

# Patrón Singleton

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**

The Ultimate Degree



**Singleton** es un **patrón de diseño de creación** que garantiza que una clase tenga una sola instancia y define un punto de acceso global para ella.

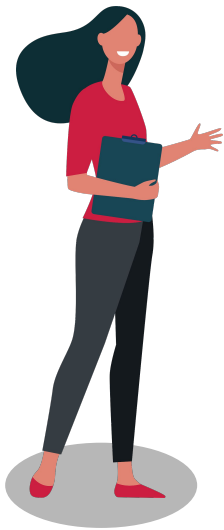


# Patrón Singleton

- En el patrón Singleton, una clase administra su propia instancia y evita que cualquier otra clase cree una instancia de ella.
- Para crear la instancia usando el patrón Singleton, debe pasar obligatoriamente por la clase, ninguna otra clase puede instanciarla.
- El patrón Singleton también proporciona un punto de acceso global a su instancia.
- La clase en sí siempre ofrecerá su propia instancia y, si aún no tiene una, crea y devuelve esta instancia recién creada.

# Crear una clase con el Patrón Singleton

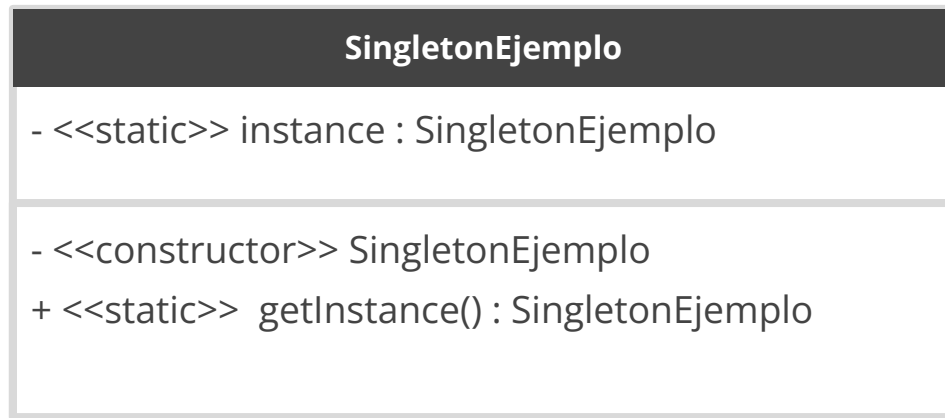
Para crear una clase con el patrón Singleton es necesario realizar los siguientes pasos:



- Crear un **atributo estático** del mismo tipo que la clase con el nombre de instancia.
- Todos los constructores de la *clase* deben usar el modificador **private**.
- Crear un método **getInstance ()** estático que devuelva el atributo de instancia.

# Ejemplo

## UML

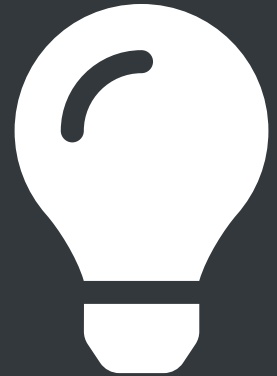


```
public class SingletonEjemplo {  
  
    //Atributo con mismo nombre de la clase  
    private static SingletonEjemplo instance = new  
SingletonEjemplo();  
  
    //Constructores privados  
    private SingletonEjemplo(){  
        }  
  
    //Método getInstance() estático  
    public static SingletonEjemplo getInstance(){  
  
        return instance;  
  
    }  
  
}
```

Código de la clase  
**SingletonEjemplo**



El patrón **Singleton** se usa cuando necesita un solo punto para crear una instancia de clase y cuando solo necesita una instancia de una clase.



DigitalHouse>  
Coding School