# Descripción del Patrón de diseño

## Definición

El patrón de diseño strategy permite establecer en tiempo de ejecución el rol de comportamiento de una clase. Stretagy se basa en el polimorfismo para implementar una serie de comportamientos que podrán ser intercambiados durante la ejecución del programa, logrando con esto que un objeto se pueda comportar de forma distinta según la estrategia establecida.

## Propósito

Define un conjunto de algoritmos, encapsula cada uno de ellos y los hace intercambiables. Permite que un algoritmo varié independientemente de los clientes que lo usan.

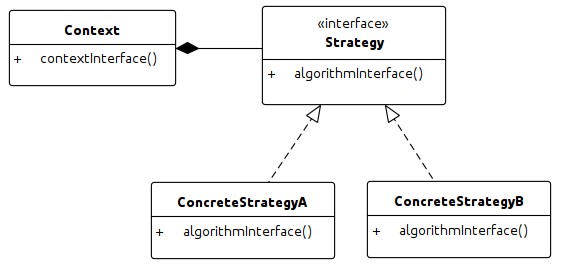
## Aplicabilidad

El patrón Estrategia se usa cuando:

* Varias clases difieren en el comportamiento de una o dos responsabilidades.
* Es necesario implementar diversas variantes de un mismo algoritmo y seleccionar una variante concreta en tiempo de ejecución.
* Una clase define muchos comportamientos y eso se traduce en un montón de condicionales en sus métodos.

## Diagrama genérico

**Modelo UML**



**Participantes:**

Las clases y objetos que participan en este modelo son:

### Strategy

* Declara una interfaz común para todos los algoritmos compatibles.
* El **Contex** utiliza esta interfaz para llamar al algoritmo definido por un **ConcreteStrategy.**

**ConcreteStrategy**

* Implementa el algoritmo utilizando la interfaz de **Strategy**.

### Context

* Instancia de un objeto **ConcreteStrategy.**
* Mantiene una referencia a un objeto de **Strategy**
* Puede definir una interfaz que le permite acceder a sus datos **Strategy**.

## Consecuencias

El uso del patrón Strategy tiene las siguientes ventajas y desventajas:

* Factoriza aspectos comunes de una familia de algoritmos y utilizarlos en las clases base de la jerarquía.
* Aumenta cohesión del cliente.
* Sistematiza el uso de implementaciones alternativas.
* El cliente es el responsable de crear estrategias, por tanto, debe comprender las posibilidades que ofrecen, esto es, debe ser relevante para el contexto del cliente.
* Menor eficiencia, Aumenta el número de objetos creados.

# Descripción del problema

Mediante la implementación del patrón de diseño Strategy desarrollar una aplicación que permita realizar operaciones aritméticas básicas (Sumar, Restar, Multiplicar y Dividir).

El usuario podrá realizar cualquier operación básica como en cualquier calculadora.

# Solución

## Diagrama de clases

