

Examen de Promoción - 2025

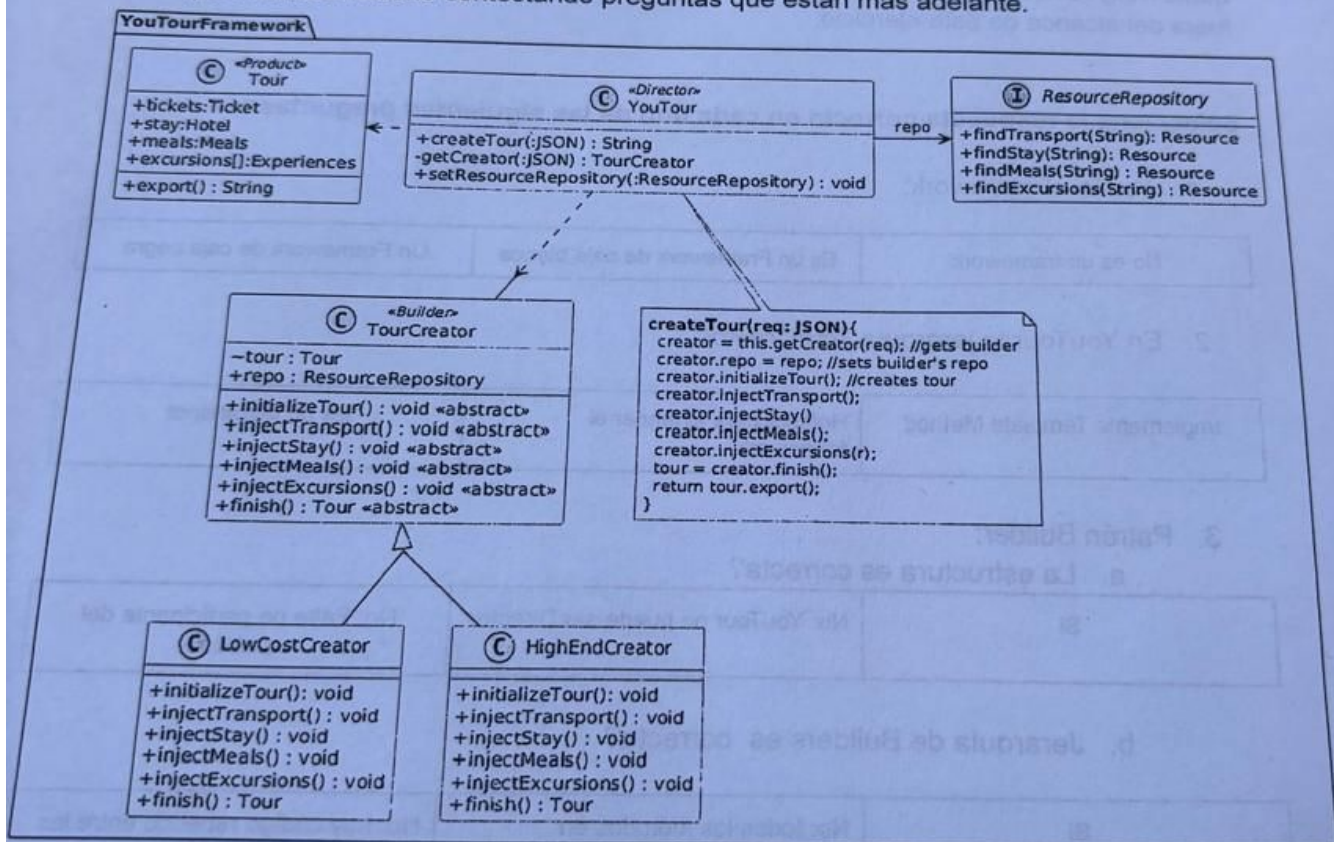
Nombre: GUAYMOS MARÍAS

Nro. alumno: 2306110

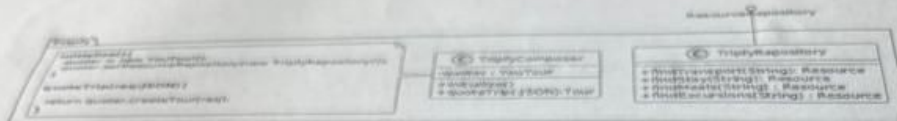
Ejercicio 1 (52 puntos)

Una empresa planea desarrollar YouTour, un framework de cotización de paquetes turísticos.

El framework fue diseñado a partir del patrón Builder. Antes de ser desarrollado se le solicita a Ud. que evalúe el diseño contestando preguntas que están más adelante.



El producto a ser construido es Tour. El framework provee una jerarquía de Builders. El builder a utilizar se elige en el método YouTour.getCreator(). Los Builders obtienen las partes utilizadas en la construcción consultando un repositorio representado por la interface ResourceRepository. El repositorio recibe consultas como un String (el detalle de las consultas está fuera del alcance del ejercicio).



Instancias de YouTour como por ejemplo Tripty instancian el framework proviendo una implementación de ResourceRepository que puedan usar los "builders". El método quoteTrip() recibe un pedido ("req") en formato JSON. Los detalles del mismo están por fuera del alcance de este ejercicio.

Seleccione la respuesta correcta en cada uno de las siguientes preguntas::

1. YouTourFramework:

No es un framework	Es un Framework de caja blanca <input checked="" type="checkbox"/>	Un Framework de caja negra <input type="checkbox"/>
--------------------	--	---

2. En YouTour la jerarquía TourCreator:

Implementa Template Method	Hotspot para extender el framework <input checked="" type="checkbox"/>	Es un frozen spot <input type="checkbox"/>
----------------------------	--	--

3. Patrón Builder:

a. La estructura es correcta?

Si <input checked="" type="checkbox"/>	No: YouTour no puede ser Director	No: Falta un participante del patrón
--	-----------------------------------	--------------------------------------

b. Jerarquía de Builders es correcta?

Si <input checked="" type="checkbox"/>	No: todos los métodos en TourCreator son abstractos	No: hay código repetido entre las subclases
--	---	---

c. Método YouTour.createTour() es correcto?:

Si <input checked="" type="checkbox"/>	No: invoca métodos del builder que no están implementados	No: debería estar implementado TourCreator (para ser heredado por las subclases)
--	---	--

Guaymas Matías

4. Se requiere agregar la posibilidad de crear Tours (economicos y VIP) sin transporte.
Dada el diseño existente debe:

Crear subclases de TourCreator	Modificar createTour()	Crear una nueva clase que funcione como Director	No se puede implementar ese requerimiento
--------------------------------	------------------------	--	---

Presente el diagrama de clases resultante (en esta hoja)

Ejercicio 2 (12 puntos)

Seleccione todas las afirmaciones correctas (y solo esas)

1. El refactoring "Extract Method" (Extraer Método) puede aplicarse para eliminar el mal olor "Código duplicado". ✓
El refactoring "Replace Temp With Query" (Reemplazar temporal con consulta) puede aplicarse para eliminar el mal olor "Envidia de atributo".
El refactoring "Move Field (Mover Atributo) puede aplicarse para eliminar el mal olor "Inappropriate Intimacy" (Intimidad inapropiada). ✓
Solo debemos invertir esfuerzo en refactoring cuando el código se vuelve ilegible.

Ejercicio 3 (12 puntos)

7p. Seleccione todas las afirmaciones correctas (y solo esas)

- ☒ 1. Un refactoring bien hecho, nunca rompe los tests de unidad.
- ☒ 2. Si hago refactoring con mucho cuidado, no es necesario que programe ni ejecute tests.
- ☒ 3. Durante el refactoring, los tests de unidad son necesarios para asegurar que no se altera el comportamiento del artefacto de software.
- ☒ 4. Si aplico refactorings con una herramienta automática como Eclipse (o IntelliJ, o similar), los tests se actualizan de manera automática en todos los casos.

9,6p. **Ejercicio 4 (12 puntos)**

Seleccione todas las afirmaciones correctas (y solo esas) que describan los escenarios en donde es necesario usar Test Doubles

- ☒ A. Es necesario disminuir el número de casos de prueba
- ☒ B. Es necesario probar la funcionalidad de un objeto independientemente del comportamiento de otro objeto, cuyos mensajes son invocados por la funcionalidad siendo testeada.
- ☒ C. Son indispensables para aumentar la cobertura del testing.
- ☒ D. Es necesario verificar salidas indirectas del sistema
- ☒ E. Hay objetos de los que depende el objeto a testear que aún no fueron creados.

8p. **Ejercicio 5 (12 puntos)**

Seleccione todas las afirmaciones correctas en la siguiente lista (y solo las correctas)

- ☒ 1. Los tests de unidad se programan una vez que el programa no sufrirá más cambios.
- ☒ 2. Test Driven Development es una metodología que evita que se genere Deuda Técnica.
- ☒ 3. En Test Driven Development, cada ciclo de desarrollo se conoce como "Red - Green - Refactor".
- ☒ 4. Puede ocurrir que un refactoring rompa los tests de unidad aunque el refactoring sea correcto.
- ☒ 5. En la metodología Extreme Programming se considera que la arquitectura surge desde el código