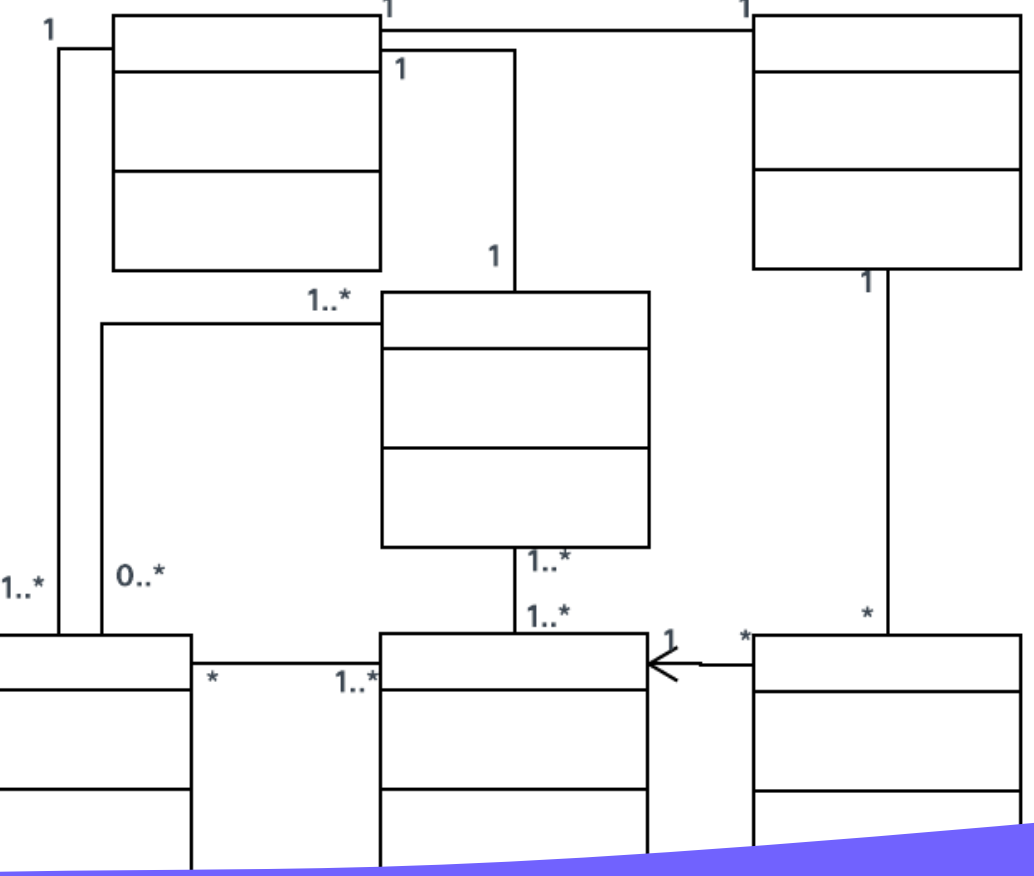
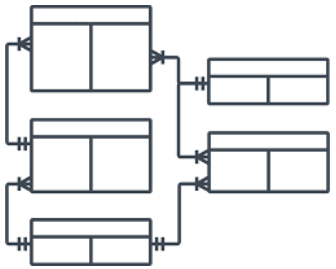


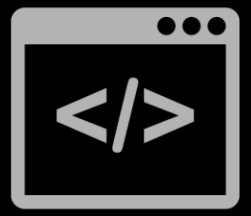


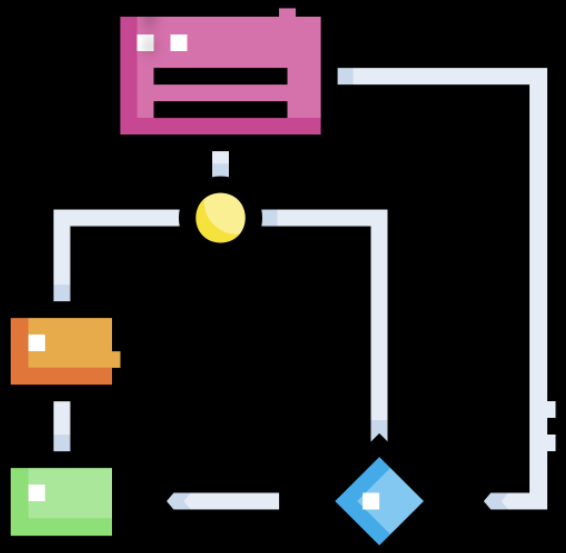
Patrones de Diseño

Agenda

- Patrón Strategy
- Patrón Adapter

Patrón State





Patrón State

Es un patrón de Comportamiento

¿Qué hace?

- Genera una abstracción (una clase) por cada posible estado que pueda tener un objeto.
- Define transiciones entre los posibles estados.

Patrón State

Se sugiere su utilización cuando:

- El comportamiento de un objeto depende de su estado y este mismo puede variar en tiempo de ejecución.

- Un método está lleno de sentencias condicionales que dependen del estado del objeto. Estos estados suelen estar representados por varios atributos de distintos tipos: primitivos (boolean, int, etc.) o por enumerados (enum).

Patrón State

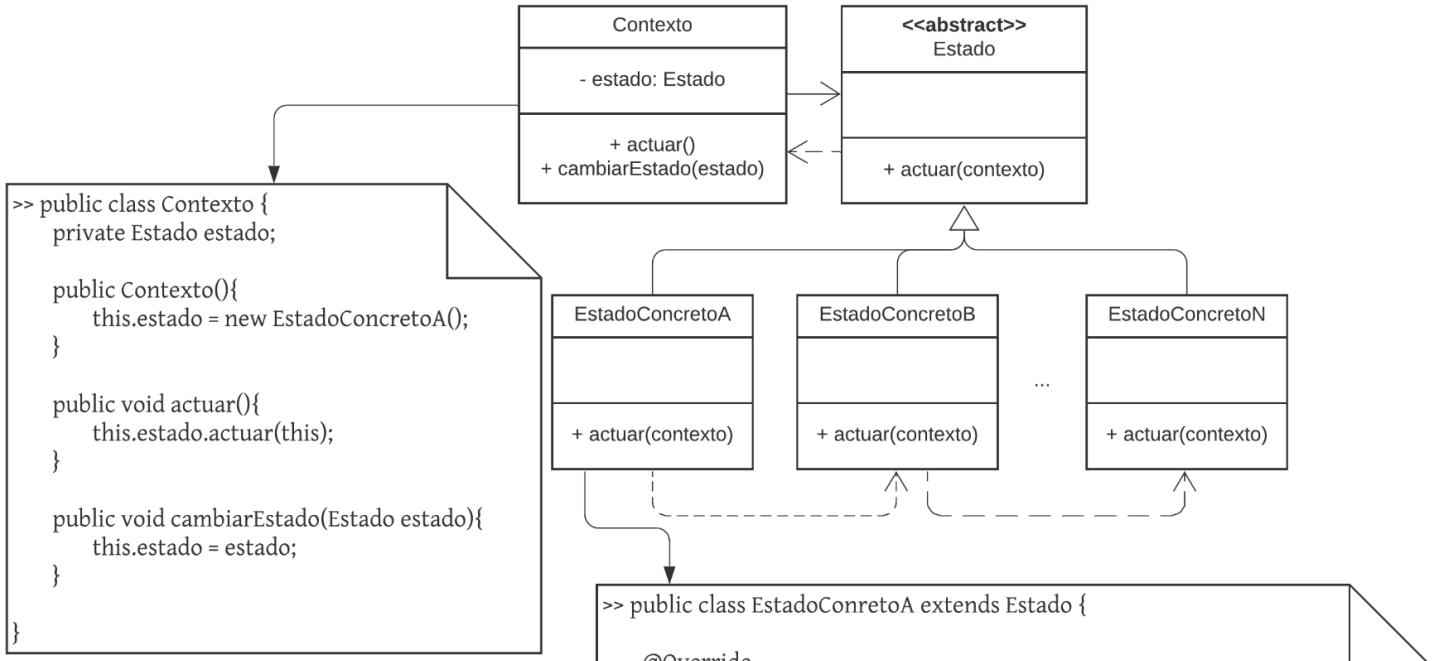
Componentes:

- ***Interface (o clase abstracta) State***: Define las firmas de los métodos que dependen del estado del objeto principal.

- ***Clases de estados concretas***: Clases que implementan la interface State (o que heredan de ella, si ésta fuera clase abstracta), es decir, que tienen la implementación real de los métodos. Son los estados posibles del objeto principal.

- ***Contexto***: Clase que tiene referencia a la interface/clase abstracta State, cuyos objetos van a delegar la responsabilidad de resolución de algunos problemas en el estado. Estos objetos utilizarán de forma polimórfica a los estados.

Patrón State




```
>> public class Contexto {
    private Estado estado;

    public Contexto(){
        this.estado = new EstadoConcretoA();
    }

    public void actuar(){
        this.estado.actuar(this);
    }

    public void cambiarEstado(Estado estado){
        this.estado = estado;
    }

}
```

```
>> public class EstadoConretoA extends Estado {

    @Override
    public void actuar(Contexto contexto){
        //hacer algo
        if(this.cumpleCondicionParaCambiarDeEstado(contexto)){
            contexto.cambiarEstado(new EstadoConcretoB());
        }
    }

    private boolean cumpleCondicionParaCambiarDeEstado(Contexto contexto){
        //verificar si se cumple la condición y devolver el valor
    }
}
```

Patrón Template Method

Es un patrón de Comportamiento

¿Qué hace?

- Define el esqueleto de un algoritmo, estableciendo los pasos que sí o sí se deben implementar.

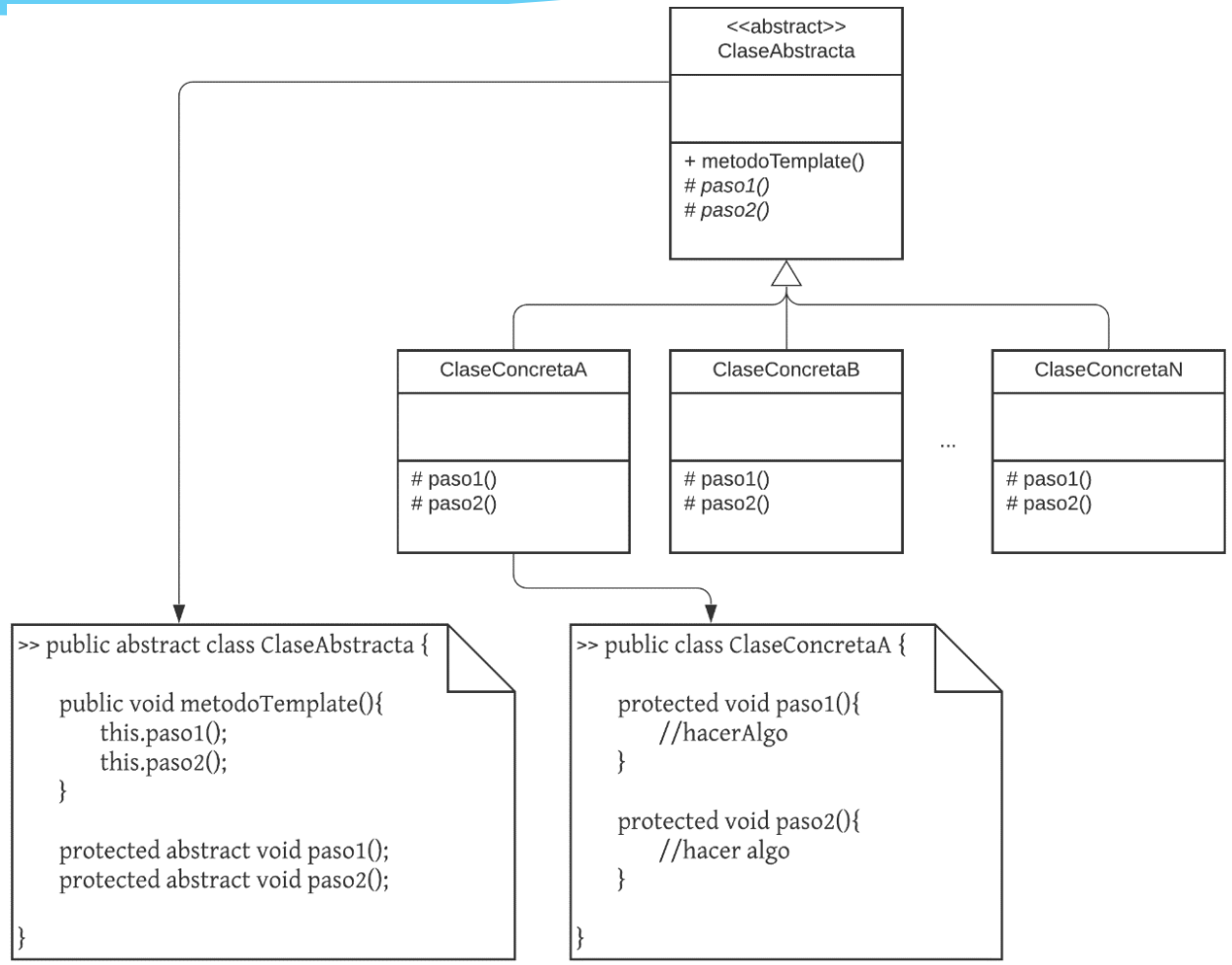
Patrón Template Method

Se sugiere su utilización cuando:

- Varias abstracciones tienen los mismos pasos y orden para realizar una determinada acción, pero cada una de ellas lo implementa de forma diferente.

- Se requiere utilizar de forma polimórfica dos o más objetos que pueden ejecutar el mismo algoritmo, respetando sus pasos, pero con implementaciones distintas para cada uno de éstos.

Patrón Template Method



Gracias