

Trabajo Práctico 1 – Java

Algoritmos y Programación 3

Curso noche

Segundo cuatrimestre del 2017

Dan Tchechenistky

100117

dtcheche@gmail.com

Introducción:

El presente informe reúne la documentación de la solución del primer trabajo práctico de la materia Algoritmos y Programación III que consiste en desarrollar una aplicación de un sistema de comercio electrónico de compra en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

Supuestos:

Al momento de implementar la solución se tomaron ciertas suposiciones para facilitar el desarrollo del programa.

En un principio se entiende que el usuario utiliza el programa correctamente. Al querer agregar un producto o un cupón a una compra estos ya fueron creados previamente.

Por otro lado, se supone que al querer agregar un producto con un nombre ya existente a AlgoBay se cambiará el valor del producto ya existente por el nuevo valor y no se creará un producto con el mismo nombre y otro valor.

Finalmente, al momento de crear un cupón se va a mantener este cupón creado por más que se use en una compra, los cupones no se eliminan luego de ser usados en una compra.

Modelo de dominio:

AlgoBay: clase con la cual interactúa el usuario. Se encarga de la creación y manejo de todas las compras. Permite crear productos, cupones, compras y obtener detalles sobre estos.

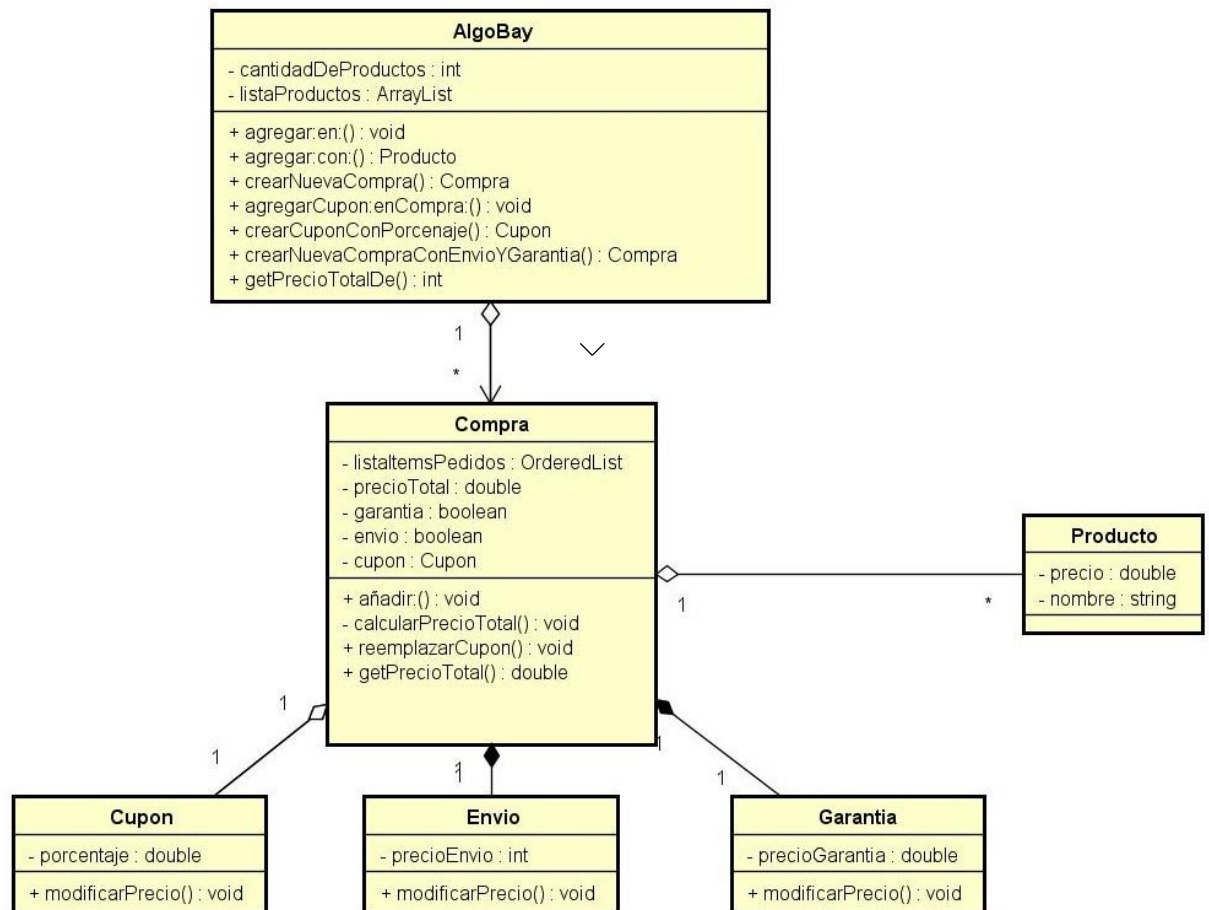
Compra: clase responsable del manejo interno de las compras. Se encarga de que los pedidos se manejen de forma correcta agregando los productos y calculando el precio final de la compra dependiendo de las distintas variables que apliquen (con envío, con cupón, con garantía...).

Producto: clase responsable de la creación de los productos con su respectivo nombre y precio.

Cupón: clase responsable de la creación de cupones con su respectivo nombre y porcentaje de descuento.

Envío: clase responsable de clasificar en caso necesario una compra bajo envío, modificando así el precio final de esta.

Garantía: clase responsable de clasificar en caso necesario una compra con garantía, modificando así el precio final de esta.

Diagrama de Clases:Detalles de implementación:

Una de las implementaciones más interesantes para evaluar es la responsable de modificar el precio final de la compra dependiendo de si esta tiene agregados ciertos elementos extra. Por ejemplo, un cupón, una garantía, un envío o todos estos al mismo tiempo. Resulta también de importancia que el precio debe modificarse en un orden específico en caso de contar con más de un beneficio.

Para resolver esto primero se calcula el precio total de la compra. Esto se hace recorriendo la `ArrayList` del objeto compra que contiene todos los productos de esta. Se obtiene el precio de cada producto y se lo suma a la variable de instancia `precioTotal`. Obtenemos así el precio final de la compra previo a modificación por posibles beneficios.

```

for(Producto producto : listaProductos){
    precioTotal = precioTotal + producto.getPrecio();
}

```

Luego, se evalúa el estado de las diferentes variables de instancia de Compra. Estas variables refieren individualmente si la compra debe o no ser con envío, con garantía o con cupón. Esto se detalla al momento de crear la compra desde la clase AlgoBay. Si se crea por ejemplo una compra crearNuevaCompraConEnvio, esto setea la variable de instancia envío a True y de esta manera el objeto compra sabe si debe o no agregar el precio del envío. Se decidió usar esta metodología debido a que estas variables se deben analizar en un orden específico para obtener el resultado correcto. Por ejemplo, primero se calcula la garantía y luego se le suma el precio de envío. Se evalúa entonces individualmente el estado de cada variable de instancia y se decide que modificaciones hay que hacerle al precio total. Para ello se utilizan entonces tres métodos que se llaman ordenadamente desde calcularPrecioTotal.

```
if (garantia){
    Garantia nuevaGarantia = new Garantia();
    precioTotal = nuevaGarantia.modificarPrecio(precioTotal);
}
if (envio){
    Envio nuevoEnvio = new Envio();
    precioTotal = nuevoEnvio.modificarPrecio(precioTotal);
}
if (cupon != null){
    precioTotal = cupon.modificarPrecio(precioTotal);
}
```

Aquí se le delega a cada la clase correspondiente que calcule a su manera como modificar el precio que se le pasa por parámetro.

Excepciones:

PorcentajeFueraDeRangoError: esta excepción se creó para el caso en el que el usuario quiera crear un cupón con un porcentaje incorrecto. Es decir, un rango fuera de 0% a 100%. Esta excepción fue creada con el finde de evitar que los precios se modifiquen a valores indeseados. Un cupón con un porcentaje fuera de rango llevaría el precio final de la compra a un precioTotal incorrecto.

PrecioNegativoError: en este caso se limita al usuario a poder crear productos únicamente con valores positivos, mayores o iguales a 0. De aquí se presupone que la empresa puede regalar productos, pero no pagarle a la gente para que se los lleve. Es algo posible, pero en el caso de una empresa parecería imprudente.

ProductoNoEncontradoError: esta excepción se lanza al momento en que el usuario quiere hacer get de un producto que nunca cargó. Esta excepción fue creada para que el usuario entienda que está intentando buscar un producto inexistente.

Diagrama de secuencia:

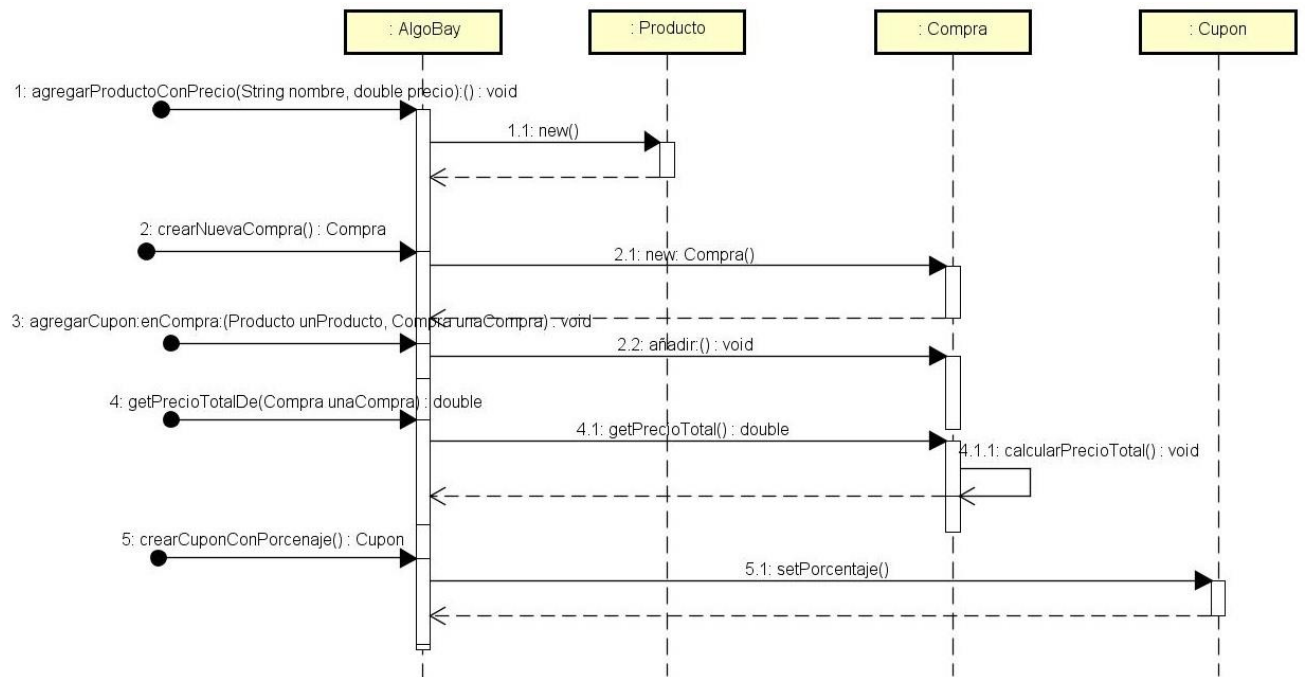


Diagrama de secuencia de funcionamiento de AlgoBay.

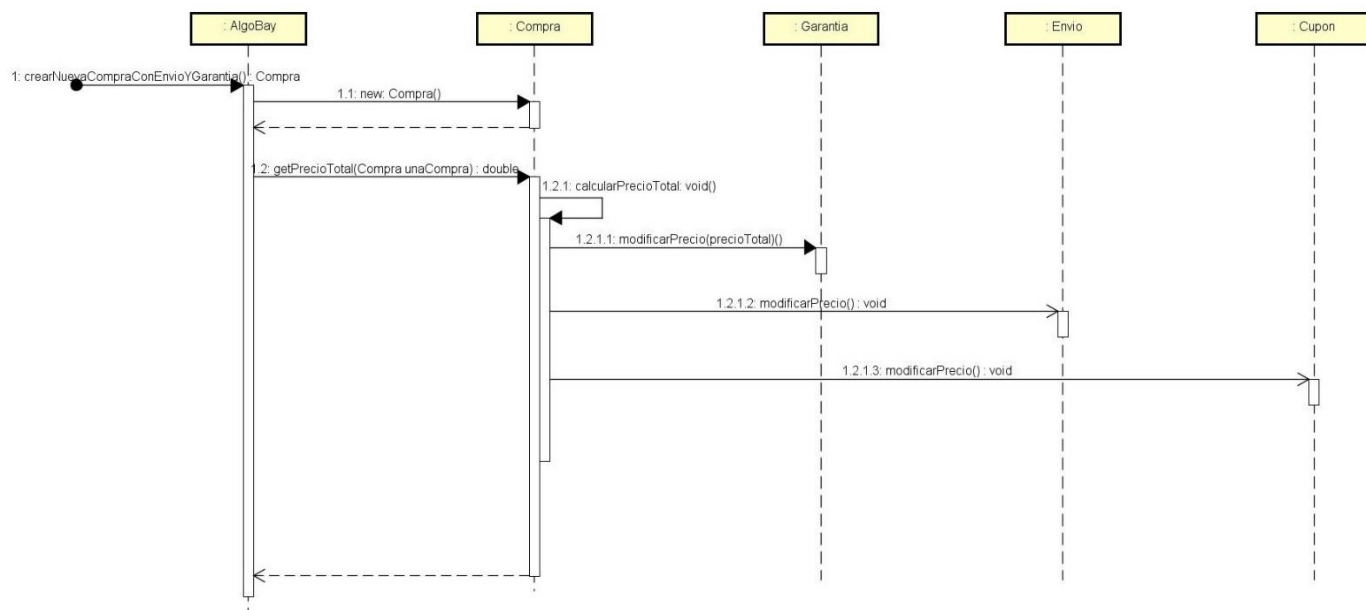


Diagrama de secuencia de una compra con envío y garantía.