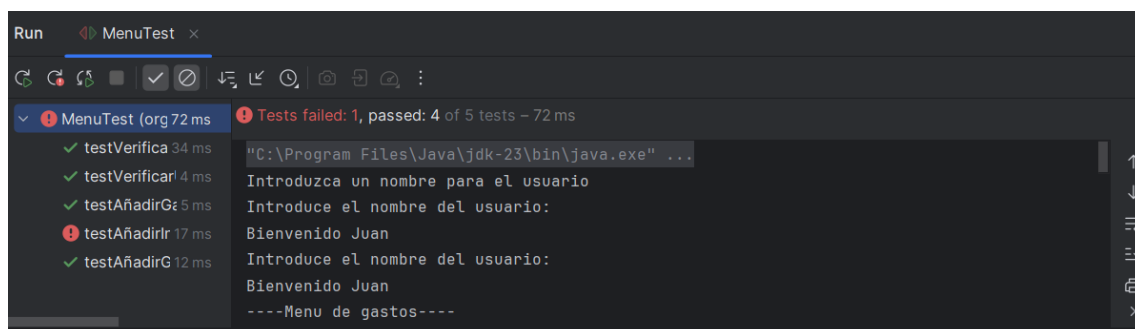


Testing del trabajo de Marco

Realizamos el testing con el id IntelliJ, una vez el compañero nos comparte el proyecto lo abrimos y realizamos el test de fallos y creamos los métodos.

```
11 public class MenuTest {
16     public void setUp() {
17         menu = new Menu();
18     }
19
20     @Test
21     public void testVerificarUsuarioSinNombre() {
22         assertFalse(menu.verificarUsuario());
23     }
24
25     @Test
26     public void testVerificarUsuarioConNombre() {
27         setInput("Juan\n");
28         menu.añadirUsuario();
29         assertTrue(menu.verificarUsuario());
30     }
31
32     @Test
33     public void testAñadirIngreso() {
34         setInput("Juan\n100.5\n");
35         menu.añadirUsuario();
36         menu.añadirIngreso();
37         assertEquals("expected: 100.5, menu.saldo, delta: 0.001");
38     }
39
40     @Test
41     public void testAñadirGastoConSaldoSuficiente() {
```

Una vez realizado ejecutamos el test.



```
Run MenuTest x
Tests failed: 1, passed: 4 of 5 tests - 72 ms
testVerifica 34 ms
testVerificar 4 ms
testAñadirG 5 ms
testAñadirIr 17 ms
testAñadirG 12 ms
"C:\Program Files\Java\jdk-23\bin\java.exe" ...
Introduzca un nombre para el usuario
Introduce el nombre del usuario:
Bienvenido Juan
Introduce el nombre del usuario:
Bienvenido Juan
----Menu de gastos----
```

Encontramos un fallo en el método del test de añadir ingresos. Vemos el fallo.

```
! Tests failed: 1, passed: 4 of 5 tests – 72 ms

Introduce el nombre del usuario:
Bienvenido Juan
Introduzca la cantidad que desea ingresar:

java.util.InputMismatchException Create breakpoint
    at java.base/java.util.Scanner.throwFor(Scanner.java:964)
    at java.base/java.util.Scanner.next(Scanner.java:1619)
    at java.base/java.util.Scanner.nextDouble(Scanner.java:2590)
    at org.example.Menu.añadirIngreso(Menu.java:75)
> at org.example.MenuTest.testAñadirIngreso(MenuTest.java:36) <1 internal line
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
```

Causa del fallo

Este error ocurre cuando `Scanner.nextDouble()` intenta leer un valor numérico en formato decimal, pero el dato que recibe no es válido para convertirlo a `double`.

Para evitar este fallo, se debe añadir una línea para **limpiar el búfer** después de leer un número decimal, justo después de `nextDouble()`:

```
cantidad = leer.nextDouble();
```

```
leer.nextLine(); // ← Limpia el salto de línea que queda pendiente
```

Con esto se garantiza que el `Scanner` no se vea afectado por entradas anteriores y se pueda continuar leyendo datos correctamente.

MEJORAS

. Evitar múltiples instancias de Scanner

En el método `menuPrincipal()` se vuelve a crear un `Scanner` cuando ya existe uno como atributo. Esto es redundante y puede causar problemas con la lectura de la entrada.

. Usar `nextLine()` con cuidado al mezclar con `nextInt()` y `nextDouble()`

Después de leer un número con `nextInt()` o `nextDouble()`, queda un salto de línea en el búfer que debe ser consumido con `nextLine()` para evitar errores posteriores.

Mejora: Añadir `leer.nextLine()`; después de cada lectura numérica.

. Agregar comentarios y documentación

No hay ningún comentario explicando la intención del código. Esto lo hace menos mantenible para otros desarrolladores.

Mejora: Agrega comentarios breves en métodos clave y bloques de lógica compleja.