

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III
Curso 1
Segundo cuatrimestre 2020

Alumno:	BILO, Lucas
Número de padrón:	103252
Email:	lbilo@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clase	2
4. Detalles de implementación	4
4.1. Clase AlgoPedidos	4
4.2. Clase Pedido	4
4.3. Clase Cupon	5
5. Excepciones	5
6. Diagramas de secuencia	6

1. 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 pedidos de productos y menus en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

1. Como máximo puede haber 1 cupón por pedido.
2. No pueden existir cupones con valor 0 o porcentaje 0. Ya que no tendría sentido el cupón.
3. Pueden existir cupones fijos con valor mayor al precio del pedido y cupones con porcentaje 100 como maximo, en ambos casos el precio final del pedido será 0.
4. Precios de productos y valor de cupones fijos son números flotantes, por lo tanto los precios de los pedidos también.
5. Los pedidos deben tener el nombre del cliente si o si, no se puede crear un pedido sin especificar el nombre. Lo mismo con los nombres de productos/menus, no pueden estar vacios.

3. Diagramas de clase

En el primer diagrama no están en detalle los objetos Pedido y Cupon, ya que si lo están en los otros 2 diagramas.

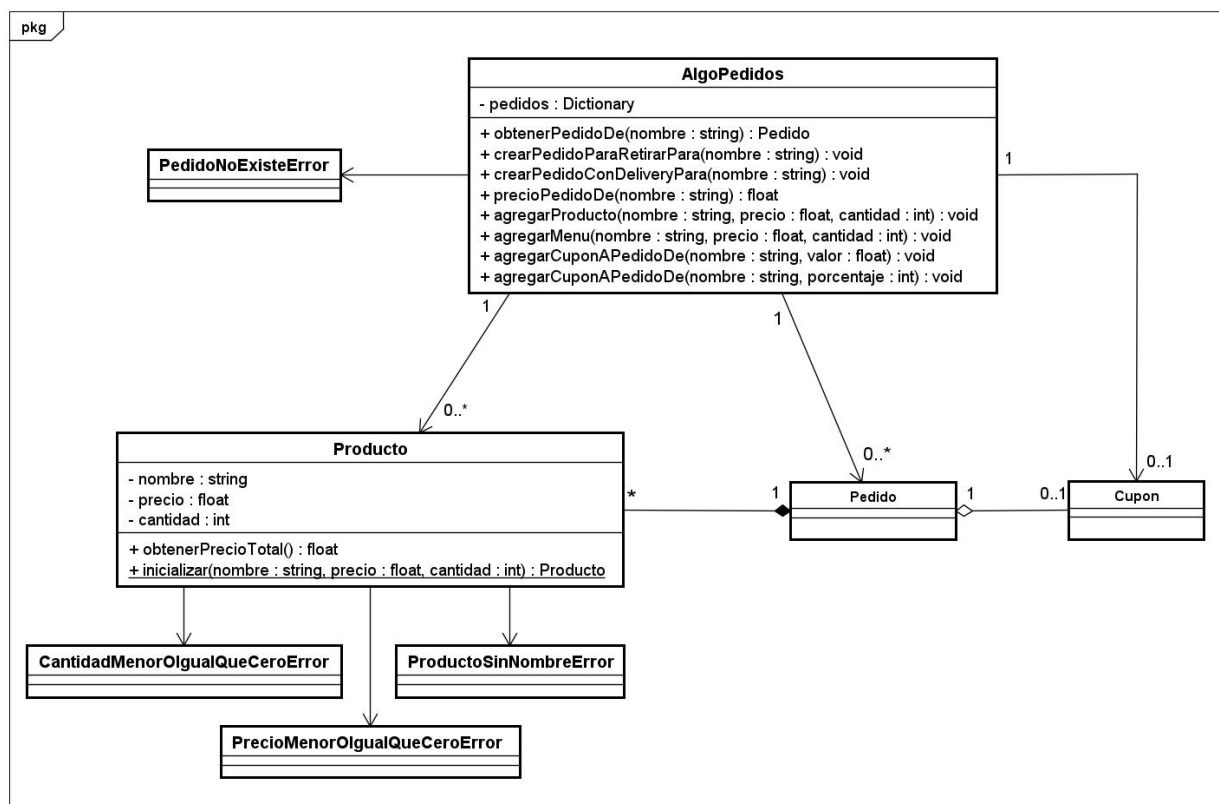


Figura 1: Diagrama general.

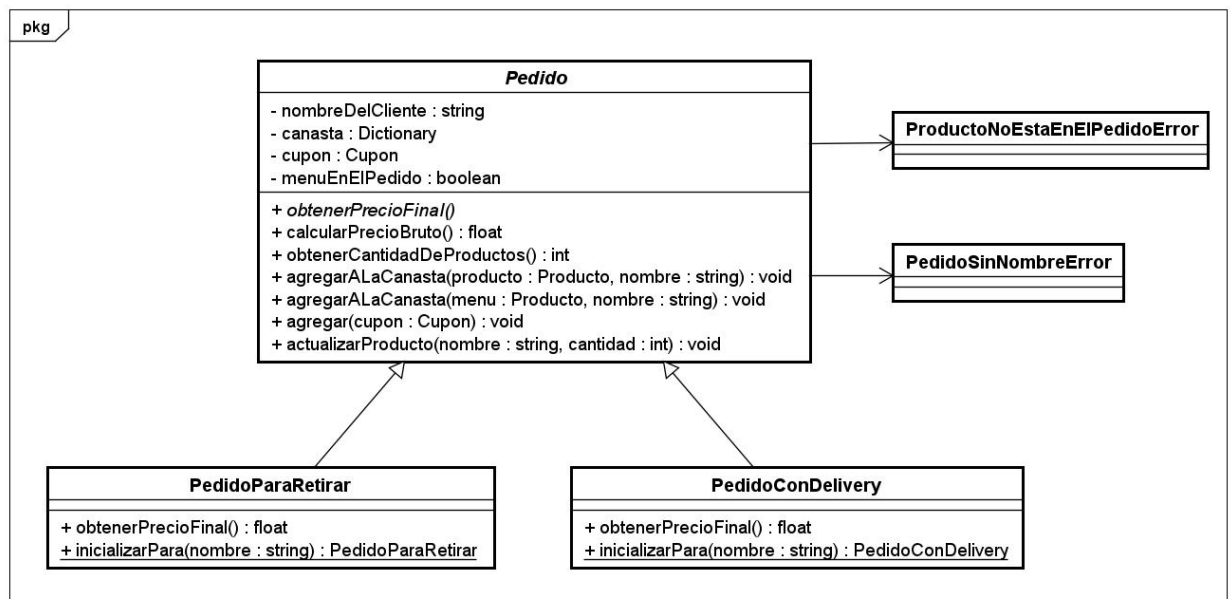


Figura 2: Herencia de Pedido y sus excepciones

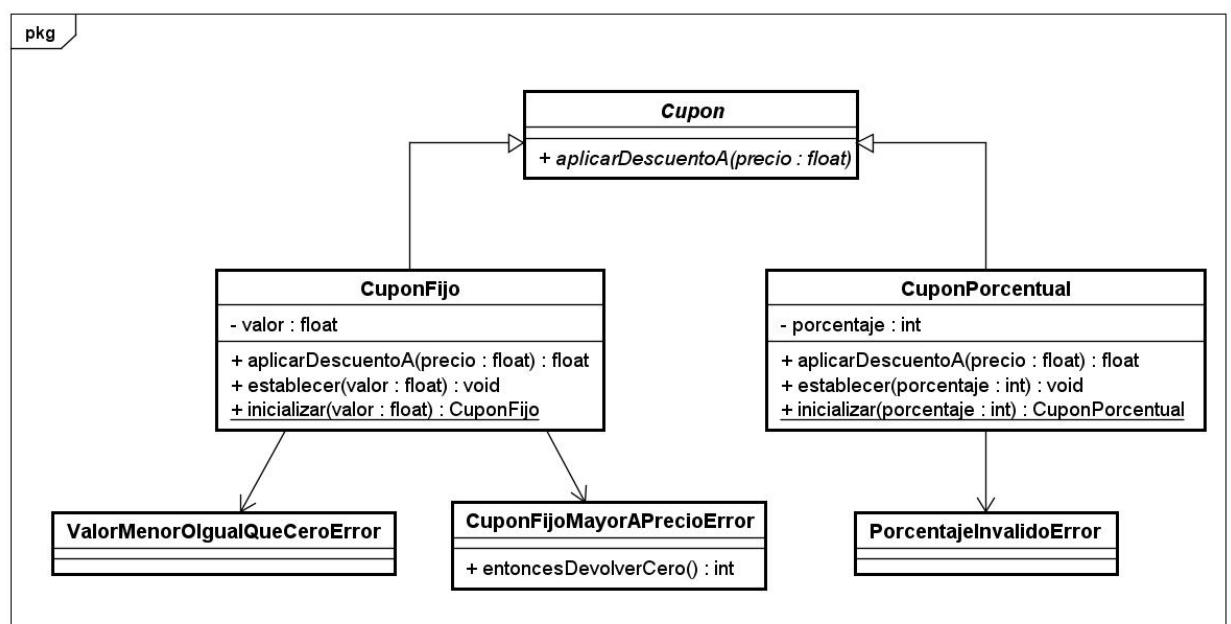


Figura 3: Herencia de Cupon y sus excepciones

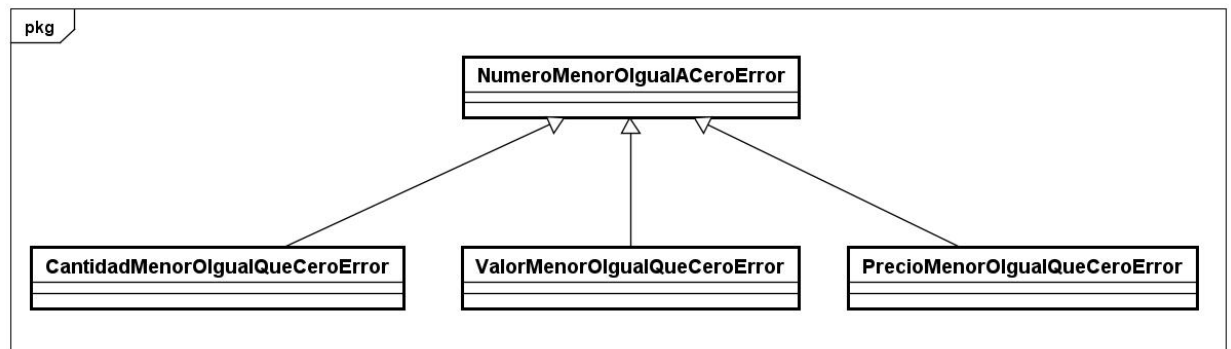


Figura 4: Herencia de NumeroMenorOIgualACeroError.

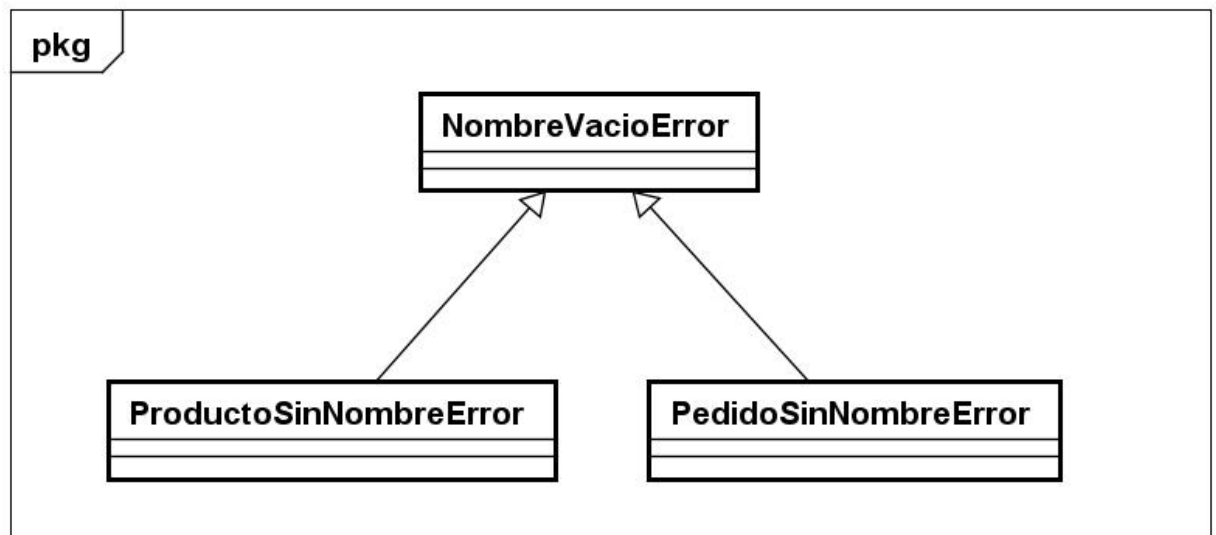


Figura 5: Herencia de NombreVacioError.

4. Detalles de implementación

4.1. Clase AlgoPedidos

Esta clase tiene un Dictionary de Pedidos. En la mayoría de sus mensajes delega la responsabilidad a alguno de los Pedidos de su diccionario.

4.2. Clase Pedido

Es una clase abstracta, de la cual heredan PedidoParaRetirar y PedidoConDelivery. Se aplica polimorfismo en ambas clases hijas, ya que ambas reaccionan distinto en la cuestión del precio final del pedido.

La clase en cuestión, tiene una variable booleana (menuEnElPedido) la cual, como dice el nombre, es para identificar si hay un menu en el pedido o no. Reconozco que no es la mejor opción

ya que luego tuve que usar un condicional para ver si aplicar un cupon o no, sinceramente no se me ocurrio otra forma de implementarlo.

4.3. Clase Cupon

Es una clase abstracta de la cual heredan CuponFijo y CuponPorcentual. Se aplico polimorfismo en ambas clases hijas, ya que las dos interpretan un descuento de formas distintas.

5. Excepciones

Exception NombreVacioError. Esta excepción fue creada para realizar una jerarquía de excepciones que puedan tener relación con nombres vacíos.

Exception ProductoSinNombreError. Hereda de NombreVacioError, esta excepcion la cree para el caso en que se crea un producto y no se especifica el nombre.

Exception PedidoSinNombreError. Hereda de NombreVacioError, esta excepcion la cree para el caso en que se crea un pedido y no se especifica el nombre del cliente.

Exception PedidoNoExisteError. Se creo para el caso en que se busca un Pedido por el nombre del cliente y este no existe.

Exception ProductoNoEstaEnElPedidoError. Se creo en el caso de que se busca un producto/menu, para quitarlo o actualizarlo, en el pedido y este no esta en el mismo.

Exception PorcentajeInvalidoError. Se creo para tratar el caso de inicializar un cupon porcentual con porcentaje mayor a 100 o porcentaje menor o igual a cero.

Exception CuponFijoAMayorPrecioError. En el momento de aplicar un descuento de cupon fijo, si este es mayor al precio devolvería un valor negativo. Con la excepcion se evito esto, en caso de que el valor a descontar sea mayor al precio, tiene un mensaje para poder devolver 0.

Exception NumeroMenorOIgualACeroError. Esta excepción fue creada para realizar una jerarquía de excepciones que tengan que ver con numeros menores o iguales a cero.

Exception CantidadMenorOIgualACeroError. Hereda de NumeroMenorOIgualACeroError. Se creo para lanzarse al momento de inicializar un producto con una cantidad cero o negativa y también para cuando se quiere actualizar una cantidad del producto.

Exception PrecioMenorOIgualACeroError. Hereda de NumeroMenorOIgualACeroError. Se creo para lanzarse al momento de inicializar un producto con un precio negativo o cero.

Exception ValorMenorOIgualACeroError. Hereda de NumeroMenorOIgualACeroError. Se creo para lanzarse al momento de inicializar un cupon fijo con un valor negativo o cero.

6. Diagramas de secuencia

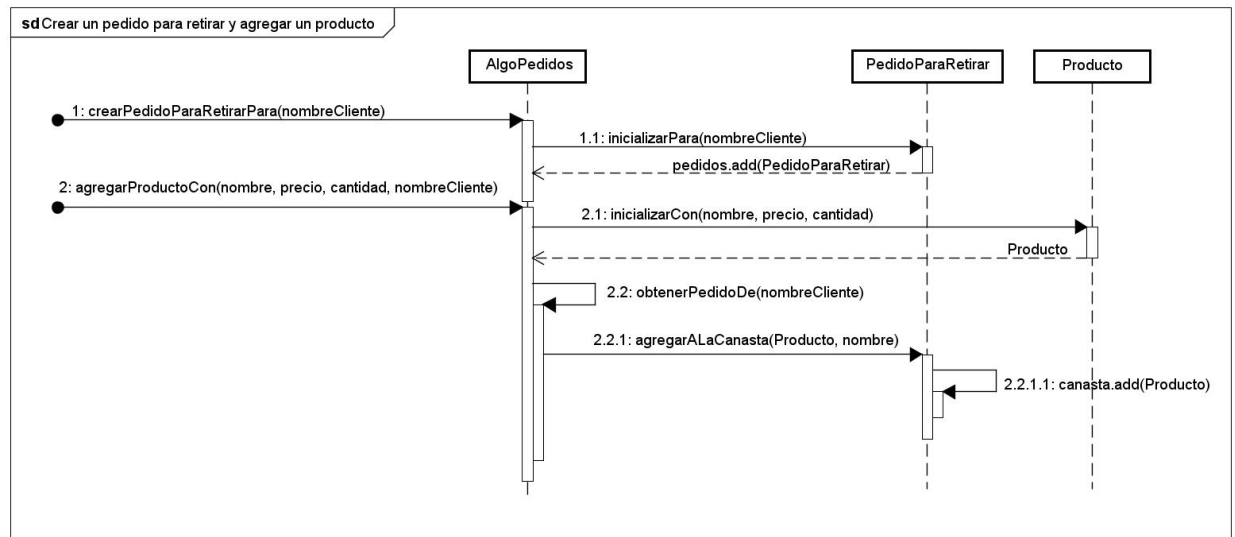


Figura 6: Crear pedido con producto.

En los siguientes diagramas se puede ver como es el proceso para calcular el precio total de un pedido. Dentro de la función obtenerPrecioTotal de Pedido hay un condicional para saber si aplicar o no el descuento del cupon.

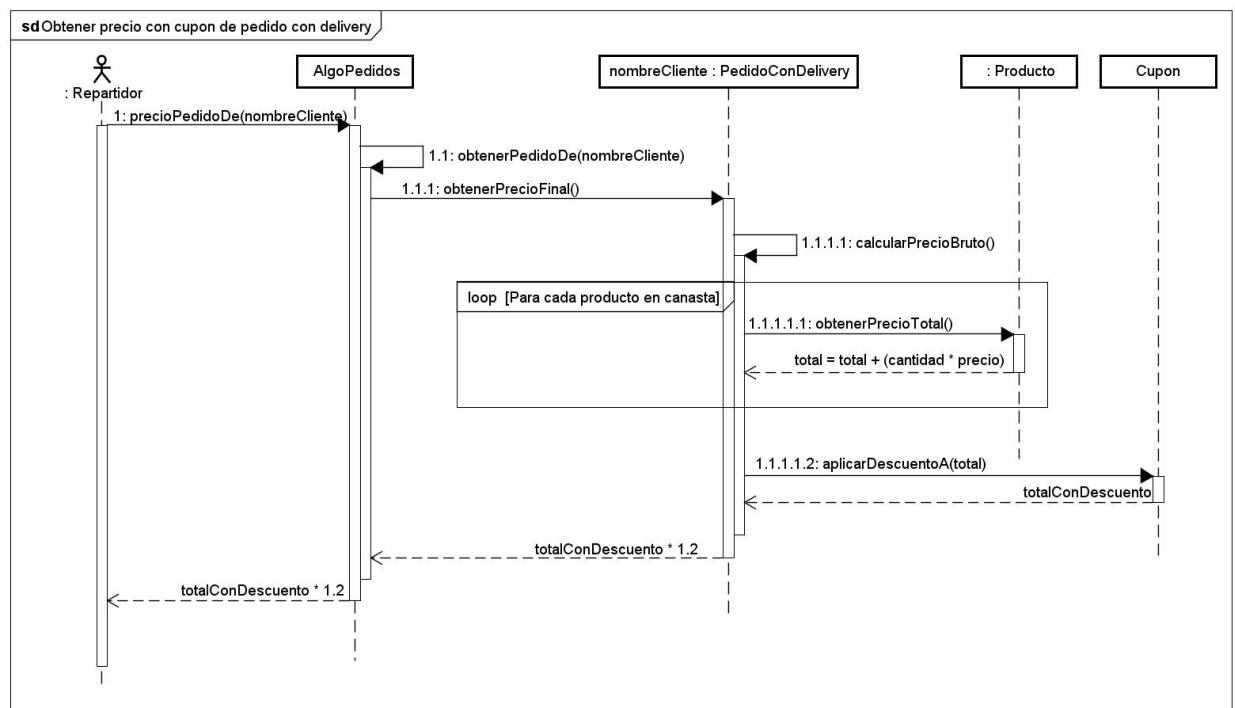


Figura 7: Obtener precio de un pedido con cupon.

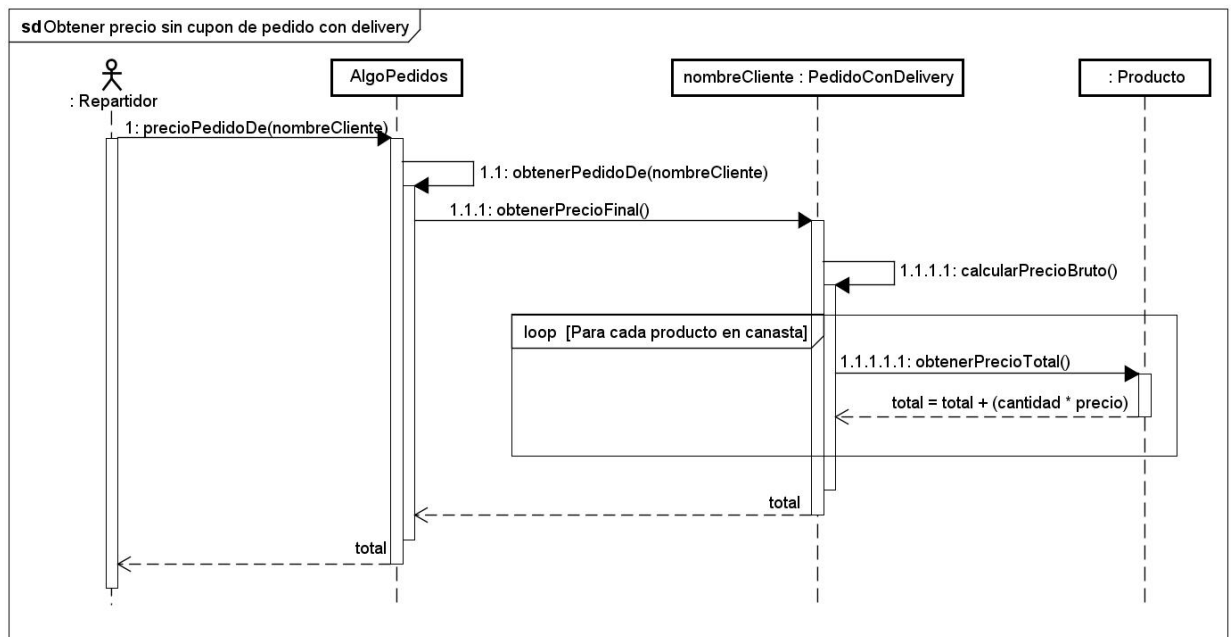


Figura 8: Obtener precio de un pedido sin cupon.