



# Orientación a Objetos 2 – Curso 2022

## Práctica 7

Fecha de última edición: 31 mayo 2022

### Ejercicio 1

Sea un lavarropas semiautomático que posee tres programas de lavado: lavado diario, ropa delicada y ropa muy sucia. Cada programa consiste de lo siguiente:

- Lavado diario: llena de agua hasta el 50%, agrega jabón en polvo, 20 minutos de lavado, cambia el agua, agrega enjuague, 7 minutos de enjuague, 3 minutos de centrifugado.
- Ropa delicada: llena de agua hasta el 100%, agrega jabón en polvo, 40 minutos de lavado, cambia el agua, agrega enjuague, 15 minutos de enjuague, no hay centrifugado
- Ropa muy sucia: llena de agua hasta el 100%, agrega jabón en polvo, 100 minutos de lavado, cambia el agua, agrega jabón en polvo, 30 minutos de lavado, cambia el agua, agrega enjuague, 15 minutos de enjuague, 10 minutos de centrifugado.

Cada programa al terminar su lavado, retorna la duración total del mismo en minutos.

Considere que el lavarropas posee los siguientes mensajes, los cuales puede utilizar sin implementar:

```
public class Lavarropas {  
  
    public void agregarAgua(int porcentaje) {...}  
    public void agregarJabonEnPolvo() {...}  
    public void lavar(int minutos) {...}  
    public void vaciarAgua() {...}  
    public void agregarEnjuague() {...}  
    public void enjuaga(int minutos) {...}  
    public void centrifugar(int minutos) {...}
```

Usted debe implementar los mensajes para poder iniciar el lavado utilizando uno de los programas mencionados.

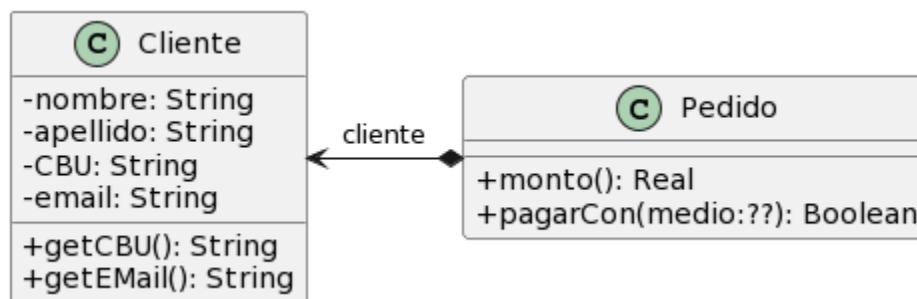
### Tareas:

1. Modele una solución para el problema planteado. Si utiliza algún patrón, indique cuál
2. Implemente en Java
3. Implemente un test para poder realizar el lavado con el programa para lavado diario.

## Ejercicio 2

En una cafetería se desea implementar el pago de los pedidos procesados mediante diferentes medios de pago (también conocidos como *gateways*). Por el momento, se requiere la posibilidad de realizar el pago por medio de débito automático y DineroMail. Sin embargo, no se descarta la posibilidad de utilizar otros medios en el futuro.

En la siguiente figura se muestra el modelo del sistema que incluye la clase Cliente, con variables de instancia para el nombre, apellido, mail y CBU (Clave Bancaria Uniforme) y sus correspondientes *accessors*. Y, por otro lado, la clase Pedido, que conoce al cliente que lo solicitó mediante la variable de instancia cliente, un mensaje `monto()` que calcula el costo del pedido y un mensaje `pagarCon(medio: ??)`. Este último retorna `true` o `false` dependiendo de si pudo o no realizar el pago.



Cada uno de los medios de pago dispone de una API con su correspondiente procedimiento de uso:

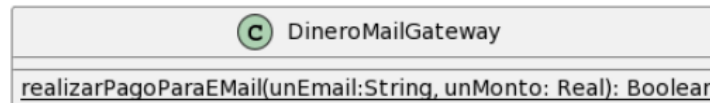
- Débito Automático: el pago requiere una autorización previa donde se valida si el cliente dispone de fondos. Para realizarlo Ud. dispone de una clase llamada **DebitoGateway** con dos mensajes de clase:
  - `autorizarMontoConCBU(unMonto: Real, cbu: String)`
  - `pagarMontoConCBU(unMonto: Real, cbu: String)`

El primero permite autorizar el monto de pago para un CBU retornando una colección de Strings que representan los errores de autorización. El segundo método, permite realizar el pago siempre y cuando no se hayan obtenido errores en la autorización anterior.

- DineroMail: el pago se realiza en un solo paso. Para realizarlo Ud. dispone de una clase llamada **DineroMailGateway** con un mensaje de clase:



- realizarPagoParaEmail (unEmail: String, unMonto: Real) que recibe como parámetro el e-mail del cliente y el monto de pago y retorna un boolean indicando si la operación se pudo realizar correctamente.



Tenga en cuenta que las API de los medios de pago no pueden ser modificadas ya que son utilizadas por varios sistemas.

## Tareas:

1. Diseñe una solución que permita soportar los diferentes medios de pago. Defina el comportamiento del mensaje que permite realizar el pago de un pedido. Si utiliza algún patrón de diseño, indique cuál mediante estereotipos. Realice el diagrama de clases UML correspondiente.
2. Implemente en Java su diseño.
3. Realice un test para mostrar cómo realizar el pago de un pedido por débito automático de un cliente que no dispone de fondos suficientes.

## Ejercicio 3

Sea un sitio web que tiene por objetivo conseguir financiación para proyectos a través de una mecánica muy sencilla. Las personas que tienen proyectos (emprendedores) publicitan los mismos y las personas que desean invertir indican en qué proyecto desean hacerlo y con qué dinero. Cuando un proyecto logra el dinero necesario se registra como conformado. Por su parte, el emprendedor puede cancelar su proyecto sólo si no está conformado. Cuando se cancela el proyecto no se puede aportar más dinero. En cambio, cuando se conforma el proyecto, es posible seguir aportando.

El sitio debe proveer la siguiente funcionalidad:

- Crear un proyecto con un título, un emprendedor responsable y un monto a alcanzar.
- Invertir un cierto monto en un proyecto dado.
- Consultar el monto total recibido en inversiones para un proyecto.
- Cancelar proyecto.



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## Tareas:

1. Diseñe una solución. Si utiliza algún patrón de diseño, indique cuál mediante estereotipos. Realice el diagrama de clases UML correspondiente.
2. Implemente en Java su diseño.
3. Realice un test para aportar dinero al proyecto “Vacaciones Kathmandu 2023” que necesita 5000 pesos y se aportan 500 pesos.