LABORATORIO NO.3

ELABORADO POR:

MIGUEL ÁNGEL SALAMANCA

JUAN CAMILO BAZURTO

PRESENTADO A:

PROF. IVAN VIASUS QUINTERO

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

CICLOS DE VIDA DE SOFTWARE

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ D. C.

2022-2

**CREAR UN PROYECTO CON MAVEN**

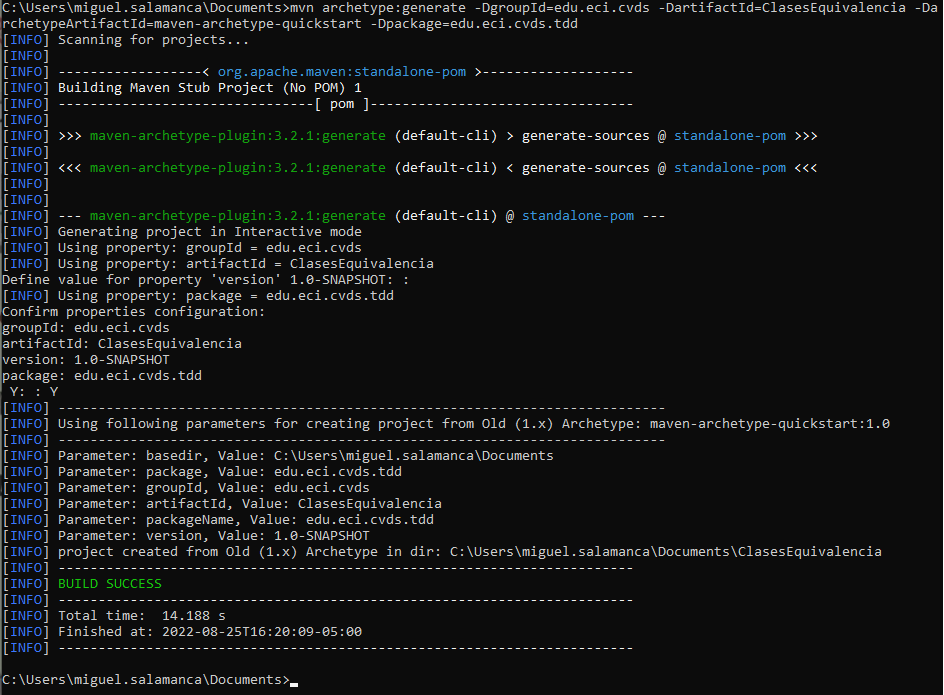
En el directorio de trabajo ejecutar el comando necesario para crear un proyecto maven, basado en un arquetipo:

Grupo: edu.eci.cvds

Artefacto: ClasesEquivalencia

Paquete: edu.eci.cvds.tdd

archetypeArtifactId: maven-archetype-quickstart



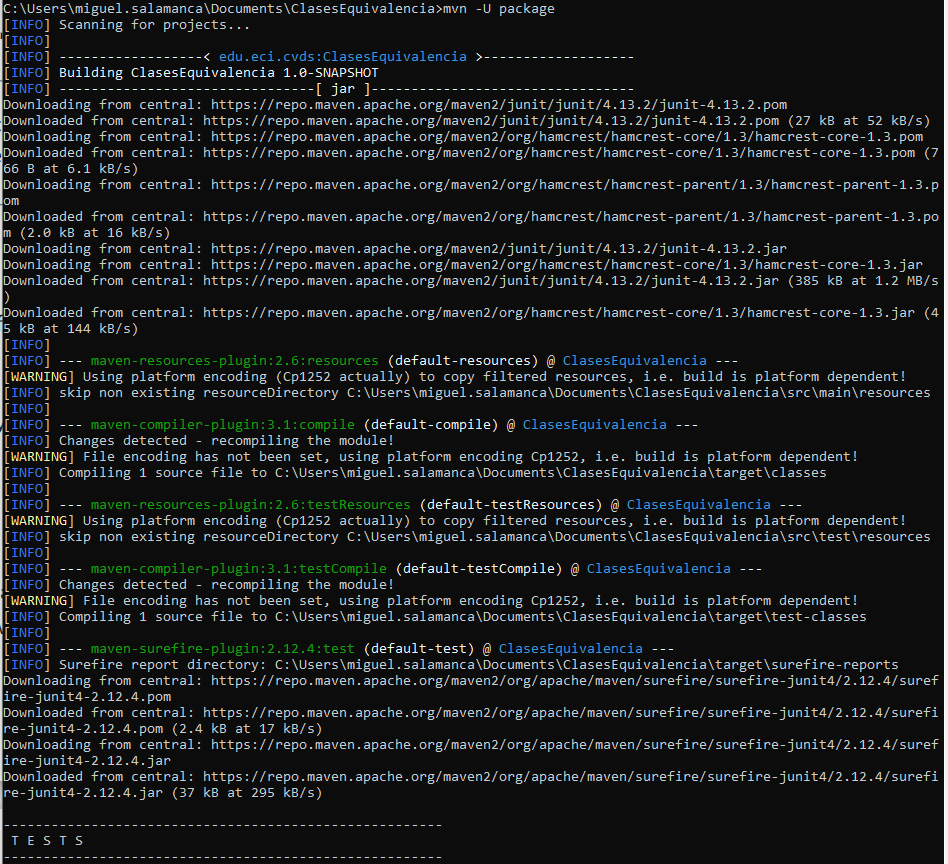
**Actualizar y crear dependencias en el proyecto**

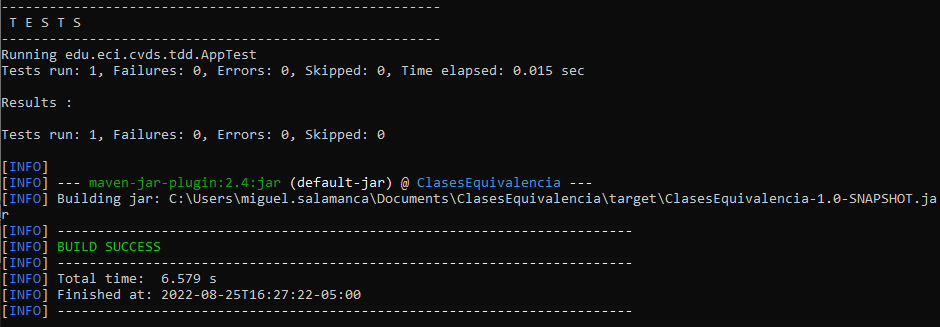
* Busque en internet el repositorio central de maven.
* Busque el artefacto JUnit y entre a la versión más nueva
* Ingrese a la pestaña de Maven y haga click en el texto de la dependencia para copiarlo al portapapeles.
* Edite el archivo pom.xml y realice las siguientes actualizaciones:
  + Agregue la dependencia copiada a la sección de dependencias.
  + Hay que cambiar la versión del compilador de Java a la versión 8, para ello, agregue la sección properties antes de la sección de dependencias:



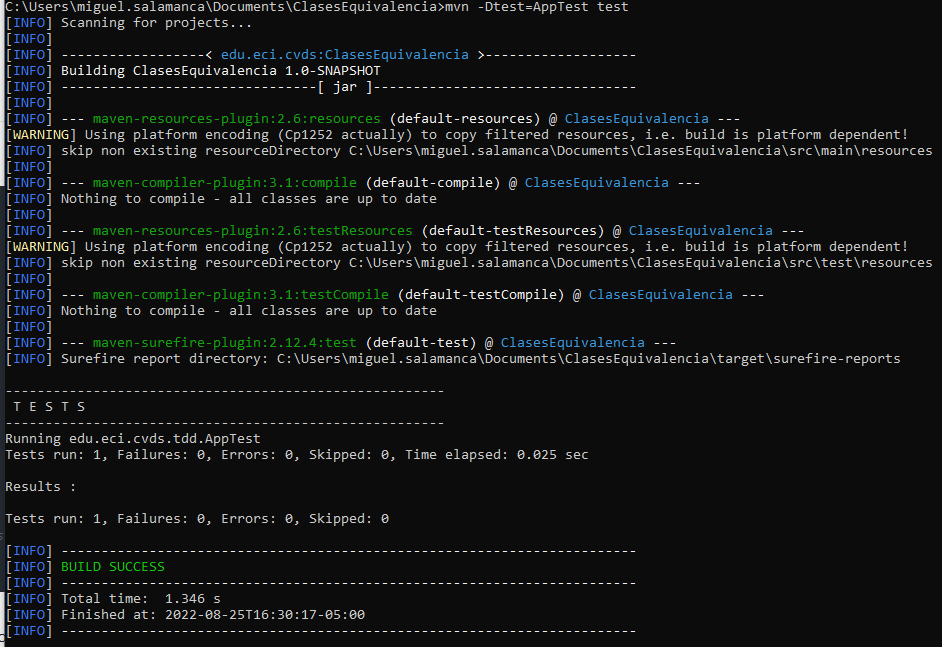
**COMPILAR Y EJECUTAR**

* Ejecute los comandos necesarios de Maven, para compilar el proyecto y verificar que el proyecto se creó correctamente y los cambios realizados al archivo pom no generan inconvenientes.





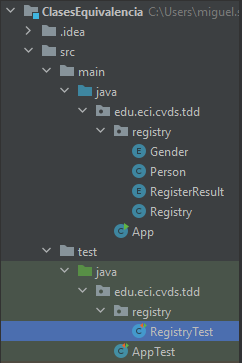
* Busque el comando requerido para ejecutar las pruebas unitarias de un proyecto desde Maven y ejecútelo sobre el proyecto. Se debe ejecutar la clase AppTest con resultado exitoso.



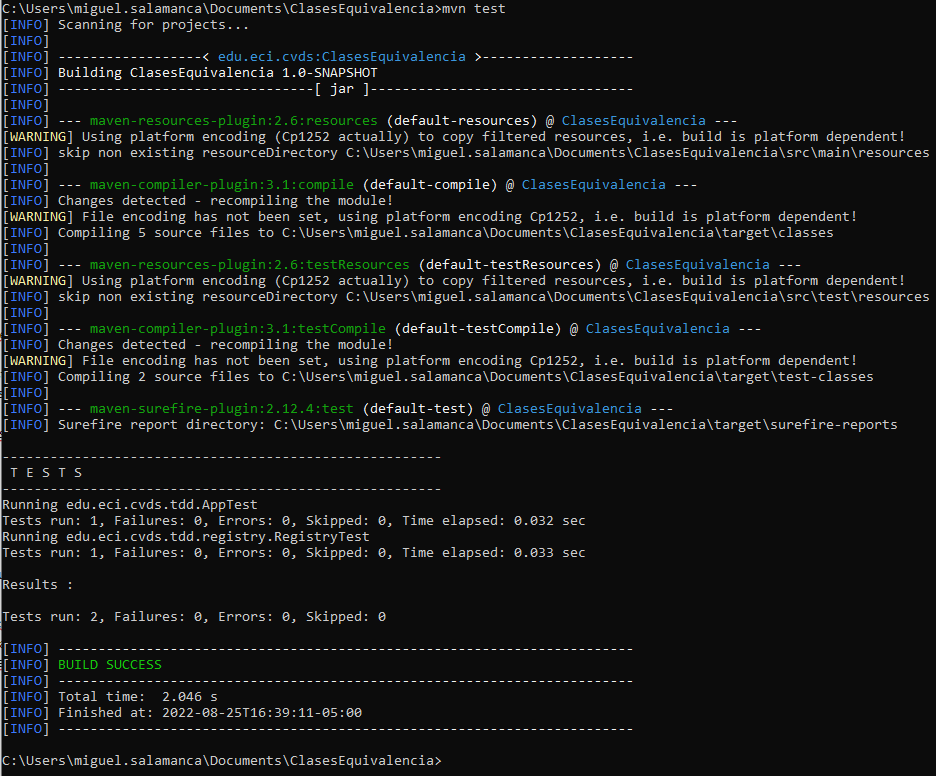
**EJERCICIO “REGISTRADURÍA”**

**HACER EL ESQUELETO DE LA APLICACION**

* Cree el archivo RegisterResult.java en el directorio edu.eci.cvds.tdd.registry con la enumeración:
* Cree el archivo Gender.java en el paquete edu.eci.cvds.tdd.registry con la enumeración:
* Cree el archivo Person.java en el paquete edu.eci.cvds.tdd.registry con el siguiente contenido:
* Cree el archivo Registry.java en el directorio edu.eci.cvds.tdd.registry con el método registerVoter:
* Cree la misma estructura de paquetes edu.eci.cvds.tdd.registry en la ruta src/test/java. Todos los archivos relacionados específicamente con los temas de pruebas, siempre deben ir bajo la carpeta test.
* Bajo la carpeta de pruebas, cree la clase RegistryTest.java en el directorio edu.eci.cvds.tdd.registry de la siguiente manera



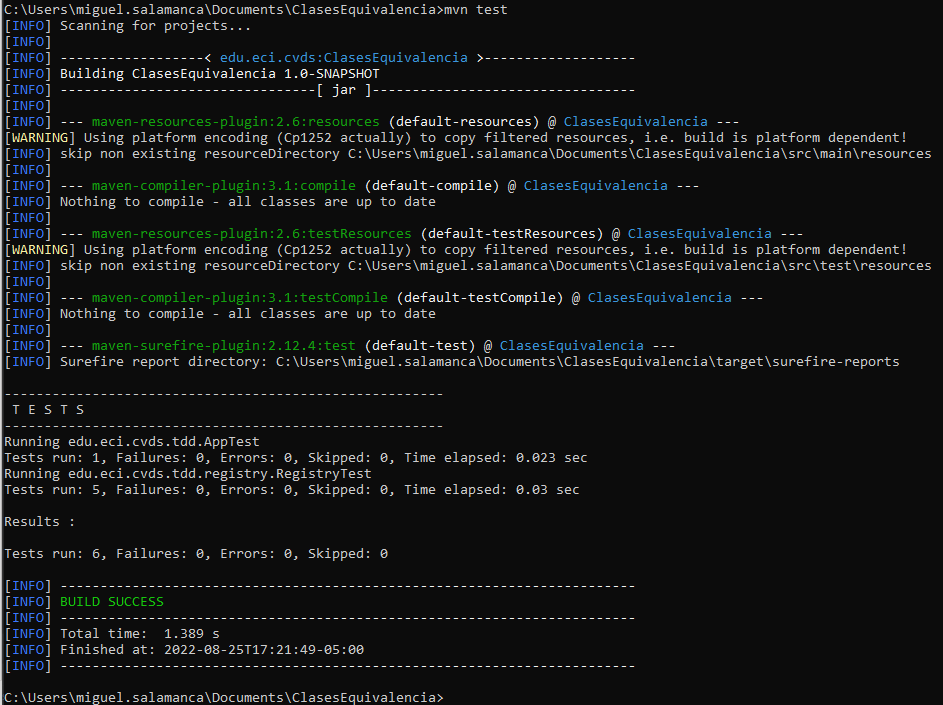
**EJECUTAR LAS PRUEBAS**



**FINALIZAR EL EJERCICIO**

* Piense en los casos de equivalencia que se pueden generar del ejercicio para la registraduría, dadas las condiciones. Deben ser al menos 5.
  + Muerto
  + Menor de edad
  + Edad inválida
  + Persona Válida
  + Duplicado
* Complete la implementación de la clase RegistryTest.java con (al menos) un método por cada clase de equivalencia, creando diferentes personas y validando que el resultado sea el esperado.
* 
* Complete la implementación del método registerVoter en la clase Registry.java para retornar el resultado esperado según la entrada.





**EJERCICIO "DESCUENTO DE TARIFAS"**

**REALIZAR DISEÑO DE PRUEBAS**

* Para realizar de forma correcta el diseño de sus pruebas responda las preguntas que se encuentran en el siguiente documento.

1.De acuerdo con lo indicado,y teniendo en cuenta que NO hay precondiciones,en qué casos se debería arrojar una excepción de tipo ExcepcionParametrosInvalidos?. Agregue esto a la especificación.

/\*\*calcular la tarifa de cada billete según el trayecto, la antelación en la que se obtiene el billete y la edad del pasajero, de acuerdo con la normativa 005.@param tarifaBase valor base del vuelo@param diasAntelacion dias de antelación del vuelo@param edad -edad del pasajero@throws ExcepcionParametrosInvalidos tarifaBase menor a cero, dias de antelación menor a cero, edad invalida\*\*/public long calculoTarifa(long tarifaBase, int diasAntelacion, int edad)

2.En la siguiente tabla enumere un conjunto de clases de equivalencia que -según usted-creen una buena división del conjunto de datos de entrada de la función anterior:

|  |  |  |
| --- | --- | --- |
| Número | Clave de equivalencia | Resultado Correcto / Incorrecto |
| 1 | TarifaBase > 0 | Correcto |
| 2 | Edad >= 0 | Correcto |
| 3 | diasAntelación > 0 | Correcto |

3. Para cada clase de equivalencia, defina un caso de prueba específico, definiendo: parámetros de entrada y resultados esperados.

TarifaBase = 300000

DiasAntelacion = 12

Edad = 70

AplicarDescuento(tarifaBase, diasAntelacion, edad)

TarifaBase = 276000

4. A partir de las clases de equivalencia identificadas en el punto 2, identifique las condiciones límite o de frontera de las mismas.

Condiciones límite 1: tarifaBase > 0

Condiciones límite 2: 0 < diasAntelacion < 365

Condiciones límite 3: 0 < edad < 18

5. Para cada una de las condiciones de frontera anteriores, defina casos de prueba específicos.

tarifaBase = 1200000

diasAntelacion = 21

edad = 20

AplicarDescuento(tarifaBase, diasAntelacion, edad)

tarifaBase=1020000

tarifaBase = 100000

diasAntelacion = 15

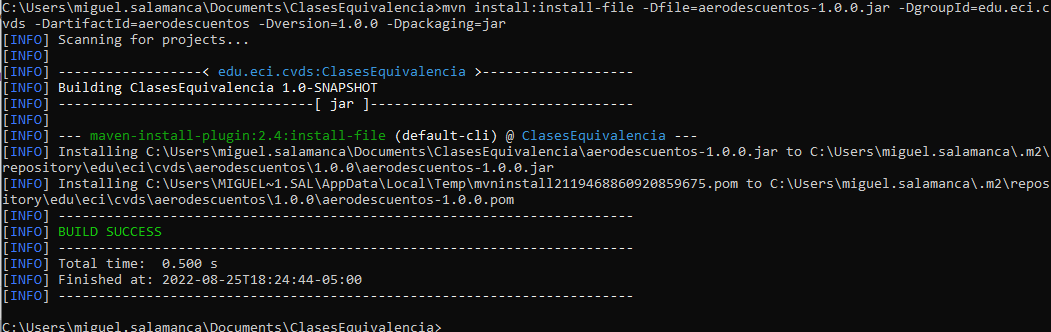
edad = 13

AplicarDescuento(tarifaBase, diasAntelacion, edad)

TarifaBase = 95000

**IMPLEMENTACIÓN DE LAS PRUEBAS**

* Descargue el archivo aerodescuentos.jar y adicione esta nueva dependencia en el archivo pom.xml de su proyecto. Para adicionar una librería personalizada al repositorio local de maven puede ejecutar el siguiente comando.



* Cree el archivo TarifasTest.java en el directorio src/test/java/edu/eci/cvds/tdd/aerodescuentos. Realice la implementación de las pruebas propuestas en la etapa de diseño de pruebas en esta clase. Para sus pruebas debe usar el método "calculoTarifa" de la clase edu.eci.cvds.tdd.aerodescuentos.CalculadorDescuentos, que se encuentran dentro del JAR de la librería personalizada.
* Ejecute el comando de Maven para las fases de compilación y pruebas. Verifique el resultado exitoso de todas las pruebas y el reporte generado.

