

Patrón Singleton

Patrón Creacional (POO)

Fuente principal:

<https://refactoring.guru/es/design-patterns/singleton>

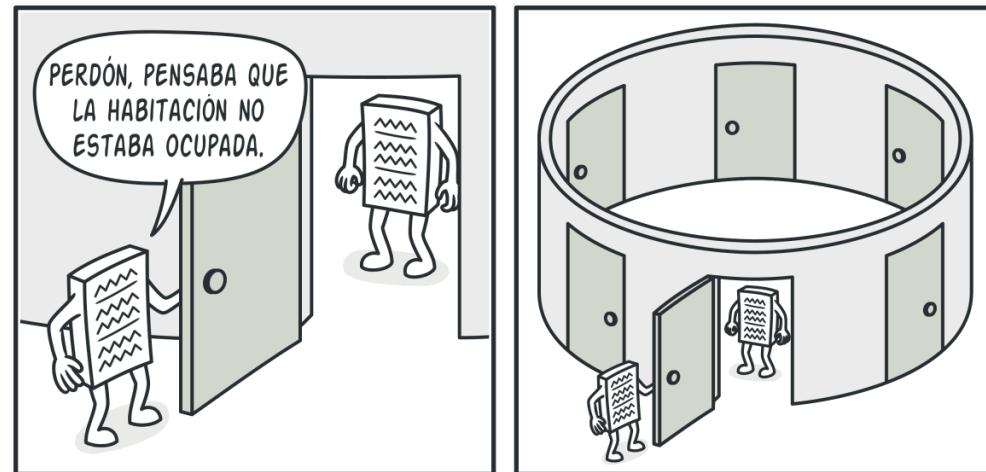
Propósito

Singleton es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia.

Problema

El patrón Singleton resuelve dos problemas al mismo tiempo, vulnerando el Principio de responsabilidad única:

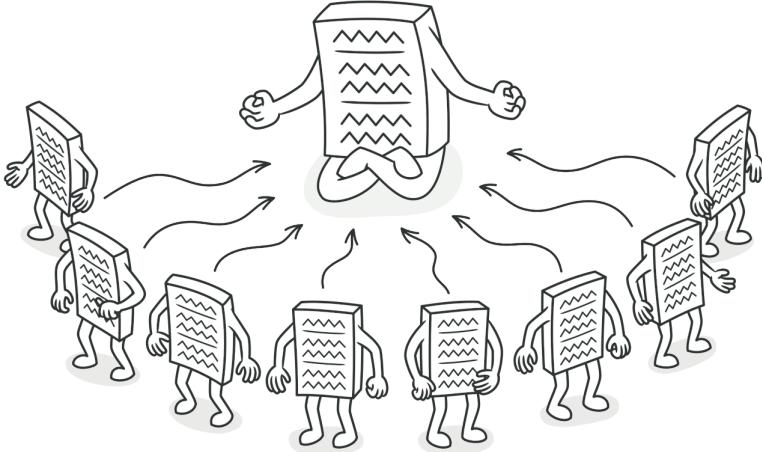
- 1. Garantizar que una clase tenga una única instancia.**
 - Motivo: controlar el acceso a algún recurso compartido (ej. base de datos).
 - Comportamiento imposible con un constructor normal.
- 2. Proporcionar un punto de acceso global a dicha instancia.**
 - Similar a variables globales pero protegiendo la instancia de sobreescritura.



Solución

Todas las implementaciones del patrón Singleton tienen dos pasos en común:

- Hacer **privado** el constructor por defecto para evitar que otros objetos utilicen el operador `new`.
- Crear un método de creación **estático** que actúe como constructor.
 - Invoca al constructor privado.
 - Guarda el objeto en un campo estático.
 - Devuelve el objeto almacenado en caché.



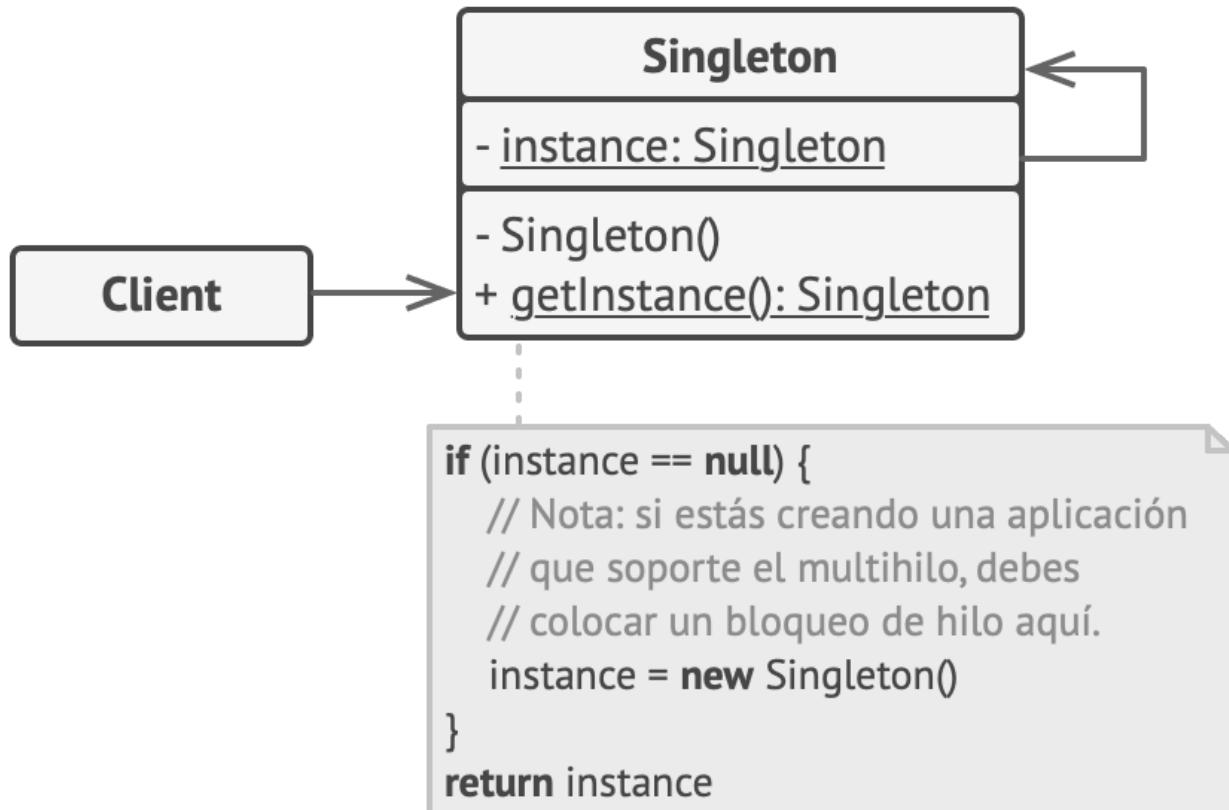
Analogía en el mundo real

El gobierno es un ejemplo excelente del patrón Singleton.

- Un país solo puede tener un gobierno oficial.
- El título "Gobierno de X" es un punto de acceso global que identifica al grupo de personas a cargo.

Estructura

1. La clase Singleton declara el método estático `obtenerInstancia` que devuelve la misma instancia de su propia clase.
2. El constructor del Singleton debe ocultarse del código cliente.



Código Java

```
package es.uva.poo.singleton;

public class Singleton {

    // Variable estática privada para la única instancia
    private static Singleton instancia;

    // Constructor privado
    private Singleton() {
        System.out.println("Inicializando instancia única...");
    }

    // Método estático para obtener la instancia
    public static Singleton obtenerInstancia() {
        if (instancia == null) {
            instancia = new Singleton();
        }
        return instancia;
    }

    public void mostrarMensaje() {
        System.out.println("Soy la única instancia.");
    }
}
```