

INFORME. TESTING PROBLEMA 2.

TRABAJO TEÓRICO ISO II

PAREJA 1. ANDRÉS DONCEL RODRÍGUEZ DE DIEGO Y ADRIÁN DELGADO MARTÍNEZ

Proyecto: Implementación de Pruebas para la clase Tarifa. **Rama de trabajo:** hotfix.

1. Estrategia y Selección de Casos de Prueba

Para la definición de los casos de prueba se han seguido las estrategias de Caja Negra vistas en teoría, asegurando una cobertura funcional completa del dominio de entrada.

- **Técnicas empleadas:**
 - **Clases de Equivalencia:** para dividir el dominio de entrada en rangos válidos e inválidos (edades, clase, etc.).
 - **Valores Límite:** selección de valores en las fronteras de las clases (ej. edades 17, 18, 25).
 - **Conjetura de Errores:** selección de valores propensos a fallos (ej. ingresos negativos, strings vacíos en clase, nulos).
- **Criterio de Cobertura:** se ha aplicado el criterio "Each Use", asegurando que cada valor interesante identificado sea utilizado al menos en un caso de prueba.

2. Definición de Valores de Prueba

Se han identificado valores clave para los atributos de Cliente y Viaje, cubriendo todas las ramas lógicas del método calcularTarifa:

- **Valores de Edad (Límites):**
 - **17:** Límite para tarifa "Pajarillo".
 - **18 y 25:** Fronteras para tarifas "Gorrión", "Viaja ahora que puedes" y "Atreviéndose a saltar del Nido".
 - **26:** Límite inferior para tarifas de adultos de media/alta renta.
- **Lógica de Ingresos y Destinos:**
 - **20.001 y 34.999:** Rango intermedio para tarifa "Conoce Europa".
 - **35.001:** Umbral para tarifas internacionales ("Conoce el Mundo").

- "Europa", "Asia", "America": Verificación de equivalencia ignorando mayúsculas/minúsculas.
- **Condiciones Booleanas y de Clase:**
 - **Frecuencia (6, 3):** Valores límite para activar descuentos por recurrencia.
 - **Tipo de Billete:** Alternancia entre TURISTA y BUSINESS según la categoría de edad/ingresos.
 - **Situación Personal:** Combinaciones de trabaja, independizado y conNinos para validar las sub-ramas de las tarifas.
- **Casos de Retorno Nulo:**
 - Configuraciones que no cumplen ningún criterio (ej. mayor de 25 con ingresos < 20.000) para asegurar que el sistema devuelve null.

3. Resolución de Tests Fallidos y Correcciones de Código

Durante la fase de ejecución de pruebas y configuración del entorno, se detectaron fallos técnicos que obligaron a refactorizar el archivo de configuración del proyecto (pom.xml) para cumplir con los requisitos de calidad:

A. Configuración de Cobertura Mínima (JaCoCo)

- **Fallo:** El reporte de JaCoCo no validaba automáticamente el umbral de calidad, permitiendo compilaciones con una cobertura de código insuficiente según los requisitos del proyecto.
- **Solución:** Se editó el archivo pom.xml para incluir una regla de verificación (check) que obliga a que el ratio de instrucciones cubiertas sea, como mínimo, del 80%. Se añadió el siguiente bloque de configuración:

B. Acotación del Análisis de Pruebas

- **Fallo:** El análisis de ejecución de pruebas incluía todas las clases del proyecto, lo que generaba ruido en los reportes y dificultaba la evaluación específica de la lógica de negocio.
- **Solución:** Se configuró el maven-surefire-plugin para restringir la ejecución de tests únicamente a la clase que contiene la lógica del problema (Tarifa). Para ello, se aplicó un filtro de inclusión en la configuración del plugin:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>3.0.0</version>
  <configuration>
    <includes>
      <include>**/Tarifa*Test.java</include>
    </includes>
  </configuration>
</plugin>
```

testing_p2_3c1

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Ctry	Missed	Lines	Missed	Methods	Missed	Classes
tarifa	98 %	86 %	8	31	1	39	1	5	0	1	0	1
Total	3 of 193	98 %	7 of 52	86 %	8	31	1	39	1	5	0	1

Created with JaCoCo 0.8.11 202310140853