

Este Patrón Estrategia se utilizar:

1. Cuando varias clases relacionadas solo difieren en su comportamiento.
2. Cuando se necesitan variantes del mismo algoritmo que se implementan como

una jerarquía de clases.

1. Cuando en situaciones un algoritmo usa datos que un cliente no tienen porqué conocer.
2. O cuando una clase define muchos comportamientos que aparecen en sentencias condicionales.

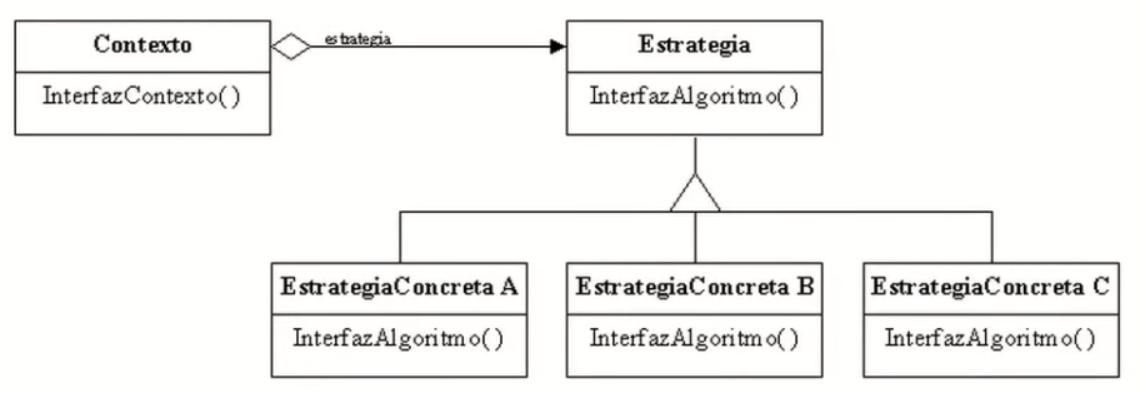
Estructura General del Patrón

CLASE CONTEXTO: El contexto (o cliente) decide la estrategia que vamos a crear. Contexto es la clase responsable de crear y mantener una referencia a una estrategia concreta.

CLASE ESTRATEGIA: Esta clase Estrategia que declara (valga la redundancia), una interfaz

común para todas las variantes de un algoritmo.

ESTRATEGIAS CONCRETAS: Estas clases son aquellas que implementan una variante del algoritmo.



CONSECUENCIA.

Este patrón nos ayuda en el momento de implementarlo, a factorizar la funcionalidad común en los algoritmos, aparte de que nos ayuda a eliminar las sentencias condicionales.

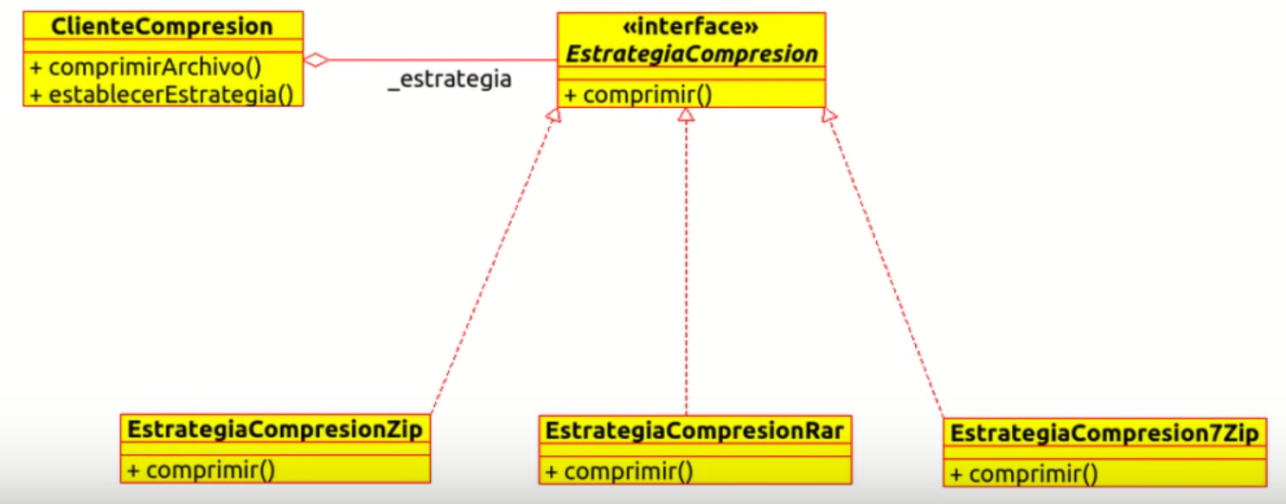
Notas: “Cada patrón es una regla de tres partes, la cual expresa una relación entre un cierto contexto, un problema y una solución”.

1. La clase cliente o clase contexto es la responsable de crear las estrategias, por lo tanto debe de comprender las posibilidades que cada una de ellas ofrece.
2. Definir una familia de algoritmos intercambiables y permite variar su comportamiento sin importar el cliente que los use.
3. Muy parecido al Patrón de Diseño ***Estado***, pero el Patrón de diseño ***Strategy*** encapsula un algoritmos o comportamientos y no el estado de un contexto.

Ejemplo No.1 :

Crear un diseño en las que una aplicación puede comprimir en diferentes Formatos de Archivos, se sabe que hay muchas formas de hacerlo pero debemos escoger alguna estrategia en tiempo de ejecución, sin embargo, no podemos tener todos estos algoritmos discriminados o mezclados por una sentencia condicional, por lo que hemos separado los formatos de compresión en diferentes estrategias.

***Estructura concreta :***



Se observa la similitud entre la Estructura general y la estructura particular de la situación en cuestión

La CLASE CONTEXTO es similar a la CLASE CLIENTECOMPRESION.

Esta clase CLIENTECOMPRESION tendrá una referencia a una Estrategia Concreta, que

en este caso están representadas por las clases EstrategiaCompresiónZip, EstrategiaCompresionRar y EstrategiaCompresion7Zip. Todas ellas implementan la interface ESTRATEGIACOMPRESION y por consiguiente su método ***comprimir( ).***

La CLASE ESTRATEGIA es similar a la CLASE ESTRATEGIACOMPRESION <<interfaz>>

Así mismo, la EstrategiaConcreta# es similar a las Clases ESTRATEGIACOMPRESION#.

***FUNCIONAMIENTO:***

Supongamos ahora que se desea comprimir un archivo en 7-ZiP, en dicho caso el método comprimirArchivo( ), llamará al método establecerEstrategia( ) y que lo que hará es establecer una referencia a la EstrategiaCompresión7Zip, luego de esto se ejecutará el método comprimir( ), que comprimirá el archivo en el formato deseado. Se observa que el el software en tiempo de ejecución se le permite al usuario elegir, cuál es el mejor método de compresión.

//----------------------------------------------------------------------------------------------------------------------

***Ejercicios***

Ejercicio No.1

En el mundo del juego del ajedrez, no todos los jugadores buscan una partida de ataque desde las primeras jugadas. Así que, para evitar las complicaciones desde el principio del juego, la apertura (son aproximadamente las primeras 7 o 8 primeras jugadas de cada contendiente) puede ser planteada como una buena estrategia de defensa. Es decir que, en principio, un juego de ajedrez la apertura debe prometer llegar simplemente a una posición sana y segura. Para ello existen algunas defensas conocidas, que suelen evitar rápidos ataques por parte del oponente. Estas defensas han sido establecidas como estrategias de aperturas aceptadas. Los nombres de algunas de ellas son los siguientes: Defensa Caro-Kann, Petrov, Berlinesa, India de Dama, Muro de Piedra, entre otras. En este ejercicio se deberá implementará el patrón Estrategia que ayudará al contexto a determinar un determinado tipo de defensa.

En el siguiente diseño se plantean solo tres tipos de defensa.

