Builder Design Pattern

Ivan Buendia

Creacional

Hablando de patrones de diseño creacionales, parece bastante semántico tener un patrón diseño Builder.

El patrón Builder nos ayuda a construir objetos complejos sin directamente instanciando su estructura, o escribiendo la lógica que requieren.

Problema:

Cuando la creación de un objeto es compleja los parámetros que puede necesitar el constructor pueden ser largos y ademas pueden cambiar con el tiempo.

Telescoping Constructor anti-pattern

```
public class Pizza {
  public Pizza(int size) { ... }
  public Pizza(int size, boolean cheese) { ... }
  public Pizza(int size, boolean cheese, boolean pepperoni) { ... }
  public Pizza(int size, boolean cheese, boolean pepperoni, boolean bacon) { ... }
  ...

}
```

Solucion:

Crear distintas variedades del objeto sin tener que contar con una larga lista de parámetros en el constructos.

Un patrón de diseño de Builder intenta:

- Creaciones complejas abstractas para que la creación de objetos esté separada del objeto usuario
- Cree un objeto paso a paso llenando sus campos y creando el incrustado objetos
- Reutilizar el algoritmo de creación de objetos entre muchos objetos.

Usar cuando:

La construcción del objeto requiere una gran cantidad de parámetros.

De esta manera el constructor no estaría contaminada.

Implementacion

```
class Frog {
  constructor(name, weight, height, gender) {
    this.name = name;
    this.weight = weight;
    this.height = height;
    this.gender = gender;
  }
}
```

```
class FrogBuilder {
  constructor(name, gender) {
    this.name = name;
    this.gender = gender;
  }
  setWeight(weight): FrogBuilder {
    this.weight = weight;
    return this;
  }
  setHeight(height): FrogBuilder {
    this.height = height;
    return this;
  }
  build(): Frog {
    if (!('weight' in this)) {
      throw new Error('Weight is missing');
    }
    if (!('height' in this)) {
      throw new Error('Height is missing');
    }
    return new Frog(this.name, this.weight, this.height, this.gender);
  }
}
```

Ejecución

```
const frog = new FrogBuilder('Leon', 'male').setWeight(14).setHeight(3.7).build();
```