

# Trabajo Práctico 1 — Smalltalk

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

Alumno:	Caserío, Juan Cruz
Número de padrón:	104927
Email:	jcaserio@fi.uba.ar

## Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	3
5. Detalles de implementación	4
6. Excepciones	4
7. Diagramas de secuencia	5

## 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 comida en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

## 2. Supuestos

Como el programa supone ser una aplicación de pedidos desarrolle un sistema que admita mas de un pedido, a su vez, cada pedido puede tener una cantidad mayor o igual a cero de cupones y o encargos. Cuando un pedido tiene un encargo del tipo menu nose se aplican descuentos.

## 3. Modelo de dominio

El usuario solo puede enviar mensajes al objeto AlgoPedidos este almacena a los pedidos en un diccionario y se encarga de delegar los mensajes al pedido correspondiente. El pedido puede tener o no delivery, y esto cambia el costo de entrega, entonces decidi crear los objetos Pedido-ParaDelivery y PedidoParaRetirar que heredan de la clase Pedido, estos utilizan toda la interfaz de la clase abstracta Pedido pero tambien por polimorfismo cambian la respuesta al mensaje calcularCostoDeEntrega.

EL Pedido tienen asociado una lista de Cupones y otra de Encargos. Para los Encargos no se tiene el mismo comportamiento si un Encargo es un menu u otro producto para poder implementar esta diferencia cree a los objetos Producto y Menu que heredan de la clase Encargo y responden polimorficamente al mensaje noAplicaCupon, además el dominio del problema se ve mejor reflejado. Para el caso del Cupon este difiere en su comportamiento de acuerdo a si es Fijo o Porcentual, si bien podria haber implementado polimorfismo sin herencia simplemente llamando al metodo en cuestion para cada objeto de la lista de Cupones en el pedido aparecian algunas repeticiones de codigo que evite heredando estos objetos a la clase Cupon y se justifica ,ya que, tanto CuponFijo como CuponProcentual son cupones y se comportan como tal.

## 4. Diagramas de clase

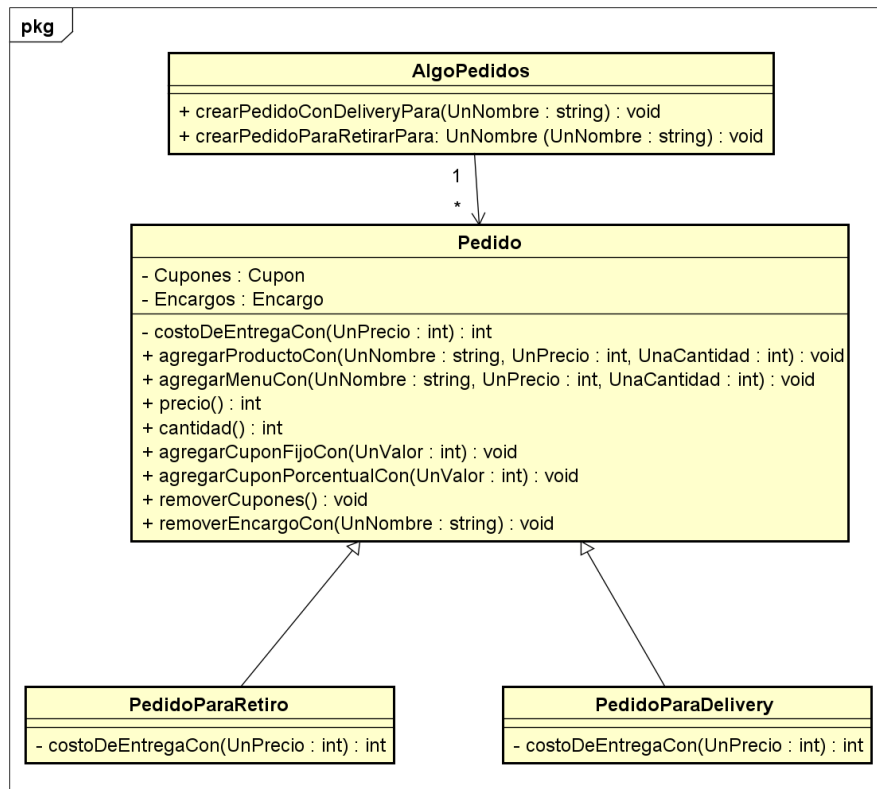


Figura 1: Diagrama de AlgoPedidos Y Pedido.

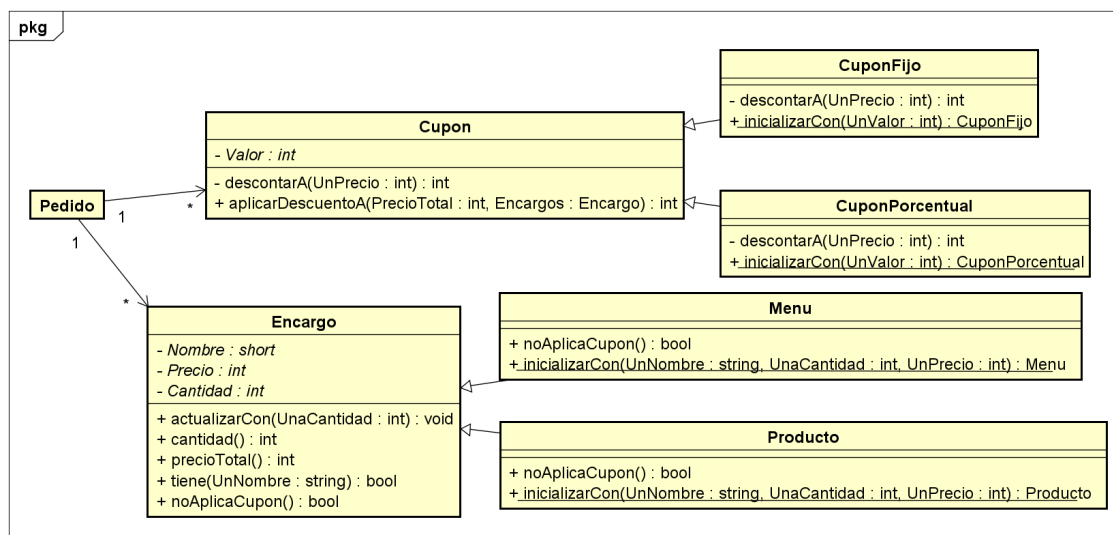


Figura 2: Diagrama de las clases asociadas al objeto Pedido.

## 5. Detalles de implementación

Un Problema picante que se presento fue pensar en resolver que pasa si se agrega un cupon, luego un menu, pero despues se elimina el menu.

Este problema me ayudo a pensar el modelo planteado, ya que, para llegar a una solucion necesitaba que el cupon no se elimine cuando se crea un menu pero de alguna manera tenia que habilitar o desabilitar el descuento de este, para mi la mejor solucion fue la respuesta polimor-fica dada por Menu y Producto al mandarles el mensaje noAplicaCupon, este incluso se podria profundizar y si se tienen otros cupones que si descuenten con menu u otros que no apliquen a producto es sencillo hacerlo. Al ser asi el modelo, cuando se elimina el menu que se encuentra en el pedido el cupon no va a ser rechazado, entonces si agregamos un cupon siempre va a estar(salvo que se elimine) y luego el mismo cupon verifica si puede usarse o no.

Tambien fue complicado pensar cuando aplicar el descuento, dado que no se sabia si se agregaria o eliminaria un menu o un cupon en el transcurso de la ejecucion, pero resolvi aplicar el descuento en el calculo final del precio y no ir modificando el precio a medida que se daban estos cambios. De esta manera calculaba una unica vez y de forma correcta el precio final.

## 6. Excepciones

**AplicarCuponSinEncargosError** Esta excepcion salta en el caso que un pedido no tenga en-cargos y se desee aplicar un cupon, se supone que los cupones descuentan, si no tienen nada a que descontarle no tiene sentido que se aplique ese descuento.

**CantidadInvalidaEnEncargoError** Este es un caso de error que salta cuando se quiere agregar un encargo con cantidad negativa, conceptualmente no tiene sentido.

**CuponConValorNegativoError** Se supone que los cupones aplican un descuento, si el valor del cupon es negativo el cupon no va a funcionar como corresponde.

**PrecioInvalidoEnEncargoError** Se supone que el programa simula una aplicacion de pedidos, no se puede dar que el precio sea negativo.

## 7. Diagramas de secuencia

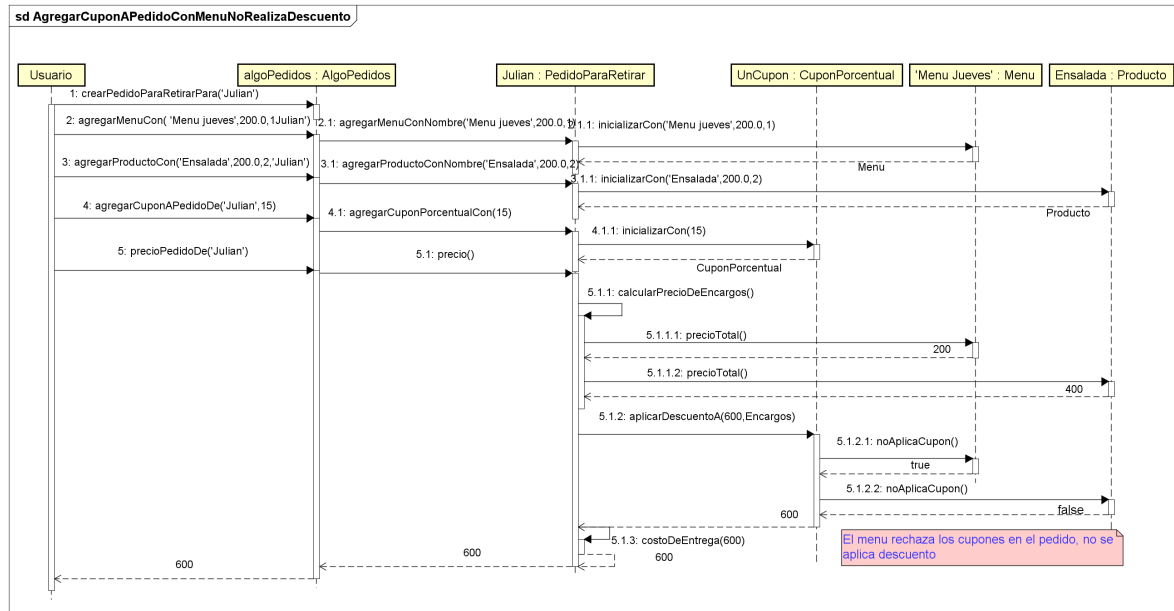


Figura 3: Primer diagrama de secuencia.

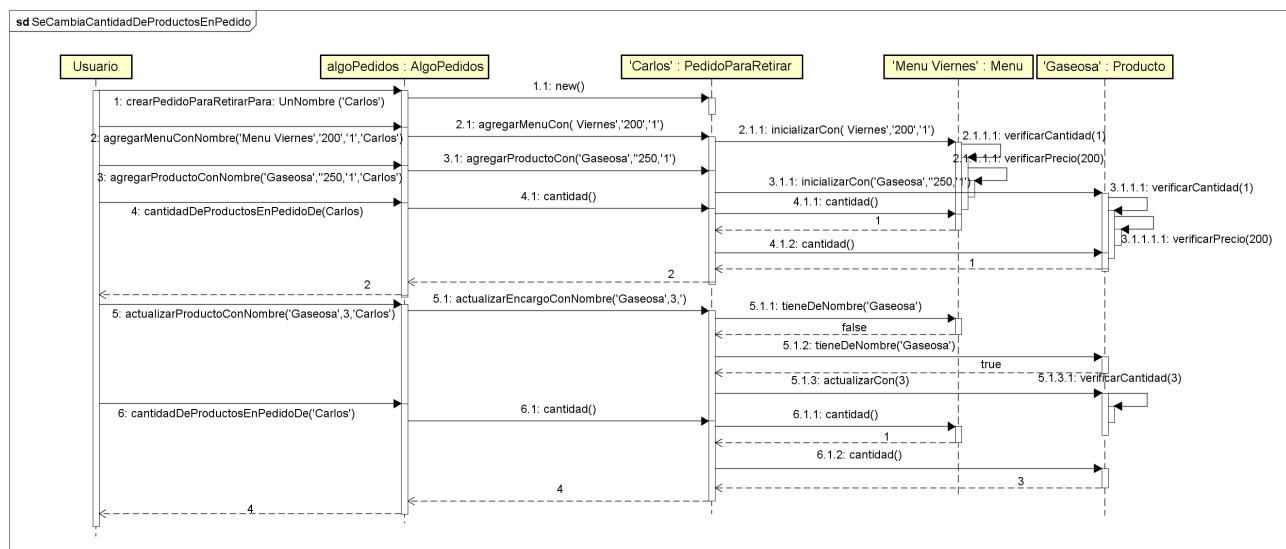


Figura 4: Segundo diagrama de secuencia.

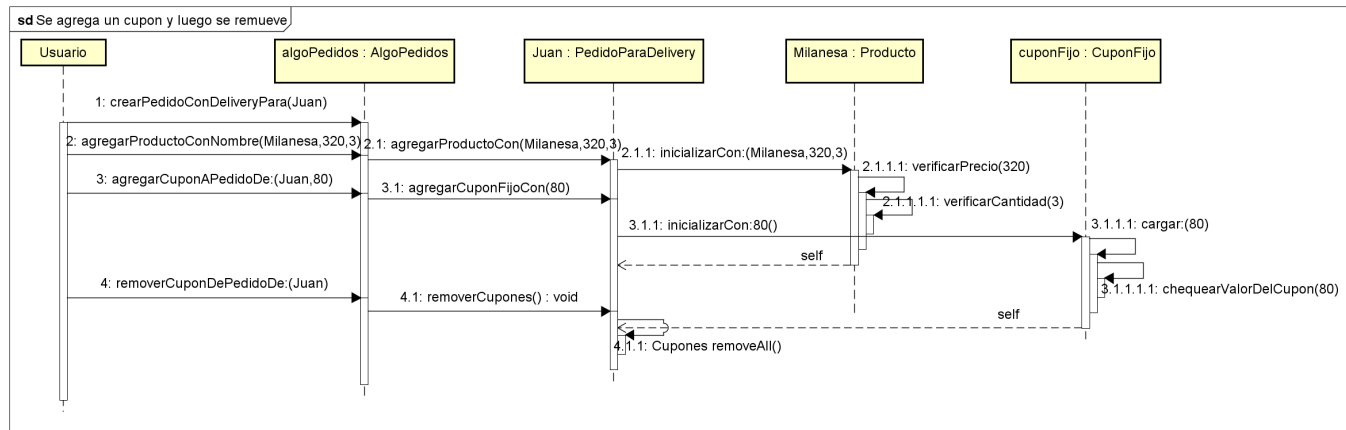


Figura 5: Tercer diagrama de secuencia.