

## DOCUMENTACIÓN DE LOS CAMBIOS DE CÓDIGO:

1-(FRAN). Simplificación de calculaDato: Creación del método aplicarDescuentosGenerales factorizando la cantidad de ifs.

2-(FRAN). Factorización de aplicarDescuentosGenerales: Optimización en base PMD del método.

3-(GUILLE). Update en el método validarProducto: Usamos las colección set para eliminar los duplicados . Con el .of introducimos los valores, significa que solo va a contener esos valores como por defecto no puede agregar ni eliminar ni modificar nada de esa lista, con el map lo comparamos por clave.

4-(FRAN). Simplificación de return en calculaDato: Simplifica dos if junto al return, devolviendo a su vez únicamente dos decimales.

5-(DAVID). Change on if: Uso switch para optimizar el if que había anteriormente y así lo simplifico.

6-(ELENA). Update of aplicarCuponDescuento: Hacemos un switch y creamos una variable para optimizar el código.

7-(FRAN). Update EntornosFactorizar.java: Comentario return.

8-(DAVID). Create Enum TipoUsuario: Creo un enum para simplificar el método aplicarDescuentoPorUsuario.

9-(DAVID). Update Usuario including TipoUsuario: Importo el enum a la clase usuario eliminando los tres boolean que había antes del cambio.

10-(DAVID). Update method aplicarDescuentoPorUsuario: Método que aplica el descuento según el enum TipoUsuario

11-(DAVID). Update if on method calculaDato: isBlank() en vez de isEmpty(): Así cubre también cupones que sean solo espacios " ". Estilo más fluido y limpio.

12-(GUILLE). Update calculoDato on new method: Se separa el aplicarCuotas en un método aparte y se reduce la complejidad de las cuotas eliminando @aplicarCuotas sustituida por la aplicación de las cuotas si es mayor que 0.

13-(GUILLE). New enum MetodoDePago: Añadimos un nuevo enum reduciendo la complejidad de método de pago

14-(GUILLE). Update Usuarios: Añadimos con un atributo el método de pago con su getter.

15-(GUILLE). Update calculaDato: Se añade en el método de aplicarDescuentosYCargosGenerales los cargos según el método de pago

16-(GUILLE). Update EntornosFactorizacion: Corrección.

17-(FRAN). Update Usuario: Las variables externas @tieneTarjetaFidelidad y @saldoTarjeta, se han añadido como atributos de la clase Usuario.

18-(FRAN). Update aplicarDescuentosYCargosGenerales: Actualizar método según la actualización de la clase Usuario, ademas de añadir imports para el correcto de Set y Map.

19-(FRAN). Update Usuario Se introduce la variable independiente @esMiembroVip como atributo de la clase Usuario ayudando a reducir la complejidad del código y a su vez su fluidez.

20-(FRAN). Update calculaDato & aplicarDescuentosYCargosGenerales: Actualización de dichos metodos, debidos a los cambios pertinentes de la clase Usuario.

- 21-(FRAN).** Add new Class called Producto: Se ha introducido la clase Producto donde se han asignado como atributos las variables independientes: @tipoProducto, @categoriaProducto, @precioBase, @precioEnvio y @cantidad.  
Mejorando así la fluidez del programa y el orden de la mano de la reducción de complejidad del programa.  
Actualizándose consigo los métodos pertinentes donde son necesarios estos atributos. Con la actualización de los mismos el método validarProducto ha sido introducido como static dentro de la clase Producto.
- 22-(FRAN).** Update EntornosFactorizar: Actualización necesaria, acorde a los cambios pertinentes según la introducción de la nueva clase Producto
- 23-(DAVID).** Factorización método aplicarDescuentoPorUsuario: Importamos TipoUsuario y hacemos que tenga un solo return con un switch dentro de una variable para dar valor.
- 24-(FRAN).** Add new method calcularBase: Se ha añadido un método dentro de la clase producto para reducir la complejidad dentro del método aplicarDescuentosYCargosGenerales dentro de la clase EntornosFactorizar.
- 25-(FRAN).** Update aplicarDescuentosYCargosGenerales and calcularBase: Se añade descuento para eliminar un if y comprimir los métodos para mayor eficiencia
- 26-(FRAN).** Add new method descuentoTarjetaFidelidad in Usuario's class: Nuevo método en la clase Usuario simplificando el método descuentoTarjetaFidelidad de la clase EntornosFactorizar.
- 27-(FRAN).** Add new method in Usuario's class: Nuevo método que simplifica el método aplicarDescuentosYCargosGenerales.
- 28-(FRAN).** Add new method recargoMetodoPago in Usuario's class: Nuevo método que simplifica el método aplicarDescuentosYCargosGenerales, cambiando un switch de 6 líneas en un return de 1.
- 29-(FRAN).** Optimize all new methods: Nuevos métodos añadidos se han optimizado con la base de PMD.
- 30-(DAVID).** Change long name tieneTarjetaFidelidad to tarjetaF: Cambiamos el nombre para evadir el error de variable demasiado larga.
- 31-(ELENA).** Update of optimización del método validarProducto: Optimización del método validarProducto.
- 32-(ELENA).** Update EntornosFactorizar: Update aplicarCuota, aplicar cuota no afectaba al total así que lo he modificado para que lo asigne al total.
- 33-(GUILLE).** Update en calculaDato: Optimizamos este código estableciendo las llaves en los if.
- 34-(GUILLE).** Update aplicarCuota: Actualizamos el método estableciendo los final y creando una nueva variable en el método con el valor de total para poner un final en el parámetro total
- 35-(ELENA).** Update of Producto: Optimizo el código y le pongo finals para que no se cambie el valor.
- 36-(FRAN).** Update aplicarDescuentoPorUsuario: Eliminación de default innecesario.
- 37-(ELENA).** Update of Usuario: Corrección de fallo en el constructor Usuario
- 38-(ELENA).** Update of Usuario: Optimizo el código y le pongo finals para que no se cambie el valor.

39-(GUILLE). Update Usuario: Optimización.

40-(FRAN). Update throw of calculaDato.

41-(ELENA). Documentación.