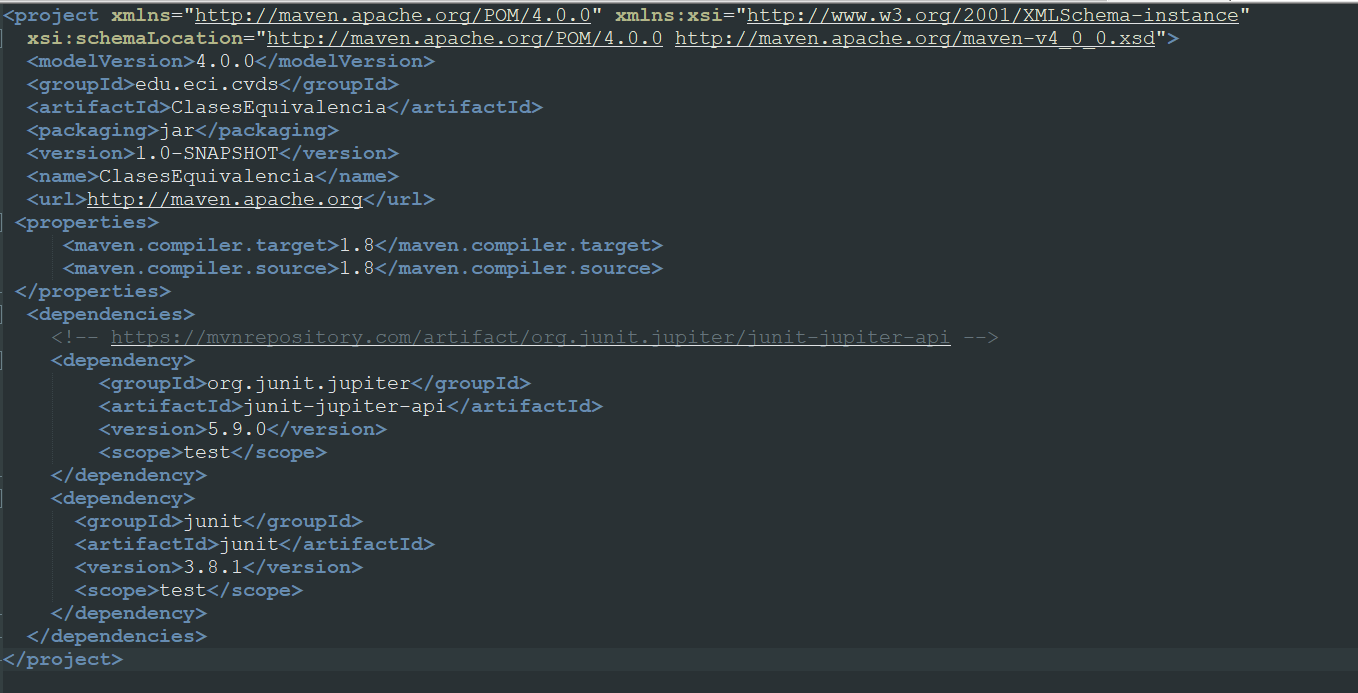
**CREAR UN PROYECTO CON MAVEN**

Comando usado: mvn archetype:generate -DgroupId=edu.eci.cvds -DartifactId=ClasesEquivalencia -DarchetypeArtifactId=maven-archetype-quickstart

pom.xml



Resultado compilación:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Resultado de la compilación de test.

Comando usado: mvn compiler:testCompile

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

EJECUTAR LAS PRUEBAS

Texto

Descripción generada automáticamente

De AppTest se ejecutó una prueba. De ResgistryTest se corrió el archivo pero no ejecutó ninguna prueba.

La diferencia entre mvn package y mvn test es que mvn test compila y ejecuta solamente lo relacionado a los tests, mientras que mvn package compila y ejecuta los tests y el resto del codigo

Ejecución de los tests

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**EJERCICIO “DESCUENTOS DE TARIFAS”**

De acuerdo con lo indicado, y teniendo en cuenta que NO hay precondiciones, ¿en qué casos se debería arrojar una excepción de tipo ExcepcionParametrosInvalidos?

* Cuando la tarifa base es menor a cero
* Cuando la edad es negativa
* Cuando los días de antelación son negativos

Para los casos 3 en adelante, se considerará una tarifaBase = 100000

Condiciones límite:

* Edad ∈(17,18, 65, 66, -1)
* diasAntelación ∈(20, 21, -1)
* tarifaBase ∈ (0, 1)

Clase de equivalencia:

* Edad < 0; 0 < Edad < 18; 18 < Edad < 65; Edad > 65
* diasAntelacion < 0; 0 < diasAntelacion < 20; díasAntelacion > 20
* tarifaBase <= 0; tarifaBase > 0

|  |  |  |
| --- | --- | --- |
| Número | Clase de equivalencia (caso de prueba) | Resultado |
| 1 | tarifaBase = -1 | Error |
| 2 | edad = -1 | Error |
| 3 | diasAntelacion = -1 | Error |
| 4 | diasAntelacion = 21; edad= 18; tarifaBase >= 0 | 85000 |
| 5 | diasAntelacion = 21; edad= 17; tarifaBase >= 0 | 80000 |
| 6 | diasAntelacion = 21; edad= 66; tarifaBase >= 0 | 77000 |
| 7 | diasAntelacion = 20; edad= 18; tarifaBase >= 0 | 100000 |
| 8 | diasAntelacion = 20; edad= 17; tarifaBase >= 0 | 95000 |
| 9 | diasAntelacion = 20; edad= 66; tarifaBase >= 0 | 92000 |

Los resultados de las pruebas son:

Texto

Descripción generada automáticamente

La explicación de los tests que fallaron es la siguiente:

* Las pruebas que tienen la palabra ‘exception’ fallaron a causa de que el método no genera excepciones, y no se puede hacer que estas generen errores ya que el comando mvn install lo que hace es ‘descomprimir’ el archivo .jar en archivos .class, no .java; por ende, no es posible editar el código para que genere las excepciones. Sabemos que no es muy ingenieril de nuestra parte hacer los tests de las excepciones de esa forma, pero al no poder editar el código de CalculadorDescuentos no podemos hacer que esta genere excepciones para poder usar assertThrows. Igual dejamos comentareado como sería la prueba si las funciones arrojaran excepciones.
* Las otras pruebas que fallaron, nos dimos de cuenta que es porque, cuando no aplica el descuento por antelación, tampoco considera el descuento por edad.