Taller Integrador: Patrones de diseño

# Sistema de Requerimientos Académicos.

# 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)
   1. Creacionales.
      1. Factory Method

No sirve dentro del desarrollo del sistema porque no existen diferentes formas de realizar las acciones que se requieren en el sistema: retiros, saldos y depósitos

* + 1. Abstract Factory

No sirve dentro del desarrollo del sistema porque no existen familias de los tipos de cuentas bancarias

* + 1. Singleton

Si sirve dentro del desarrollo del sistema porque este permite inicializar el único cajero automático que se encuentra en el main

* + 1. Builder

No sirve dentro del desarrollo del sistema porque el mismo no requiere de un sistema complejo que se pueda dividir en partes

* 1. Estructurales.
     1. Adapter

No sirve dentro del desarrollo del sistema porque no se están adaptando interfaces dentro del sistema

* + 1. Facade

Si sirve dentro del desarrollo del sistema porque se va a desarrollar un menú de tal forma que el cliente no tenga que conocer todas las implementaciones sino que solo selecciona la opción que quiere y se realiza la consulta correspondiente

* + 1. Composite

No sirve dentro del desarrollo del sistema porque el sistema no se va a comportar como estructura de árbol

* + 1. Decorator

No sirve dentro del desarrollo del sistema porque el sistema no adquiere responsabilidades dinamícamente.

* 1. De Comportamiento.
     1. Chain of Responsability

Si sirve dentro del desarrollo del sistema porque este se implementa en lo que son las diferentes opciones de transacción: deposito, retiro y balance

* + 1. Iterator

No sirve dentro del desarrollo del sistema porque el sistema no se necesita recorrer objetos de manera secuencial

* + 1. Memento

No sirve dentro del desarrollo del sistema porque no se requiere guardar ningún estado anterior

* + 1. Strategy

Si sirve dentro del desarrollo del sistema porque tenemos diferentes números de denominación lo cual influye en la implementación de cada función bancaria

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