EJERCICIO (30 min.)

# **HAPPY PATH - EXCEPTIONAL PATH - CASOS MÁS SIMPLES PRIMERO**

**PROBAR EL MÉTODO “AGREGARLINEA”**

1. Revisa el método “AgregarLinea” y entiende su funcionamiento”.
2. ¿Cuáles son los comportamientos o secuencias de ejecución que se deben probar en este método?



1. Describe cada uno de los comportamientos utilizando la convención de nombres “NombreMétodo\_Condición\_ComportamientoEsperado”.






1. De estos comportamientos, ¿cuál es el más simple para comenzar a probar?
2. Dentro de Eclipse, implementa la prueba para el comportamiento más simple. Utiliza la convención de nombres “NombreMétodo\_Condición\_ComportamientoEsperado”.

**AYUDA***: No siempre el método bajo prueba devuelve un resultado, recuerda que para estos casos debemos buscar otro método a través del cual consultemos el estado de la clase.*   
El método “AgregarLinea” no devuelve ningún resultado pero altera internamente el estado de las líneas del carro de compras, ¿Qué método público de la clase “CarroCompras” nos devuelve información sobre las líneas? Utiliza este método dentro de la prueba para verificar que se ha alterado correctamente su estado.

1. Dentro de Eclipse, implementa la prueba para el siguiente comportamiento.

**HAPPY PATHS – CASOS SIMPLES PRIMERO - EXCEPTIONAL PATHS**

1. Revisa el método “ActualizarLinea” y entiende su funcionamiento”.
2. ¿Cuáles son los comportamientos o secuencias de ejecución que se deben probar en este método? ¿Cuál es su orden recomendado de implementación?.

1.

2.

3.

1. Describe cada uno de los comportamientos utilizando la convención de nombres “NombreMétodo\_Condición\_ComportamientoEsperado”.

1.

2.

3.

1. Dentro de Eclipse, implementa las pruebas para todos los comportamientos siguiendo el orden recomendado, no olvides utilizar la convención de nombres: “NombreMétodo\_Condición\_ComportamientoEsperado”.

EJERCIOCIO (15 min.)

# TEST FIRST PROGRAMMING

Se necesita implementar el nuevo método “RemoverLinea” dentro de la clase “CarroCompras”.

* El método debe tener la siguiente firma: “public void removerLinea(int productoId)”.
* El método debe cumplir los siguientes requerimientos:
* Remueve una línea del carro de compras.
* Si no se encuentra una línea debe lanzar una excepción “IllegalStateException”.

Implementa el nuevo método siguiendo el siguiente proceso:

1. Dentro de la clase “CarroComprasTest”, crea una nueva prueba para el primer requerimiento.
2. Completar el código de producción hasta que compile la prueba, no agregar lógica en el código de producción. (Crear únicamente el método “RemoverLínea” sin ningún contenido, esto hará compilar la prueba)
3. Ejecutar la prueba y comprobar que esta falle.

Aún no se ha implementado la lógica del requerimiento por eso debe fallar la prueba.

1. Implementar el requerimiento dentro del código de producción.

AYUDA:

* El método “buscarLinea” de la clase “CarroCompras” nos permite para encontrar una línea dado el id del producto.
* Las listas del tipo “Set” tienen un método “remove” que permite eliminar elementos dentro de la lista.
* La lógica es similar al método “ActualizarLinea”.

IMPORTANTE:

Solo implementar el código necesario que haga pasar la prueba.

1. Ejecutar la prueba y comprobar que esta pase.
2. Repetir para el siguiente requerimiento.

¿Qué beneficios trae escribir la prueba antes en comparación a escribirla después del código?