

Programación web con PHP



Patrones: State

Temario

Tipo.....	4
Propósito.....	4
Utilización.....	4
Ventajas.....	4
Desventajas.....	4
Estructura.....	5
¿Cómo funciona?.....	5
Código.....	6
Contexto.....	6
Interfaz Estado Abstracto.....	7
Estado Concreto A.....	7
Clase Estado Concreto B.....	7
index.php.....	8

Tipo

De comportamiento (Nivel objetos)

Propósito

Permite que un objeto cambie de comportamiento de acuerdo a su estado interno

Utilización

- Cuando el comportamiento de un objeto dependa de su estado y este puede variar dinámicamente durante la ejecución
- Cuando las operaciones tienen definiciones largas y con condiciones múltiples dependiendo del estado del objeto

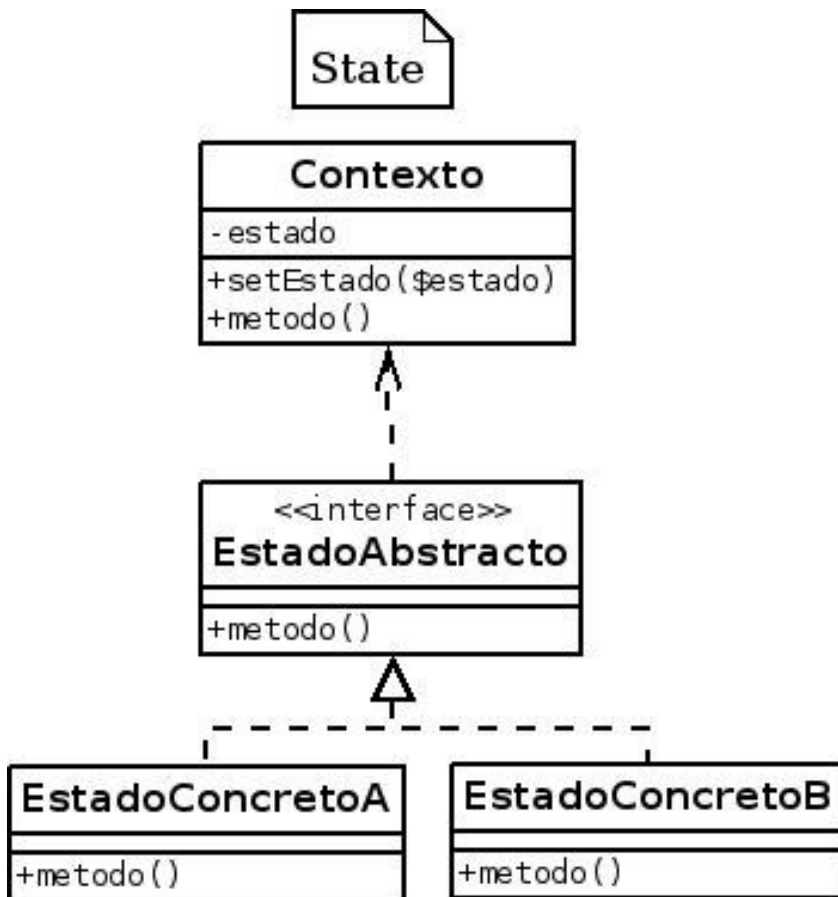
Ventajas

- Flexibilidad para agregar nuevos estados y transiciones entre estados
- Evita definiciones con muchas condiciones
- Hace explícitas las transiciones y mas legibles al separarlas en clases diferentes
- Si los estados no tienen atributos propios y todos estan definidos en el Estado Abstracto se puede compartir con diferentes contextos

Desventajas

- Aumenta el numero de clases para cada estado nuevo

Estructura



¿Cómo funciona?

- Definimos el Contexto que tendrá un atributo estado y un método para cambiar el estado que instanciara el estado requerido y lo guardará en el atributo. También tendrá por cada método del Estado Abstracto un método que devolverá la ejecución de ese método para el estado guardado como atributo.
- Definimos la interfaz Estado Abstracto que contendrá las definiciones de los métodos de los Estados Concretos.
- Definimos tantos Estados Concretos como sea necesario que implementes el Estado abstracto e implementen los métodos.
- En nuestro programa instanciamos un contexto, luego le otorgamos un estado utilizando la función `setEstado` y pasando una instancia del Estado Concreto que necesitemos. También es posible setear un estado en el constructor del Contexto.
- Cuando le pedimos al Contexto que realice un método este utiliza el estado que tiene

almacenado y ejecuta su definición del método y luego devuelve este resultado. De esta manera si el estado guardado es EstadoConcretoA al llamar al método de contexto se ejecutará el método de EstadoConcretoA y si el estado guardado es EstadoConcretoB se ejecutará el método de EstadoConcretoB.

- También es posible crear métodos en el Contexto que generen transiciones de un estado a otro

Código

Contexto

```
<?php
//Contexto
class Contexto{
    private $estado;

    function __construct()
    {
        echo "Creando Contexto...<br>";
        //Inicializo en un estado
        $this->estado=new EstadoConcretoA();
    }

    public function setEstado($estado){
        $this->estado=$estado;
    }

    public function metodo(){
        echo "Metodo...<br>";
        //Llamo al metodo del estado guardado
        $this->estado->metodo();
    }
}
```

Interfaz Estado Abstracto

```
<?php
//Interfaz Estado Abstracto
interface EstadoAbstracto{
    public function metodo();
}
```

Estado Concreto A

```
<?php
//Estado Concreto A
class EstadoConcretoA implements EstadoAbstracto{
    function __construct()
    {
        echo "Estado: Estado Concreto A...<br>";
    }
    public function metodo(){
        echo "Metodo desde Estado Concreto A...<br>";
    }
}
```

Clase Estado Concreto B

```
<?php
//Estado Concreto B
class EstadoConcretoB implements EstadoAbstracto{
    function __construct()
    {
        echo "Estado: Estado Concreto B...<br>";
    }
    public function metodo(){
        echo "Metodo desde Estado Concreto B...<br>";
    }
}
```

index.php

```
<?php
//Creo contexto con un estado inicial
$contexto=new Contexto();//Creando Contexto...Estado: Estado Concreto
A...
//Llamo al método
$contexto>metodo();//Metodo...Metodo desde Estado Concreto A...
//Cambio el estado a Estado Concreto B
$contexto>setEstado(new EstadoConcretoB);//Estado: Estado Concreto
B...
//Llamo al método
$contexto>metodo();//Metodo...Metodo desde Estado Concreto B...
//Cambio el estado a Estado Concreto A
$contexto>setEstado(new EstadoConcretoA);//Estado: Estado Concreto
A...
//Llamo al método
$contexto>metodo();//Metodo...Metodo desde Estado Concreto A...
```