1
<b>Diagrama:.....</b>	4
<b>ENTIDADES.....</b>	6
<b>Justificación:.....</b>	6
<b>PRUEBAS: .....</b>	7
2.1: Ok – Se pudo inicializar el Sistema.....	7
2.2: Ok – Se pudo registrar en el Sistema .....	8
2.2: Error 1 .....	8
2.2: Error 2 .....	9
2.2: Error 3 .....	10
2.3: Ok.....	10
2.3: Error 1 .....	11
2.3: Error 2 .....	12
2.3: Error 3 .....	12
2.4: Ok.....	13
2.4: Error 1 .....	13
2.4: Error 2 .....	14
2.4: Error 3 .....	14
2.4: Error 4 .....	15
2.5: Ok.....	15
2.5: Error 1 .....	16
2.5: Error 2 .....	16
2.5: Error 3 .....	17
2.5: Error 4 .....	17
2.6: Ok.....	18
2.6: Error 1 .....	18
2.6: Error 2 .....	19
2.6: Error 3 .....	19

3.1: Ok .....	20
3.1: Error 1 .....	21
3.1: Error 2 .....	21
3.1: Error 3 .....	22
3.2: Ok – Listar Usuario Vacío .....	22
3.2: Ok – 2 – Listar Usuario Solo Un Usuario.....	23
3.2: Ok – 3 – Listar Usuario Ingresado Ordenado .....	23
3.3: Ok.....	24
3.4.1:.....	24
3.4.2.....	25
3.4.4:.....	26

## Diagrama:







## ENTIDADES

- **Usuario** : Se usa en registrar/obtener/listar usuarios y en alquiler/devolución
- **Bicicleta** : Se usa en registrar bici, mantenimiento, reparar, asignar, listar, etc..
- **Estación** : Se usa para “anclajes” para bicis, cola de espera de alquiler (usuarios) y cola de espera de anclaje (usuarios que quieren devolver sin espacio)
- **Depósito**: Se usa para las bicis cuando no están asignadas a ninguna estación ni alquiladas.
- **Mantenimiento**: Se usa para Facilita registrar “poner en mantenimiento” y “reparar” manteniendo trazabilidad mínima.
- **Alquiler**: modela un préstamo de una bicicleta a un usuario por un período de tiempo. Es el registro que conecta Usuario con Bicicleta y con Estación
- **Barrio**: Agrupa estaciones cada estación pertenece a un barrio.

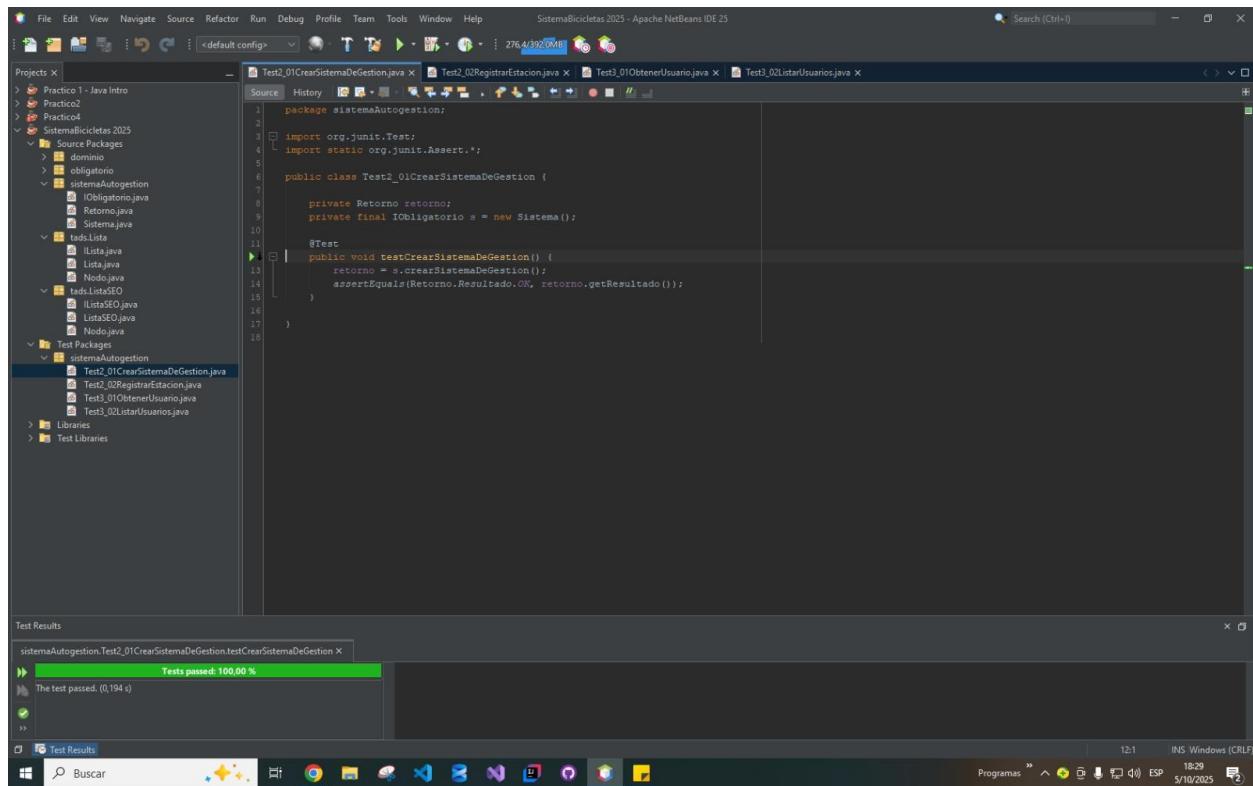
## Justificación:

- Un **Barrio** puede tener el **Depósito** y/o una o varias **Estaciones**.
- El **Depósito** contiene un número muy grande (no definido ni alcanzable) de **Bicicletas**, las cuales se encuentran únicamente en reparación. Una vez reparadas, son enviadas a una **Estación**.
- Una **Estación** contiene un número finito (definido y alcanzable) de **Bicicletas** en sus anclajes. Puede estar sin bicicletas y, en ese caso, cuenta con una lista de **Usuarios** en espera para alquilar una. También puede tener sus anclajes completos; en esa situación, los **Usuarios** que desean devolver una bicicleta deben esperar, por lo que la estación mantiene una lista de espera para retornos. Además, cada estación guarda un registro de usuarios que servirá para reportes y estadísticas futuras.

- La clase **Sistema** incluye dos listas de **Mantenimiento** para poder cumplir con el punto 2.11, que requiere devolver los últimos mantenimientos realizados. De esta forma, al solicitar los últimos  $n$  mantenimientos, se eliminan los registros más antiguos.
- La clase **Alquiler** se relaciona con **Estación**, **Usuario** y con la primera **Bicicleta** disponible para realizar la operación.
- La clase **Mantenimiento** se relaciona con una **Bicicleta** y la traslada al **Depósito** para su reparación.

## PRUEBAS:

### 2.1: Ok – Se pudo inicializar el Sistema.



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and the title 'SistemasBicicletas 2025 - Apache NetBeans IDE 25'. Below the menu is a toolbar with various icons. The main area is divided into several panes: 'Projects' on the left listing 'Práctico 1 - Java Intro', 'Práctico2', 'Práctico4', and 'SistemasBicicletas 2025' (selected), which contains 'Source Packages' (dominio, obligatorio, sistemasAutogestion), 'Test Packages' (sistemasAutogestion), and 'Libraries'. The central pane displays the Java code for 'Test2\_01CrearSistemaDeGestion.java'. The code imports org.junit.Test and static org.junit.Assert.\*; and defines a public class Test2\_01CrearSistemaDeGestion with a single test method: @Test public void testCrearSistemaDeGestion() { retorno = s.crearSistemaDeGestion(); assertEquals(Retorno.Resultado.OK, retorno.getResultado()); }. The bottom pane, 'Test Results', shows a summary for 'sistemasAutogestion.Test2\_01CrearSistemaDeGestion.testCrearSistemaDeGestion': 'Tests passed: 100,00 %' and 'The test passed. (0,194 s)'. The status bar at the bottom right shows '12:1 INS Windows (CRLF)', 'Programas', '18:29', 'ESP', and the date '5/10/2025'.

```

package sistemaAutogestion;
import org.junit.Test;
import static org.junit.Assert.*;
public class Test2_01CrearSistemaDeGestion {
    private Retorno retorno;
    private final IObligatorio s = new Sistema();
    @Test
    public void testCrearSistemaDeGestion() {
        retorno = s.crearSistemaDeGestion();
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    }
}

```

## 2.2: Ok – Se pudo registrar en el Sistema

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profiler, Team, Tools, Window, Help, and SistemaBicicletas 2025 - Apache NetBeans IDE 25. The bottom status bar shows the date as 5/10/2025 and the time as 18:11. The main area displays the code for `Test2_02RegistrarEstacion.java`. The code is a JUnit test for registering a station. It imports org.junit.Before, org.junit.Test, and static org.junit.Assert.\*. The class `sistemaAutogestion` contains a method `Test2_02RegistrarEstacion`. The test cases include `registrarEstacionOk()` and `registrarEstacionError01()`, which check for success and various error conditions respectively. Below the code editor is a "Test Results" window showing a single test case `sistemaAutogestion.Test2_01CrearSistemaDeGestion.testCrearSistemaDeGestion` has passed 100.00%. The taskbar at the bottom shows various application icons.

```
package sistemaAutogestion;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
public class Test2_02RegistrarEstacion {
    private Retorno retorno;
    private final IObligatorio s = new Sistema();
    @Before
    public void setUp() {
        s.crearSistemaDeGestion();
    }
    @Test
    public void registrarEstacionOk() {
        retorno = s.registrarEstacion("Estacion01", "Centro", 5);
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    }
    @Test
    public void registrarEstacionError01() {
        retorno = s.registrarEstacion("", "Centro", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion(" ", "Centro", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion("Estacion01", " ", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion("Estacion01", "", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion(null, "Centro", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    }
}
```

## 2.2: Error 1

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profiler, Team, Tools, Window, Help, and SistemaBicicletas 2025 - Apache NetBeans IDE 25. The bottom status bar shows the date as 5/10/2025 and the time as 24:45. The main area displays the code for `Test2_02RegistrarEstacion.java`. The code is a JUnit test for registering a station. It imports org.junit.Before, org.junit.Test, and static org.junit.Assert.\*. The class `sistemaAutogestion` contains a method `Test2_02RegistrarEstacion`. The test cases include `registrarEstacionOk()` and `registrarEstacionError01()`, which check for success and various error conditions respectively. Below the code editor is a "Test Results" window showing a single test case `sistemaAutogestion.Test2_01CrearSistemaDeGestion.testCrearSistemaDeGestion` has passed 100.00%. The taskbar at the bottom shows various application icons.

```
public class Test2_02RegistrarEstacion {
    private Retorno retorno;
    private final IObligatorio s = new Sistema();
    @Before
    public void setUp() {
        s.crearSistemaDeGestion();
    }
    @Test
    public void registrarEstacionOk() {
        retorno = s.registrarEstacion("Estacion01", "Centro", 5);
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    }
    @Test
    public void registrarEstacionError01() {
        retorno = s.registrarEstacion("", "Centro", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion("Estacion01", " ", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion(" ", "Centro", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion("Estacion01", " ", " ");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion(null, "Centro", 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
        retorno = s.registrarEstacion("Estacion01", null, 5);
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    }
}
```

## 2.2: Error 2

The screenshot shows the Apache NetBeans IDE 25 interface with the following details:

- Title Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, SistemaBicicletas 2025 - Apache NetBeans IDE 25, Search (Ctrl+).
- Toolbar:** Standard icons for file operations.
- Project Explorer:** Shows the project structure under "Práctico 4" and "SistemaBicicletas 2025".
- Code Editor:** Displays the `Test2_02RegistrarEstacion.java` file with Java code for testing the `s.registerarEstacion` method. The code includes various assertions for different input parameters (e.g., empty strings, null, negative values) and error codes (e.g., `ERROR_1`, `ERROR_2`, `ERROR_3`).
- Test Results:** Shows the execution results for the test cases:
  - `sistemaAutogestion.Test2_02RegistrarEstacion.registrarEstacionError01`: Tests passed: 100.00 %
  - `sistemaAutogestion.Test2_02RegistrarEstacion.registrarEstacionOk`
  - `sistemaAutogestion.Test3_01ObtenerUsuario.obtenerUsuarioOk`
  - `sistemaAutogestion.Test3_01ObtenerUsuario.obtenerUsuarioError1`
  - `sistemaAutogestion.Test3_01ObtenerUsuario.obtenerUsuarioError2`
- Task List:** Shows the test results: "The test passed. (0,778 s)".
- Bottom Bar:** Icons for file operations, search, and system status (Programas, Windows, ESP, 18:48, 5/10/2025).

## 2.2: Error 3

The screenshot shows the NetBeans IDE interface with the following details:

- Projects:** Práctica 1 - Java Intro, Práctica2, Práctica4, SistemaBicicletas 2025.
- Source Packages:** sistemaAutogestion contains classes like Alquiler.java, AlquilerUsuarioBicicleta.java, Barrio.java, Bicicleta.java, Deposito.java, Estacion.java, Invalidable.java, Mantenimiento.java, Usuario.java, and several test classes (Test2\_02RegistrarEstacion.java, Test3\_01ObtenerUsuario.java, etc.).
- Code View:** The code for `Test2_02RegistrarEstacion.java` is displayed, showing various test cases for registering stations with different parameters (empty strings, null values, negative numbers) and asserting the expected error codes.
- Test Results:** A green bar indicates "Tests passed: 100.00 %". Below it, a message says "The test passed. (0,182 s)".
- System Tray:** Shows icons for network, battery, volume, and system status.
- Bottom Bar:** Includes "Programas", "INS Windows (CRLF)", "18:49", and a date/time stamp "5/10/2025".

## 2.3: Ok

The screenshot shows the NetBeans IDE interface with the following details:

- Projects:** Práctica 1 - Java Intro, Práctica2, Práctica4, SistemaBicicletas 2025.
- Source Packages:** sistemaAutogestion contains classes like Alquiler.java, AlquilerUsuarioBicicleta.java, Barrio.java, Bicicleta.java, Deposito.java, Estacion.java, Invalidable.java, Mantenimiento.java, Usuario.java, and several test classes (Test2\_02RegistrarUsuario.java, Test3\_01ObtenerUsuario.java, etc.).
- Code View:** The code for `Test2_02RegistrarUsuario.java` is displayed, showing test cases for registering users with valid and invalid IDs and asserting the expected success or error codes.
- Test Results:** A green bar indicates "Tests passed: 100.00 %". Below it, a message says "The test passed. (0,178 s)".
- System Tray:** Shows icons for network, battery, volume, and system status.
- Bottom Bar:** Includes "Programas", "INS Windows (CRLF)", "18:59", and a date/time stamp "5/10/2025".

### 2.3: Error 1

The screenshot shows the Apache NetBeans IDE 25 interface with the following details:

- File Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** SistemaBicicletas 2025 - Apache NetBeans IDE 25
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Explorer:** Shows the project structure:
  - Práctico 1 - Java Intro
  - Práctico2
  - Práctico4
  - SistemaBicicletas 2025
    - Source Packages
      - dominio
        - Alquiler.java
        - AlquilerUsuarioBicicleta.java
        - Banco.java
        - Bicicleta.java
        - Deposito.java
        - Estacion.java
        - Validable.java
        - Mantenimiento.java
        - Usuario.java
      - obligatorio
      - sistemaAutogestion
        - IObligatorio.java
        - Retorno.java
        - Sistema.java
      - tads.Lista
        - Lista.java
        - Nodo.java
      - tads.ListaSE
        - ListaSE.java
        - ListaSEObjeto.java
        - Nodo.java
      - Test Packages
        - sistemaAutogestion
          - Test2\_01CrearSistemaDeGestion.java
          - Test2\_02RegistrarEstacion.java
          - Test2\_02RegistrarUsuario.java
          - Test3\_01ObtenerUsuario.java
          - Test3\_02ListarUsuarios.java
    - Libraries
    - Test Libraries
  - Code Editor:** The main editor window displays the `Test2_02RegistrarUsuario.java` file. The code contains several test methods using JUnit annotations (`@Test`) and assertions (`assertEquals`) to verify the behavior of the `s.registerarUsuario` method.
  - Test Results:** A separate window titled "Test Results" shows the execution status of the tests. It includes a summary bar with the message "Tests passed: 100.00 %". Below the bar, it says "The test passed. (0,188 s)".
  - System Bar:** Shows the taskbar with various pinned icons and the system clock indicating 18:59 on 5/10/2025.

## 2.3: Error 2

The screenshot shows the NetBeans IDE interface with the following details:

- Projects:** Práctica 1 - Java Intro, Práctica2, Práctica3, Práctica4, SistemaBicicletas 2025.
- Source Packages:** sistemaAutogestion contains classes like Alquiler.java, Barrio.java, Bicicleta.java, Deposito.java, Estacion.java, Invalidable.java, Mantenimiento.java, Usuario.java, and several List and Nodo classes.
- Test Packages:** sistemaAutogestion contains tests for Test2\_01ClearSistemaDeGestion.java, Test2\_02RegistrarEstacion.java, Test2\_02RegistrarUsuario.java, Test3\_01ObtenerUsuario.java, and Test3\_02ListarUsuarios.java.
- Code Editor:** The code editor displays the Java code for `Test2_02RegistrarUsuario.java`. It includes several `@Test` blocks for testing user registration with various parameters (empty strings, null, invalid IDs).
- Test Results:** A separate window shows the test results for `sistemaAutogestion.Test2_02RegistrarEstacion.registrarEstacionError01`, `sistemaAutogestion.Test2_02RegistrarEstacion.registrarEstacionOk`, `sistemaAutogestion.Test3_01ObtenerUsuario.obtenerUsuarioOk`, `sistemaAutogestion.Test3_01ObtenerUsuario.obtenerUsuarioError1`, and `sistemaAutogestion.Test3_02ListarUsuarios.listarUsuariosOk`. All tests passed with 100.00% success rate.

## 2.3: Error 3

The screenshot shows the NetBeans IDE interface with the following details:

- Projects:** Práctica 1 - Java Intro, Práctica2, Práctica3, Práctica4, SistemaBicicletas 2025.
- Source Packages:** sistemaAutogestion contains classes like Alquiler.java, Barrio.java, Bicicleta.java, Deposito.java, Estacion.java, Invalidable.java, Mantenimiento.java, Usuario.java, and several List and Nodo classes.
- Test Packages:** sistemaAutogestion contains tests for Test2\_01ClearSistemaDeGestion.java, Test2\_02RegistrarEstacion.java, Test2\_02RegistrarUsuario.java, Test3\_01ObtenerUsuario.java, and Test3\_02ListarUsuarios.java.
- Code Editor:** The code editor displays the Java code for `Test2_02RegistrarUsuario.java`. It includes several `@Test` blocks for testing user registration with various parameters (empty strings, null, invalid IDs).
- Test Results:** A separate window shows the test results for `sistemaAutogestion.Test2_02RegistrarEstacion.registrarEstacionError01`, `sistemaAutogestion.Test2_02RegistrarEstacion.registrarEstacionOk`, `sistemaAutogestion.Test3_01ObtenerUsuario.obtenerUsuarioOk`, `sistemaAutogestion.Test3_01ObtenerUsuario.obtenerUsuarioError1`, and `sistemaAutogestion.Test3_02ListarUsuarios.listarUsuariosOk`. All tests passed with 100.00% success rate.

## 2.4: Ok

The screenshot shows the Apache NetBeans IDE interface with the title "SistemaBicicletas 2025 - Apache NetBeans IDE 25". The left pane displays the project structure under "Source Packages" for the "SistemaBicicletas 2025" module. The right pane shows the code for the test class "Test2\_04RegistrarBicicleta.java". The code contains several test methods using annotations like @Test and @Before. Below the code editor is the "Test Results" panel, which shows a single test named "sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarBicicletaOk" has passed. The status bar at the bottom indicates the test passed in 0.182 seconds.

```
public class Test2_04RegistrarBicicleta {
    private Retorno retorno;
    private final IObligatorio s = new Sistema();

    @Before
    public void setUp() {
        s.crearSistemaDeGestion();
    }

    @Test
    public void registrarBicicletaOk() {
        retorno = s.registrarBicicleta("123456", "URBANA");
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());

        retorno = s.registrarBicicleta("XZT789", "MOUNTAIN");
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());

        retorno = s.registrarBicicleta("ELE456", "ELECTRICA");
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    }

    @Test
    public void registrarEstacionBicicletaOk() {
        retorno = s.registrarBicicleta("", "URBANA");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

        retorno = s.registrarBicicleta(" ", "URBANA");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

        retorno = s.registrarBicicleta("123456", " ");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

        retorno = s.registrarBicicleta(null, "URBANA");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    }
}
```

Test Results

sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarBicicletaOk | sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarEstacionBicicletaOk | Tests passed: 100.00 %

The test passed. (0.182 s)

18:1 INS Windows (CRLF) 5/10/2025

## 2.4: Error 1

The screenshot shows the Apache NetBeans IDE interface with the title "SistemaBicicletas 2025 - Apache NetBeans IDE 25". The left pane displays the project structure under "Source Packages" for the "SistemaBicicletas 2025" module. The right pane shows the code for the test class "Test2\_04RegistrarBicicleta.java". The code is identical to the one in the previous screenshot. Below the code editor is the "Test Results" panel, which shows the same test has passed. The status bar at the bottom indicates the test passed in 0.193 seconds.

```
public class Test2_04RegistrarBicicleta {
    private Retorno retorno;
    private final IObligatorio s = new Sistema();

    @Before
    public void setUp() {
        s.crearSistemaDeGestion();
    }

    @Test
    public void registrarBicicletaOk() {
        retorno = s.registrarBicicleta("123456", "URBANA");
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());

        retorno = s.registrarBicicleta("XZT789", "MOUNTAIN");
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());

        retorno = s.registrarBicicleta("ELE456", "ELECTRICA");
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    }

    @Test
    public void registrarEstacionBicicletaOk() {
        retorno = s.registrarBicicleta("", "URBANA");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

        retorno = s.registrarBicicleta(" ", "URBANA");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

        retorno = s.registrarBicicleta("123456", " ");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

        retorno = s.registrarBicicleta(null, "URBANA");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    }
}
```

Test Results

sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarBicicletaOk | sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarEstacionBicicletaOk | Tests passed: 100.00 %

The test passed. (0.193 s)

19:14 INS Windows (CRLF) 5/10/2025

## 2.4: Error 2

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects:** Práctica1 - Java Intro, Práctica2, Práctica4, SistemaBicicletas 2025.
- Source Packages:** sistemaAutogestion (containing sub-packages domino, obligatorio, and sistemaAutogestion), and Test Packages (containing sub-packages sistemaAutogestion).
- Code Editor:** Displays Java test code for registering bicycles. The code includes various assertions for different input scenarios (e.g., empty strings, null values, specific IDs like "12345678", "123456789", and "1234567890").
- Test Results:** Shows a summary of test results for the selected test class. It indicates 100.00% tests passed in 0.189 seconds.
- System Bar:** Shows the taskbar with various application icons and the system clock at 19:16 on 5/10/2025.

## 2.4: Error 3

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects:** Práctica1 - Java Intro, Práctica2, Práctica4, SistemaBicicletas 2025.
- Source Packages:** sistemaAutogestion (containing sub-packages domino, obligatorio, and sistemaAutogestion), and Test Packages (containing sub-packages sistemaAutogestion).
- Code Editor:** Displays Java test code for registering bicycles. The code includes various assertions for different input scenarios (e.g., empty strings, null values, specific IDs like "12345678", "123456789", and "1234567890").
- Test Results:** Shows a summary of test results for the selected test class. It indicates 100.00% tests passed in 0.185 seconds.
- System Bar:** Shows the taskbar with various application icons and the system clock at 19:17 on 5/10/2025.

## 2.4: Error 4

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects:** Shows several Java projects: Práctico 1 - Java Intro, Práctico2, Práctico4, and SistemaBicicletas 2025.
- SistemaBicicletas 2025:** Contains Source Packages like `sistemaAutogestion` and `tads`, and Test Packages like `sistemaAutogestion` and `tads`.
- Code Editor:** Displays a test class `Test2_04RegistrarBicicleta.java` with several `@Test` annotations and corresponding test methods.
- Test Results:** Shows a green bar indicating "Tests passed: 100.00 %".
- System Bar:** Shows the date and time as 5/10/2025 19:17, and the operating system as INS Windows (CRLF).

## 2.5: Ok

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects:** Shows Java projects: Práctico 1 - Java Intro, Práctico2, Práctico4, and SistemaBicicletas 2025.
- SistemaBicicletas 2025:** Contains Source Packages like `sistemaAutogestion` and `tads`, and Test Packages like `sistemaAutogestion` and `tads`.
- Code Editor:** Displays a test class `Test2_05MarcarMantenimiento.java` with several `@Test` annotations and corresponding test methods.
- Test Results:** Shows a green bar indicating "Tests passed: 100.00 %".
- System Bar:** Shows the date and time as 5/10/2025 19:40, and the operating system as INS Windows (CRLF).

## 2.5: Error 1

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** SistemaBicicletas 2025 - Apache NetBeans IDE 25
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer (Projects X):**
  - Practico 1 - Java Intro
  - Practico2
  - Practico4
  - SistemaBicicletas 2025
    - Source Packages
      - domingo
        - Alquiler.java
        - AlquilerUsuarioBicicleta.java
        - Bandeja.java
        - Bicicleta.java
        - Deposito.java
        - Estacion.java
        - IValidable.java
        - Mantenimiento.java
        - Usuario.java
      - obligatorio
        - sistemaAutogestion
          - ILista.java
          - Lista.java
          - Nodo.java
        - Tads-Lista
          - IILista.java
          - Lista.java
          - Nodo.java
        - Tads-ListaSO
          - ListaSEObjava
          - ListaSEObjava
          - Nodo.java
      - Test Packages
        - sistemaAutogestion
          - Test0\_0CrearSistemaDeGestion.java
          - Test0\_0RegistrarEstacion.java
          - Test0\_0RegistrarUsuario.java
          - Test2\_0RegistrarBicicleta.java
          - Test2\_0RegistrarBicicletaOk.java
          - Test2\_0MarcarMantenimiento.java
          - Test3\_01HacerLlamada.java
          - Test3\_02ListarUsuarios.java
        - Test0\_0RegistrarEstacionOk.java
        - Test0\_0RegistrarUsuarioOk.java
        - Test0\_0RegistrarBicicletaOk.java
        - Test0\_0HacerLlamadaOk.java
        - Test0\_0ListarUsuariosOk.java
  - Code Editor:** The current file is Test2\_05MarcarMantenimiento.java, containing test cases for the MarcarMantenimiento method.
  - Test Results:** The results show 100.00% passed tests.
  - System Status:** Shows the date as 5/10/2025 and the time as 19:41.

## 2.5: Error 2

The screenshot shows the Apache NetBeans IDE interface with the following details:

- File Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** Sistemanbicicletas 2025 - Apache NetBeans IDE 25.
- Project Explorer (Projects):**
  - Practico 1 - Java Intro
  - Practico2
  - Practico4
  - SistemaBicicletas 2025
    - Source Packages
      - domingo
        - Alquiler.java
        - AlquilerUsuarioBicicleta.java
        - Banijo.java
        - Bicicleta.java
        - Bicicleta.java
        - Estacion.java
        - Validable.java
        - Mantenimiento.java
        - Usuario.java
      - obligatorio
        - sistemaAutogestion
          - Obligatorio.java
          - Retorno.java
          - Sistema.java
        - tads.Lista
          - Lista.java
          - Lista.java
          - Nodo.java
        - tads.ListaSO
          - ListadoObjeto.java
          - ListaSO.java
          - Nodo.java
  - Code Editor:** Displays Java code for `Test2_05MarcarMantenimiento.java`. The code contains several test methods using `@Test` annotations and `assertEquals` assertions to verify the behavior of the `s.marcarEnMantenimiento` method.
  - Test Results:** Shows the results of the executed tests:
    - testsAutogestion.Test2\_04RegistrarBicicleta.registrarBicicletaOk X
    - sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarEstacionBicicleta01 X
    - sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarEstacionBicicleta02 X
    - sistemaAutogestion.Test2\_04RegistrarBicicleta.registrarEstacionBicicleta03 X

Tests passed: 100.00 %

The test passed. (0.19 s)
  - Bottom Bar:** Windows Taskbar showing various application icons like File Explorer, Edge, and File Manager.
  - Status Bar:** Shows the date and time as 5/10/2025 19:42.

## 2.5: Error 3

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** SistemaBicicletas 2025 - Apache NetBeans IDE 2025
- Toolbar:** File, Edit, View, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer (Projects X):**
  - Practico 1 - Java Intro
  - Practico 2
  - Practico 4
  - SistemaBicicletas 2025
    - Source Packages
      - domingo
        - Alquiler.java
        - AlquilerUsuarioBicicleta.java
        - Bicicleta.java
        - Deposito.java
        - Estacion.java
        - IValidable.java
        - Mantenimiento.java
        - Usuario.java
      - obligatorio
        - IObligatorio.java
        - Retorno.java
        - Sistema.java
      - tadsLista
        - Lista.java
        - Nodo.java
      - tadsListaSEO
        - ListaSEOD.java
        - ListaSEO.java
        - Nodo.java
      - Test Packages
        - sistemaAutogestion
          - CrearSistemaDeGestion.java
          - Test2\_01RegistrarEstacion.java
          - Test2\_02RegistrarUsuario.java
          - Test2\_03RegistrarBicicleta.java
          - Test2\_04RegistrarBicicleta.java
          - Test2\_05MarcarMantenimiento.java
          - Test2\_06ObtenerUsuario.java
          - Test2\_07ListarEstacion.java
  - Code Editor:** The main editor window displays the `Test2_05MarcarMantenimiento.java` file. The code contains several test methods using the `@Test` annotation to verify the functionality of the `s.marcarEnMantenimiento` and `s.registrarBicicleta` methods.
  - Test Results:** A separate window titled "Test Results" shows the outcome of the tests:
    - Test: sistemaAutogestion.Test2\_05MarcarMantenimiento.MarcarMantenimiento03 X
    - Status: Tests passed: 100.00 %
    - Message: The test passed. (0.211 s)
  - System Tray:** Shows icons for battery, signal strength, and system status.
  - Bottom Bar:** Includes "Buscar" (Search), "Output", and system status indicators for Windows 10/2025.

## 2.5: Error 4

## 2.6: Ok

The screenshot shows the Apache NetBeans IDE 25 interface with the following details:

- Projects:** Practice 1 - Java Intro, Practice2, Practice4, SistemaBicicletas 2025.
- Source Packages:** sistema, obligatorio, sistemaAutogestion, Test Packages.
- Code Editor:** The code for `Test2_06RepararBicicleta.java` is displayed, showing various test methods like `repararBicicletasOk()`, `repararBicicleta01()`, and `repararBicicleta02()`.
- Test Results:** A green bar at the bottom indicates "Tests passed: 100.00 %". Below it, a message says "The test passed. (0.187 s)".
- System Bar:** Shows the date and time as 10:21 on 6/10/2025.

## 2.6: Error 1

The screenshot shows the Apache NetBeans IDE 25 interface with the following details:

- Projects:** Practice 1 - Java Intro, Practice2, Practice4, SistemaBicicletas 2025.
- Source Packages:** sistema, obligatorio, sistemaAutogestion, Test Packages.
- Code Editor:** The code for `Test2_06RepararBicicleta.java` is displayed, showing various test methods like `repararBicicletasOk()`, `repararBicicleta01()`, and `repararBicicleta02()`.
- Test Results:** A red bar at the bottom indicates "Tests passed: 100.00 %". Below it, a message says "The test passed. (0.196 s)".
- System Bar:** Shows the date and time as 10:21 on 6/10/2025.

## 2.6: Error 2

The screenshot shows the Apache NetBeans IDE 25 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and SistemaBicletas 2025 - Apache NetBeans IDE 25. The left sidebar displays the Projects tree with several Java packages: Practico 1 - Java Intro, Practico2, Practico4, and SistemaBicletas 2025. The SistemaBicletas 2025 package contains Source Packages (Alquiler, Bicicleta, Deposito, Estacion, Invalidable, Mantenimiento, Usuario) and Test Packages (Obligatorio, sistemaAutogestion, TadsList, TadsListEO, Test Packages). The Source Packages folder for SistemaBicletas 2025 contains the Sistema.java file, which is open in the main editor window. The code in Sistema.java includes several @Test annotations for the repairBicicleta method. Below the editor is the Test Results panel, which shows a green bar indicating "Tests passed: 100.00 %". The status bar at the bottom right shows the date and time as 6/10/2025 10:21.

```
26     assertEquals(Retorno.Resultado.OK, retorno.getResultado());
27 }
28
29 @Test
30 public void repararBicicleta01() {
31     retorno = s.repararBicicleta("");
32     assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
33
34     retorno = s.repararBicicleta(" ");
35     assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
36
37     retorno = s.repararBicicleta(null);
38     assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
39 }
40
41 @Test
42 public void repararBicicleta02() {
43     retorno = s.repararBicicleta("123");
44     assertEquals(Retorno.Resultado.ERROR_2, retorno.getResultado());
45
46     retorno = s.repararBicicleta("12");
47     assertEquals(Retorno.Resultado.ERROR_2, retorno.getResultado());
48 }
49
50 @Test
51 public void repararBicicleta03() {
52     retorno = s.repararBicicleta("123456");
53     assertEquals(Retorno.Resultado.ERROR_3, retorno.getResultado());
54 }
```

## 2.6: Error 3

The screenshot shows the Apache NetBeans IDE 25 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and SistemaBicletas 2025 - Apache NetBeans IDE 25. The left sidebar displays the Projects tree with several Java packages: Practico 1 - Java Intro, Practico2, Practico4, and SistemaBicletas 2025. The SistemaBicletas 2025 package contains Source Packages (Alquiler, Bicicleta, Deposito, Estacion, Invalidable, Mantenimiento, Usuario) and Test Packages (Obligatorio, sistemaAutogestion, TadsList, TadsListEO, Test Packages). The Source Packages folder for SistemaBicletas 2025 contains the Sistema.java file, which is open in the main editor window. The code in Sistema.java includes several @Test annotations for the repairBicicleta method. Below the editor is the Test Results panel, which shows a green bar indicating "Tests passed: 100.00 %". The status bar at the bottom right shows the date and time as 6/10/2025 10:22.

```
25     retorno = s.repararBicicleta("456789");
26     assertEquals(Retorno.Resultado.OK, retorno.getResultado());
27 }
28
29 @Test
30 public void repararBicicleta01() {
31     retorno = s.repararBicicleta("");
32     assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
33
34     retorno = s.repararBicicleta(" ");
35     assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
36
37     retorno = s.repararBicicleta(null);
38     assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
39 }
40
41 @Test
42 public void repararBicicleta02() {
43     retorno = s.repararBicicleta("123");
44     assertEquals(Retorno.Resultado.ERROR_2, retorno.getResultado());
45
46     retorno = s.repararBicicleta("12");
47     assertEquals(Retorno.Resultado.ERROR_2, retorno.getResultado());
48 }
49
50 @Test
51 public void repararBicicleta03() {
52     retorno = s.repararBicicleta("123456", "URBANA");
53     assertEquals(Retorno.Resultado.OK, retorno.getResultado());
54
55     retorno = s.repararBicicleta("123456");
56     assertEquals(Retorno.Resultado.ERROR_3, retorno.getResultado());
57 }
```

### 3.1: Ok

The screenshot shows the Apache NetBeans IDE 25 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar displays "SistemaBicicletas 2025 - Apache NetBeans IDE 25". The left sidebar shows the project structure under "Projects X". The main editor window contains the Java code for "Test3\_01ObtenerUsuario.java". The code uses JUnit annotations like @Before and @Test. The bottom tab bar shows several test results, all of which have passed. The status bar at the bottom right indicates the date and time as "6/10/2025 10:24".

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

public class Test3_01ObtenerUsuario {

    private Retorno retorno;
    private final IObligatorio s = new Sistema();

    @Before
    public void setUp() {
        s.crearSistemaDeGestion();
    }

    @Test
    public void obtenerUsuarioOk() {
        s.registrarUsuario("12345678", "Usuario01");
        retorno = s.obtenerUsuario("12345678");
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    }

    @Test
    public void obtenerUsuarioError1() {
        s.registrarUsuario("", "Usuario1");
        assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    }

    @Test
    public void obtenerUsuarioError01() {
        //Completar
    }

    @Test
    public void obtenerUsuarioError02() {
        //Completar
    }
}
```

Test Results

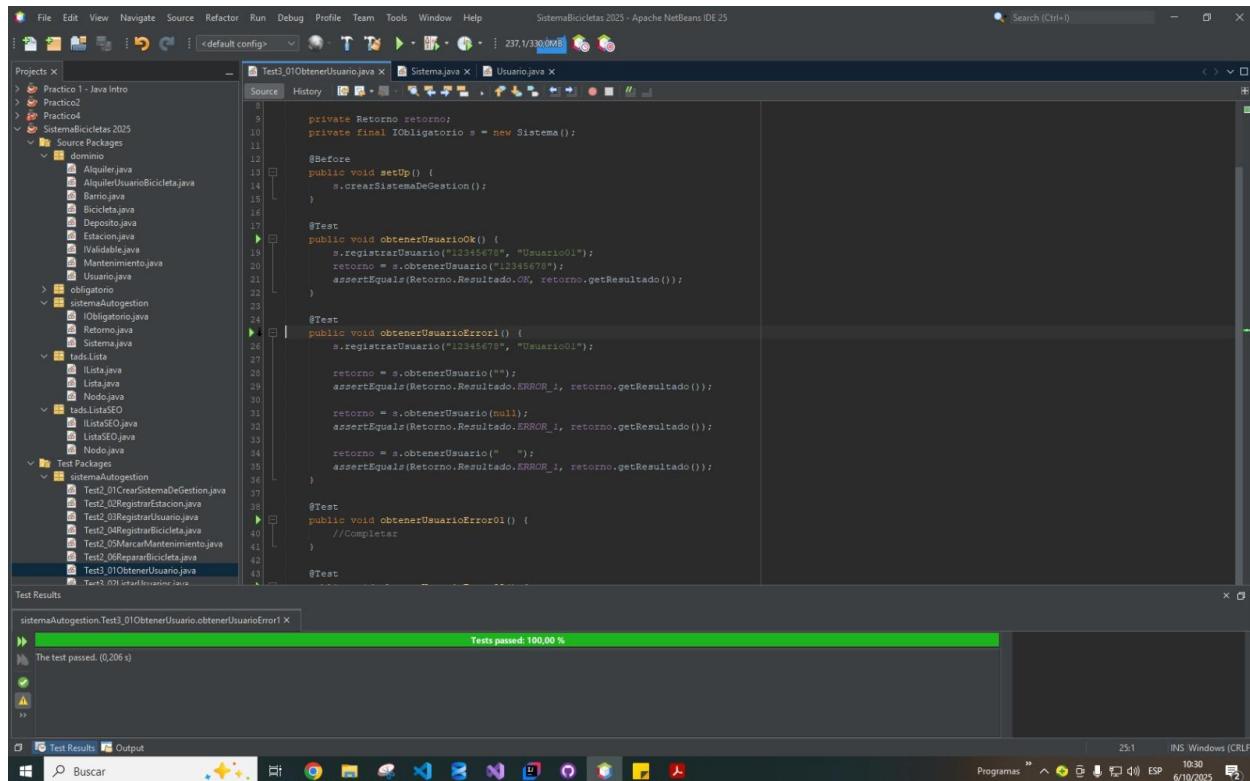
sistemasAutogestion.Test2\_01ClearSistemaDeGestion X sistemasAutogestion.Test2\_05MarcarMantenimiento.MarcarMantenimiento03 X sistemasAutogestion.Test2\_05MarcarMantenimiento.MarcarMantenimiento04 X sistemasAutogestion.Test2\_06RepararBicicleta.repararBicicletaOk X sistemasAutogestion.Test2\_06RepararBicicleta.repararBicicleta01 X

Tests passed: 100.00 %

The test passed. (0.189 s)

10:24 INI Windows (CRLF)

### 3.1: Error 1



```
private Retorno retorno;
private final IObligatorio s = new Sistema();

@Before
public void setUp() {
    s.crearSistemaDeGestion();
}

@Test
public void obtenerUsuarioOk() {
    s.registrarUsuario("12345678", "Usuario01");
    retorno = s.obtenerUsuario("12345678");
    assertEquals(Retorno.Resultado.OK, retorno.getResultado());
}

@Test
public void obtenerUsuarioError() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario(null);
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario("  ");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
}

@Test
public void obtenerUsuarioError01() {
    //Completar
}

@Test
public void obtenerUsuarioError02() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("123456");
    assertEquals(Retorno.Resultado.ERROR_2, retorno.getResultado());
}

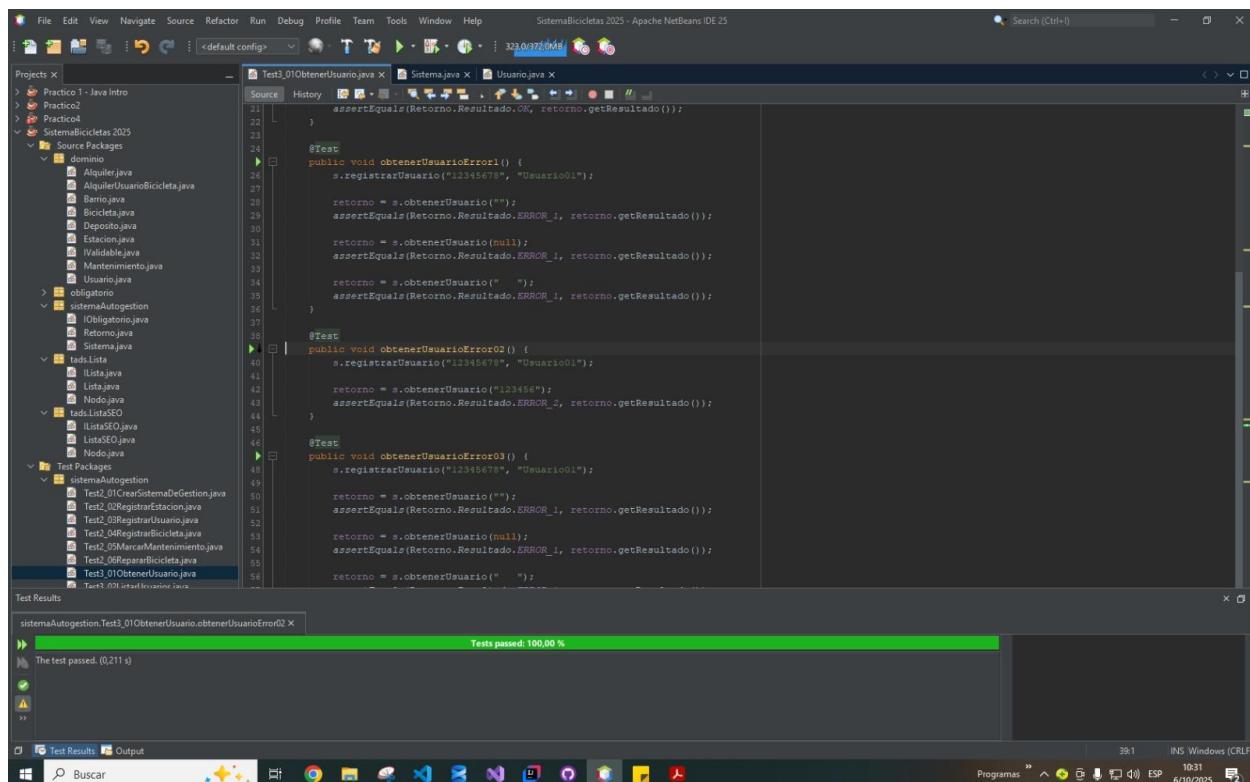
@Test
public void obtenerUsuarioError03() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario(null);
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario("  ");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
}
```

### 3.1: Error 2



```
assertEquals(Retorno.Resultado.OK, retorno.getResultado());

@Test
public void obtenerUsuarioError01() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario(null);
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario("  ");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
}

@Test
public void obtenerUsuarioError02() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("123456");
    assertEquals(Retorno.Resultado.ERROR_2, retorno.getResultado());
}

@Test
public void obtenerUsuarioError03() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario(null);
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());

    retorno = s.obtenerUsuario("  ");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
}
```

### 3.1: Error 3

The screenshot shows the Apache NetBeans IDE interface with several tabs open at the top, including Test3\_03ListarUsuarios.java, Test3\_04MapaDeEstaciones.java, Test2\_06RepararBicicleta.java, Test2\_07RegistrarEstacion.java, Test2\_08RegistrarBicicleta.java, Test2\_09MarcaMantenimiento.java, Test3\_01ObtenerUsuario.java, and Test3\_02ListarBicicletasEnDeposito.java. The main editor window displays Java test code for error handling:

```
public void obtenerUsuarioError1() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    assertEquals("", retorno.getValorString());

    retorno = s.obtenerUsuario(null);
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    assertEquals("", retorno.getValorString());

    retorno = s.obtenerUsuario("   ");
    assertEquals(Retorno.Resultado.ERROR_1, retorno.getResultado());
    assertEquals("", retorno.getValorString());
}

@Test
public void obtenerUsuarioError02() {
    s.registrarUsuario("12345678", "Usuario01");

    retorno = s.obtenerUsuario("123456");
    assertEquals(Retorno.Resultado.ERROR_3, retorno.getResultado());
}

@Test
public void obtenerUsuarioError03() {
    retorno = s.obtenerUsuario("12345678");
    assertEquals(Retorno.Resultado.ERROR_3, retorno.getResultado());
}
```

The Test Results panel at the bottom shows a green bar indicating "Tests passed: 100.00 %".

### 3.2: Ok – Listar Usuario Vacío

The screenshot shows the Apache NetBeans IDE interface with several tabs open at the top, including Test3\_01ObtenerUsuario.java, Sistema.java, Usuario.java, and Test3\_02ListarUsuarios.java. The main editor window displays Java test code for listing an empty user:

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

public class Test3_02ListarUsuarios {

    private Retorno retorno;
    private final IObligatorio s = new Sistema();

    @Before
    public void setUp() {
        s.crearSistemaDeGestion();
    }

    @Test
    public void listarUsuariosVacio() {
        retorno = s.listarUsuarios();
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
        assertEquals("", retorno.getValorString());
    }

    @Test
    public void listarUsuariosSoloUnoUsuario() {
        s.registraUsuario("12345678", "Usuario01");
        retorno = s.listarUsuarios();
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
        assertEquals("Usuario01|12345678", retorno.getValorString());
    }

    @Test
    public void listarUsuariosIngresosOrdenados() {
        s.registraUsuario("11111111", "Usuario01");
        s.registraUsuario("31221111", "Usuario02");
        s.registraUsuario("11331111", "Usuario03");
        retorno = s.listarUsuarios();
        assertEquals(Retorno.Resultado.OK, retorno.getResultado());
        assertEquals("Usuario01|11111111|Usuario02|31221111|Usuario03|11331111", retorno.getValorString());
    }
}
```

The Test Results panel at the bottom shows a green bar indicating "Tests passed: 100.00 %".

### 3.2: Ok – 2 – Listar Usuario Solo Un Usuario

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects:** Shows several Java projects including "Practico 1 - Java Intro", "Practico2", "Practico4", and "SistemaBicletas 2025".
- Source Packages:** Under "SistemaBicletas 2025", there are several packages: "domingo", "obligatorio", "sistemaAutogestion", and "Test Packages".
- Code Editor:** The main editor window displays Java test code for "Test3\_02ListarUsuarios.java". The code includes annotations like @Test and @Before, and methods for testing user listing.
- Test Results:** A results panel at the bottom shows a green bar indicating "Tests passed: 100.00 %". It also displays the message "The test passed. (0,207 s)".
- System Bar:** The bottom bar shows system icons and the date/time: "11/10/2025 11:10".

### 3.2: Ok – 3 – Listar Usuario Ingresado Ordenado

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects:** Shows several Java projects including "Practico 1 - Java Intro", "Practico2", "Practico4", and "SistemaBicletas 2025".
- Source Packages:** Under "SistemaBicletas 2025", there are several packages: "domingo", "obligatorio", "sistemaAutogestion", and "Test Packages".
- Code Editor:** The main editor window displays Java test code for "Test3\_02ListarUsuarios.java". The code includes annotations like @Test and @Before, and methods for testing user listing in an ordered manner.
- Test Results:** A results panel at the bottom shows a green bar indicating "Tests passed: 100.00 %". It also displays the message "The test passed. (0,192 s)".
- System Bar:** The bottom bar shows system icons and the date/time: "11/10/2025 11:10".

### 3.3: Ok

The screenshot shows the Apache NetBeans IDE interface with the following details:

- File Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** SistemaBicicletas 2025 - Apache NetBeans IDE 2025
- Project Explorer (Projects X):**
  - Practico 1 - Java Intro
  - Practico2
  - Practico4
  - SistemaBicicletas 2025
    - Source Packages
      - dominio
        - Aquila.java
        - AquilaUsuarioBicicleta.java
        - Banco.java
        - Bicicleta.java
        - Deposito.java
        - Estacion.java
        - Invalidable.java
        - Mantenimiento.java
        - Usuario.java
      - obligatorio
        - IObligatorio.java
        - Retorno.java
        - Sistema.java
      - tads.Lista
        - Lista.java
        - Lista.java
        - Nodo.java
      - tads.ListaSEO
        - ListaSEODjave.java
        - ListaSEODjave.java
        - Nodo.java
    - Test Packages
      - sistemaAutogestion
        - Test2\_0CrearSistemaDeGestion.java
        - Test2\_0ListarBicicletasEnDeposito.java
        - Test2\_0RegistrarEstacion.java
        - Test2\_0RegistrarUsuario.java
        - Test2\_0ReestarBicicleta.java
        - Test2\_0MarcaMantenimiento.java
        - Test2\_0ReportarBicicleta.java
        - Test3\_0ListarUsuario.java
- Code Editor:** The main editor window displays Java test code for the `sistemaAutogestion` package. It includes tests for listing empty deposits, listing a single bike in a deposit, and listing bikes ordered by availability.
- Test Results:** A separate window titled "Test Results" shows the outcome of the tests:
  - Test cases: sistemaAutogestion.Test2\_0ListarBicicletasEnDeposito.listarBicicletasEnDepositoVacio, sistemaAutogestion.Test2\_0ListarBicicletasEnDeposito.listarBicicletaEnDepositoUnaBici, sistemaAutogestion.Test3\_0ListarBicicletasEnDeposito.listarBicicletasEnDepositoOrdenado.
  - Status: Tests passed: 100.00 %
  - Message: The test passed. (0.207s)
- Bottom Bar:** Shows the Windows taskbar with various application icons and the system clock indicating 11:47 AM on 6/10/2025.

### 3.4.1:

The screenshot shows the Apache NetBeans IDE 25 interface with the following details:

- File Bar:** File, Edit, View, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Title Bar:** SistemaBicicletas 2025 - Apache NetBeans IDE 25
- Toolbar:** Includes icons for file operations like New, Open, Save, Cut, Copy, Paste, Find, and Run.
- Project Explorer (Projects):**
  - Practico 1 - Java Intro
  - Practico2
  - Practico4
  - SistemaBicicletas 2025
    - Source Packages
      - domingo
        - Alquiler.java
        - AlquilerUsuarioBicicleta.java
        - Bario.java
        - Bicicleta.java
        - Bicicleta.java
        - Estacion.java
        - Validable.java
        - Mantenimiento.java
        - Usuario.java
      - obligatorio
        - IObligatorio.java
        - Retorno.java
        - Sistema.java
      - tads.Lista
        - ILista.java
        - Lista.java
        - Lista.java
        - Nodo.java
      - tads.ListaSEO
        - IListaSEOD.java
        - ListaSEOD.java
        - Nodo.java
    - Test Packages
      - sistemaAutogestion
        - Test2\_0ClearSistemaDeGestion.java
        - Test2\_0RegistrarEstacion.java
        - Test2\_0RegistrarUsuario.java
        - Test2\_0RegistrarBicicleta.java
        - Test2\_0Mantenimiento.java
        - Test3\_0ObtenerUsuario.java- Code Editor:** The main editor window displays the Java code for `Test3_04MapaDeEstaciones.java`. The code is a JUnit test for the `informacionMapa` method, comparing expected output strings against actual results from the `Sistema` class.
- Test Results:** A separate window titled "Test Results" shows the execution status of the tests. It indicates 100.00% tests passed in 0.188 seconds.
- Bottom Bar:** Includes icons for Test Results, Output, Buscar (Search), and various system status indicators.
- System Status:** Shows the date and time as 12:01 on 6/10/2025.

### 3.4.2

The screenshot shows the Apache NetBeans IDE 25 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and SistemaBicicletas 2025 - Apache NetBeans IDE 25. The left sidebar displays the Projects X tree, showing Práctico 1 - Java Intro, Práctico2, Práctico4, and SistemaBicicletas 2025. The SistemaBicicletas 2025 node has several source packages expanded, including dominio, obligatorio, sistemasAutogestion, and tads. The code editor window contains the Test3\_04MapaDeEstaciones.java file, which contains Java test code for the SistemaBicicletas system. The bottom status bar shows the date and time as 6/10/2025 12:01.

```
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

```
    retorno = s.informacionNapa(mapa);
    assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    assertEquals("3#columna/existe", retorno.getValorString());
}

@Test
public void informacionNapa_Ejemplo2() {
    String[][] mapa = new String[][]{
        {"0", "0", "E1", "E2"}, 
        {"0", "0", "E3", "0"}, 
        {"0", "E4", "0", "0"}, 
        {"0", "0", "0", "0"}
    };
}

retorno = s.informacionNapa(mapa);
assertEquals(Retorno.Resultado.OK, retorno.getResultado());
assertEquals("2#ambas|existe", retorno.getValorString());
}

@Test
public void informacionNapa_Ejemplo3() {
    String[][] mapa = new String[][]{
        {"E1", "0", "0", "E2"}, 
        {"E3", "0", "0", "0"}, 
        {"0", "E4", "0", "E5"}, 
        {"0", "E6", "0", "0"}
    };
}

retorno = s.informacionNapa(mapa);
assertEquals(Retorno.Resultado.OK, retorno.getResultado());
assertEquals("2#ambas|no existe", retorno.getValorString());
}

@Test
public void informacionNapa_Vacio() {
    String[][] mapa = new String[][]{
        {"0", "0", "0"}, 
        {"0", "0", "0"}, 
        {"0", "0", "0"}
    };
}
```

Test Results

sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Ejemplo1 X sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Ejemplo2 X sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Ejemplo3 X sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Vacio X

Tests passed: 100.00 %

The test passed. (0,207 s)

INN Windows (CRLF)

### 3.4.3

The screenshot shows the Apache NetBeans IDE 25 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and SistemaBicicletas 2025 - Apache NetBeans IDE 25. The left sidebar displays the Projects X tree, showing Práctico 1 - Java Intro, Práctico2, Práctico4, and SistemaBicicletas 2025. The SistemaBicicletas 2025 node has several source packages expanded, including dominio, obligatorio, sistemasAutogestion, and tads. The code editor window contains the Test3\_04MapaDeEstaciones.java file, which contains Java test code for the SistemaBicicletas system. The bottom status bar shows the date and time as 6/10/2025 12:01.

```
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

```
    retorno = s.informacionNapa(mapa);
    assertEquals(Retorno.Resultado.OK, retorno.getResultado());
    assertEquals("3#columna/existe", retorno.getValorString());
}

@Test
public void informacionNapa_Ejemplo2() {
    String[][] mapa = new String[][]{
        {"0", "0", "E1", "E2"}, 
        {"0", "0", "E3", "0"}, 
        {"0", "E4", "0", "0"}, 
        {"0", "0", "0", "0"}
    };
}

retorno = s.informacionNapa(mapa);
assertEquals(Retorno.Resultado.OK, retorno.getResultado());
assertEquals("2#ambas|existe", retorno.getValorString());
}

@Test
public void informacionNapa_Ejemplo3() {
    String[][] mapa = new String[][]{
        {"E1", "0", "0", "E2"}, 
        {"E3", "0", "0", "0"}, 
        {"0", "E4", "0", "E5"}, 
        {"0", "E6", "0", "0"}
    };
}

retorno = s.informacionNapa(mapa);
assertEquals(Retorno.Resultado.OK, retorno.getResultado());
assertEquals("2#ambas|no existe", retorno.getValorString());
}

@Test
public void informacionNapa_Vacio() {
    String[][] mapa = new String[][]{
        {"0", "0", "0"}, 
        {"0", "0", "0"}, 
        {"0", "0", "0"}
    };
}
```

Test Results

sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Ejemplo1 X sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Ejemplo2 X sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Ejemplo3 X sistemaAutogestion.Test3\_04MapaDeEstaciones.informacionNapa\_Vacio X

Tests passed: 100.00 %

The test passed. (0,194 s)

INN Windows (CRLF)

### 3.4.4:

The screenshot shows the Apache NetBeans IDE 25 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a status bar indicating "SistemaBicicletas 2025 - Apache NetBeans IDE 25" and "220.3/834.0MB".

The left sidebar displays the "Projects X" tree, showing several Java packages under "Práctico 1 - Java Intro", "Práctico2", "Práctico4", and "SistemaBicicletas 2025". The "SistemaBicicletas 2025" package contains sub-packages like "Alquiler", "Banco", "Bicicleta", "Depósito", "Estación", "Validable", "Mantenimiento", "Usuario", "obligatorio", "sistemaAutogestion", and "tads.Lista".

The main editor area shows Java code for a test class, specifically for testing the "informacionMapa" method. The code uses assertions to check if the method returns the expected results for different map configurations.

```
42     assertEquals("Retorno.Resultado.OK", retorno.getResultado());
43     assertEquals("2#ambas/existe", retorno.getValorString());
44 }
45
46 @Test
47 public void informacionMapa_Ejemplo3() {
48     String[][] mapa = new String[]{}{
49         {"*", "*", "E", "O", "E22"}, // Row 1
50         {"E23", "E", "O", "O", "O"}, // Row 2
51         {"O", "E", "O", "O", "EST"}, // Row 3
52         {"O", "E", "O", "O", "OM"} // Row 4
53     };
54
55     retorno = s.informacionMapa(mapa);
56     assertEquals("Retorno.Resultado.OK", retorno.getResultado());
57     assertEquals("2#ambas/no existe", retorno.getValorString());
58 }
59
60 @Test
61 public void informacionMapa_Vacio() {
62     String[][] mapa = new String[]{}{
63         {"O", "O", "O"}, // Row 1
64         {"O", "O", "O"}, // Row 2
65         {"O", "O", "O"} // Row 3
66     };
67
68     retorno = s.informacionMapa(mapa);
69     assertEquals("Retorno.Resultado.OK", retorno.getResultado());
70     assertEquals("0#ambas/no existe", retorno.getValorString());
71 }
72 }
```

The bottom section shows the "Test Results" tab with a green bar indicating "Tests passed: 100.00 %". It also displays the message "The test passed. (0.231 s)".

The taskbar at the bottom includes icons for Test Results, Output, and Buscar, along with the Windows Start button and other system icons.