

# Taller Integrador: Patrones de diseño

## Sistema de transacciones de cuentas.

### Objetivos

- Evaluar que patrones de diseño se deben utilizar en el desarrollo de un sistema.
- Aplicar varios patrones de diseño dentro de un mismo sistema.

### Requerimientos del sistema:

Se desea desarrollar un sistema de cajero automático que permita manejar consultas de saldo, retiros y depósitos de cuentas bancarias, para esto se le provee una clase llamada Account, la cual es la única que no debe ser modificada, pero también se proveen 3 clases más, que pueden ser modificadas a conveniencia, y sobre las que se debe aplicar los patrones de diseño elegidos.

### Desarrollar

1. Indique para cada uno de los patrones estudiados si pudiera o no servir dentro del desarrollo de este sistema. (explique)
  - a. Creacionales.

Abstract factory	No hay creación de objetos por parte del sistema por tanto no es necesario este patron.
Factory Method	Lo mismo que abstract factory no hay creación sobre la marcha de ningún objeto.
Singleton	Dado que en la clase account su estructura no cambia y siempre se llama a una cada vez, entonces es una clase que usa singleton. Por otro lado, atmUK también es única y es la que tiene los manejadores de dinero, es así que también usa ese patrón.
Builder	No hay construcción compleja de ningún objeto y sus respectivas clases concretas.

- b. Estructurales.

Adapter	No se adapta ningún servicio o clase al sistema, el cual este predefinido y tenga poco margen para el cambio del mismo.
Composite	No se evidencia ninguna estructura que se pudiese almacenar en forma de árbol, así no es tan claro su uso en el actual sistema.

Facade	La clase VentanaPrincipal puede depender de pequeños subsistemas o servicios como retirar y depositar dinero, por tanto, si puede usarse.
Decorator	Puede ser que las cuentas se puedan adaptar para diferentes fines, pero para el sistema actual no se ve tan necesario hacerlo.

c. De Comportamiento.

Observer	No hay objetos esperando eventos en el sistema para tomar decisiones en base a los cambios que obtenga una clase principal, es así que no es tan evidente usarlo.
Chain of responsibility	Puede ser una opción ya que los manejadores de dinero pueden ir probándose hasta encontrar la solución de receptor esa cantidad de dinero.
Iterator	Se recorren las cuentas en una para encontrar una con un id y es así que si puede ser usado iterador.
Strategy	Si el sistema fuese más complejo y hubiese distintas clases de tarjetas puede que sea necesario utilizar diferentes algoritmos, sin embargo, en el sistema que se propone no es tan claro usarlo.
Memento	No es necesario guardar estados anteriores en el formato actual, sin embargo, si hubiese una base de datos puede que para cuestiones de backup si fuese necesario usar memento.

2. Diseñe un diagrama de clases del sistema, aplicando los patrones elegidos.

Se adjuntará en el repositorio por cuestiones de calidad de imagen