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.

- **Factory Method:** Debido a que el sistema que se va a implementar no es muy complejo, consideramos que este patrón de diseño es el que más se adecúa para el desarrollo del mismo.

-  **Abstract Factory:** Ya que no es necesario crear familias de objetos relacionados o dependientes entre sí, sin especificar sus clases concretas, no consideramos que este patrón sea el indicado para ser aplicado.

**- Singleton:** Este patrón de diseño es generalmente útil cuando necesitemos que exista sólo un objeto de una determinada Clase.

**- Builder:** Debido a que no necesitamos crear objetos complejos compuestos de varias partes independientes, no necesitamos este patrón de diseño.

* 1. Estructurales.
* **Adapter**: Nos da la flexibilidad para agregar nuevas funcionalidades al programa sin tener que modificar las clases o interfaces previamente definidas. Por ejemplo, si queremos que el cajero sea más robusto, se le puede implementar el pago de servicios básicos sin que tengamos que cambiar el diseño original.
  1. De Comportamiento.
* **Chain of responsibility:** Delega la responsabilidad del retiro y la inserción de los distintos billetes y monedas a distintos handlers que heredan de la clase abstracta Handler

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

<https://www.lucidchart.com/invitations/accept/c9ac7b46-dd13-4310-9107-3c6ea43971f0>