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.

|  |  |
| --- | --- |
| Abstract Factory | No necesitamos crear familias de objetos dependientes. |
| Singleton | Sí, dado a que solamente necesitamos una instancia de la clase ATMUK, ya que este objeto funcionará únicamente en un cajero, y no importa que se puedan instanciar más de un objeto de este mismo tipo. |
| Factory Method | No necesitamos definir diferentes objetos para cada clase |

* 1. Estructurales.

|  |  |
| --- | --- |
| Adapter | No necesitamos que trabajen juntas clases con interfaces incompatibles. |
| Composite | No necesitamos que un objeto contenga a otro u otros |
| Decorator | Sí, dado a que existe la posibilidad de que nos encontremos con cuentas de diferentes características por ejemplo, ya sean estas de tipo empresarial(coorporativa) o de usuario y por ende es recomendable agregar estas funciones con ayuda de este patrón. |

* 1. De Comportamiento.

|  |  |
| --- | --- |
| Chain of Responsability | Sí, ya que tenemos métodos en los cuales podemos implementar varios manejadores, como en el de sacar dinero y depositar dinero, en los cuales podemos tener manejadores de los diferentes tipos de dinero |
| Iterator | No necesitamos iterar elementos |
| Memento | No necesitamos regresar a un punto anterior |
| Strategy | No necesitamos que un objeto tenga varias estrategias para la misma resolución sino varios métodos |

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

