

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III
Curso 2
Segundo cuatrimestre de 2020

Alumno:	Villores, Alejo
Número de padrón:	105285
Email:	avillores@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	2
5. Detalles de implementación	4
5.1. Asignacion de producto al usuario	4
6. Excepciones	4
7. Diagramas de secuencia	4

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 un sistema de pedidos de comidas en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

Como supuestos se contempla la aplicación de un menu por pedido para un usuario , además se toma como supuesto el uso de ambas categorías de cupones a la hora de aplicar los descuentos para cada persona que realizó un pedido.

Se toma como supuesto la utilización de usuarios con nombres diferentes para realizar los pedidos ya que de no ser así , una excepcion será lanzada indicando el error.

3. Modelo de dominio

El diseño del trabajo practico realizado fue llevado a cabo pensando como se realizan los pedidos en nuestra vida cotidiana ,haciendo un buen uso del paradigma,éste nos provee la oportunidad de realizar entidades irreales tales como son `CuponActivo` `CuponInactivo`"para una mejor solucion a nuestro problema.

El sistema de pedidos se trata de una clase denominada `AlgoPedidos` que tiene interaccion directa con los usuarios almacenados en una `Collection` , estos `Usuarios` poseen un `Pedido` el cual éste es el encargado de realizar los agregados ,modificaciones y los usos de los cupones por medio de métodos que delegan las diferentes acciones a sus respectivas clases; la idea principal de las delegaciones es seguir con la idea trabajada en clase de "Tell ,don´t ask". A su vez , de manera encapsulada se logra la idea de que cada objeto sepa su rol y sepa que debe hacer y a quien debe servir, es por eso que no se busca el usos de clases particulares como lo son "Delivery , Cupon , Producto , Pedido "de manera directa.

4. Diagramas de clase

Las siguientes ilustraciones representan el diagrama de clase , mostrando las relaciones estáticas de estas.Podemos apreciar las delegaciones y relaciones a a travez del flechado. Se buscó que se vea lo mejor posible. De igual manera está como archivo externo.

Este posee ocho(8) clases diferentes , `AlgoPedidos` , cuya clase contiene varios `Usuario` , esta a su vez contiene un `Pedido`. `Pedido` es una clase importantisima ya que la existencia de esta genera la existencia de las clases mas chicas como lo son `Producto` , `Delivery` , `CuponActivo` y `CuponInactivo`.

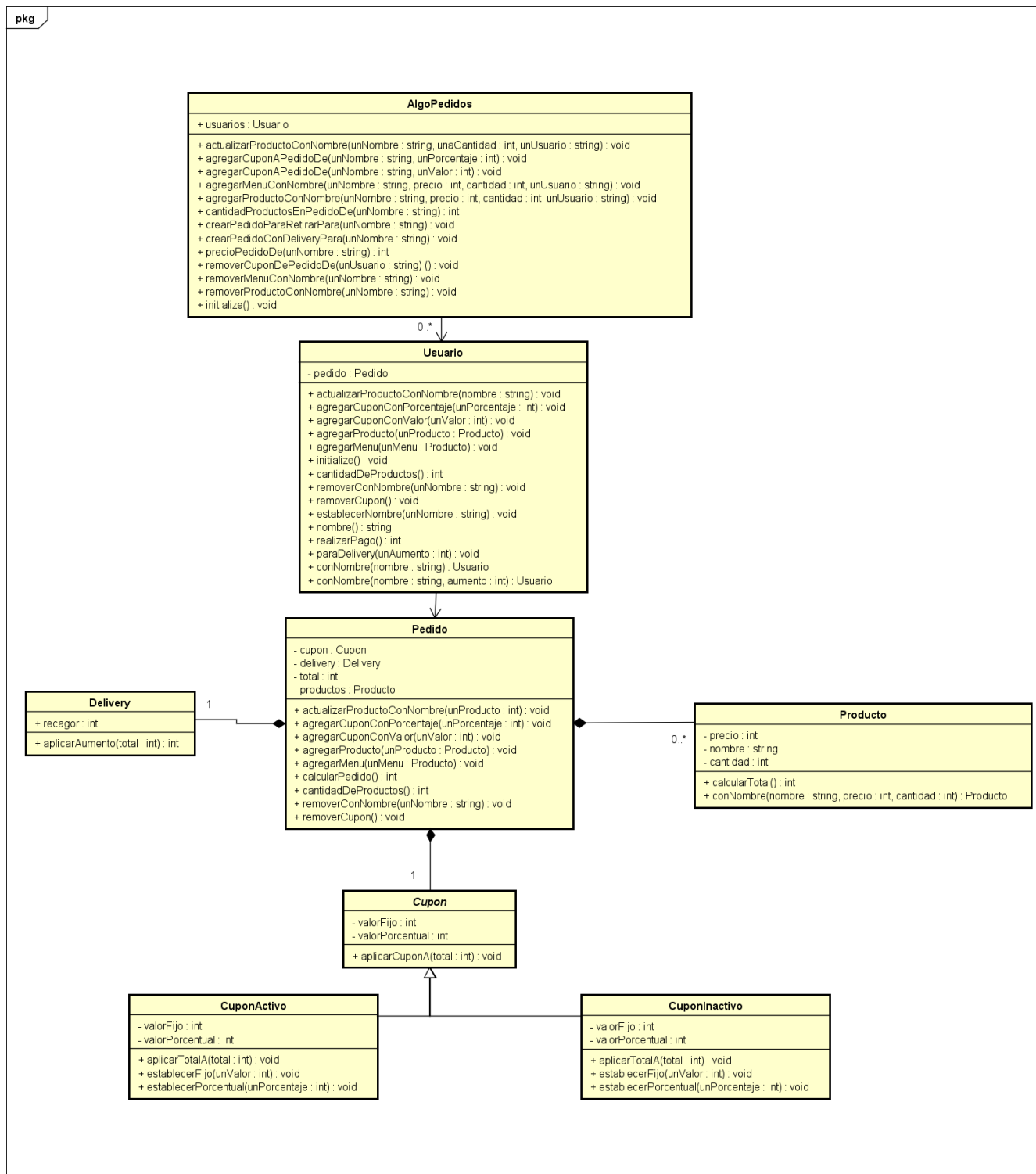


Figura 1: Diagrama de clases.

5. Detalles de implementación

5.1. Asignacion de producto al usuario

Al momento de pensar cuando se asigna o cuando se instancia un producto para el usuario correcto , la idea que se tuvo fue ponerse en el lugar del restaurante o lugar donde se realiza el pedido , es por eso que la implemtancion de esta está dada de la siguiente manera

```
| producto |
```

```
producto := Producto conNombre: unNombre precio:unPrecio yCantidad:unaCantidad.
(usuario detect: [:usuario | usuario nombre = unUsuario]).
ifNone:[ UsuarioNoEncontradoError signal])agregarProducto: producto.
```

6. Excepciones

UsuarioNoEncontradoError es lanzada cuando el usuario no se encuentra en la Collection de usuarios .

ProductoNoEncontradoError es lanzada cuando el producto no se encuentra en la Collection de pedibles .

PedidoConCeroError es lanzada cuando se desea hacer un pedido con cantidad de valor 0.

PorcentajeNoDisponibleError es lanzada cuando al usuario se le desea dar un porcentaje mayor a 100.

7. Diagramas de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

Para esta ocasión se muestra un diagrama de secuencia para los test mas completos como lo son el de agregarProducto..., precioPedidoDe y de actualizarElPedido... ,que personalmente considero los mas importantes y abarcativos.

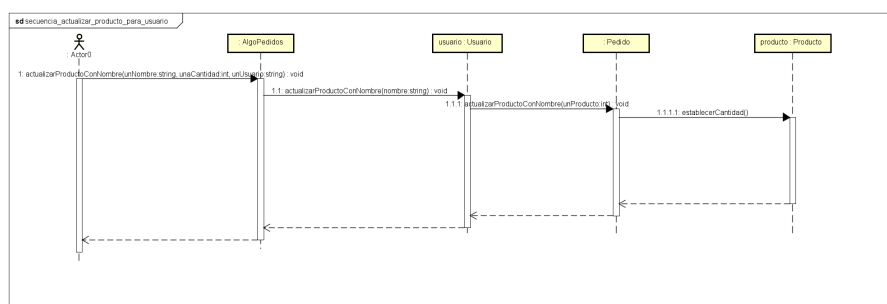


Figura 2: Secuencia de actualizar un producto

Esta secuencia comienza con un Actor que ejecuta el metodo actualizarProductoConNombre , éste lo que hace es , luego de detectar el usuario que posee el producto a actualizar llama al metodo actualizarProductoConNombre que delega al pedido la accion a realizar. El pedido detecta de sus productos aquel que su nombre coincide con el pasado por parametro y establece una nueva cantidad para aquél producto.

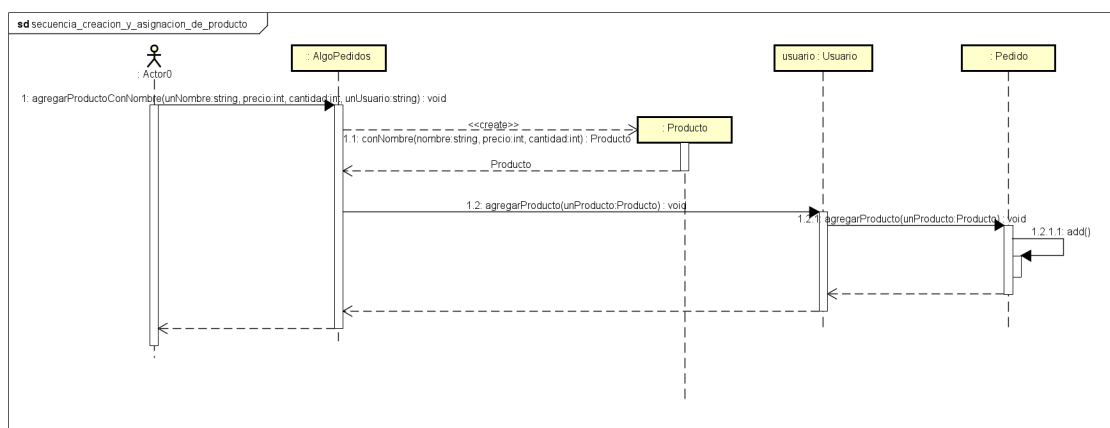


Figura 3: Secuencia de creacion y asignacion de producto

Nuevamente el actor es el que da inicio a esta secuencia , invocando al metodo `agregarProductoConNombre` , donde el objeto producto es instanciado en ese método y es pasado por parametro al usuario correspondiente.

Luego, el usuario delega a su pedido la agregacion del producto a la coleccion de productos.

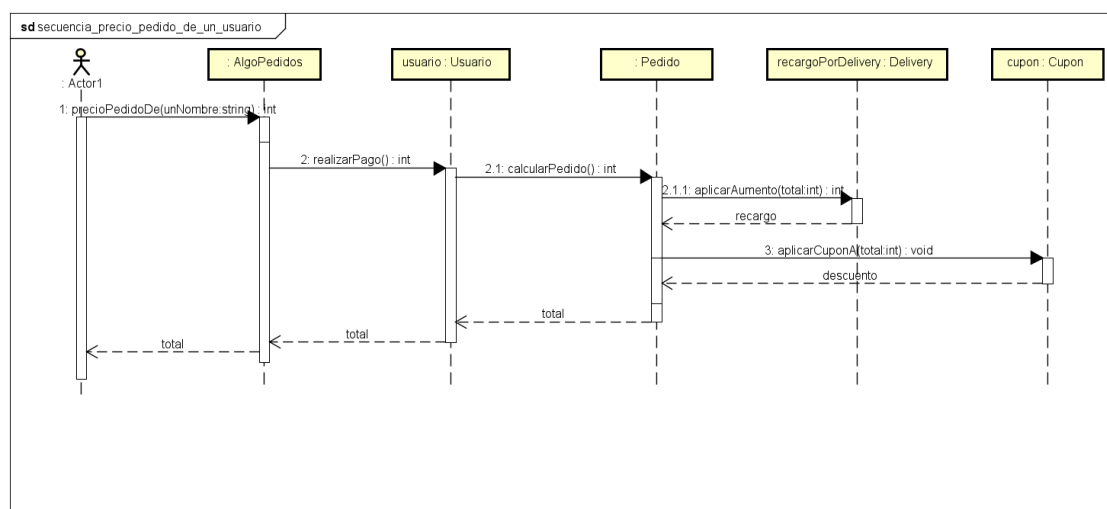


Figura 4: Secuencia de precio pedido de un usuario

Esta secuencia muestra como se realiza el recuento total del pedido a un determinado usuario de la Collection.

Comienza detectando al usuario para despues llamar al método que éste posee , `realizarPago`. Este metodo delega al pedido el calculo total a pagar y realizar los aumentos o descuentos necesarios delegando esas acciones a sus atributos Cupon y Delivery.