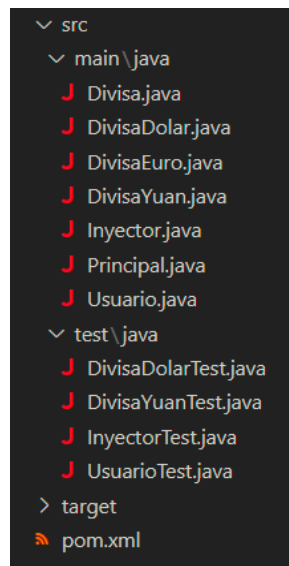


## Programa Junit

El programa ejemplifica el uso de Junit5, en este caso se usa de base el proyecto anterior (Inyección de Dependencias). A continuación se muestran las nuevas clases que se realizaron para hacer las pruebas. Estas pruebas ayudan a garantizar que las clases del proyecto funcionen según lo esperado y permiten la detección temprana de posibles problemas cuando se realizan cambios en el código.



Estructura del proyecto

### UsuarioTest.java

```
src > test > java > J UsuarioTest.java > ...
1  import static org.junit.jupiter.api.Assertions.*;
2
3  import org.junit.jupiter.api.Test;
4
5  class UsuarioTest {
6
7      @Test
8      void testRealizaConversion() {
9          Usuario usuario = new Usuario(tipo:'D', cantidad:10.5);
10         usuario.setmyDivisa(new DivisaDolar());
11
12         // Asegurarse de que no se produzca una excepción
13         assertDoesNotThrow(() -> usuario.realizaConversion());
14     }
15 }
16
```

En este archivo de prueba, estamos utilizando JUnit 5. La clase UsuarioTest contiene un método de prueba llamado testRealizaConversion(). Este método crea una instancia de Usuario con un tipo 'D' y una cantidad de 10.5, luego le inyecta una instancia de DivisaDolar y finalmente invoca el método realizaConversion(). La afirmación assertDoesNotThrow asegura que no se lance ninguna excepción durante la conversión.

#### InjectorTest.java

```
src > test > java > J InjectorTest.java > ...
1  import static org.junit.jupiter.api.Assertions.*;
2
3  import org.junit.jupiter.api.Test;
4
5  class InjectorTest {
6
7      @Test
8      void testInyectarConversion() {
9          Usuario usuario = new Usuario(tipo:'E', cantidad:10);
10         Injector.inyectarConversion(usuario);
11
12         assertEquals(expected:DivisaEuro.class, usuario.getmyDivisa().getClass(),
13             message:"La divisa debería ser de tipo DivisaEuro");
14     }
15 }
```

En este archivo de prueba, estamos probando el comportamiento del Injector. Creamos un Usuario con tipo 'E' y cantidad 10, luego utilizamos el Injector para inyectar la Divisa correspondiente. La afirmación assertEquals verifica que la divisa asignada al usuario es de tipo DivisaEuro.

#### DivisaDolarTest.java

```
src > test > java > J DivisaDolarTest.java > ...
1  import static org.junit.jupiter.api.Assertions.*;
2  import org.junit.jupiter.api.Test;
3
4  class DivisaDolarTest {
5
6      @Test
7      void testConversion() {
8          DivisaDolar divisaDolar = new DivisaDolar();
9          double cantidad = 5.0;
10
11          // Asegurarse de que la conversión es correcta
12          assertEquals(5.0 * divisaDolar.tipoDeCambio,
13                      divisaDolar.conversion(cantidad));
14      }
15  }
```

En este archivo de prueba, estamos verificando el método `conversion()` de la clase `DivisaDolar`. Creamos una instancia de `DivisaDolar`, establecemos una cantidad y utilizamos `assertEquals` para asegurarnos de que la conversión sea correcta según el tipo de cambio definido en `DivisaDolar`.