



SINGLETON

PATRONES DE DISEÑO **TIPO CREACIONAL.**

Integrantes
Gardiola Joaquin
Giachero Ezequiel

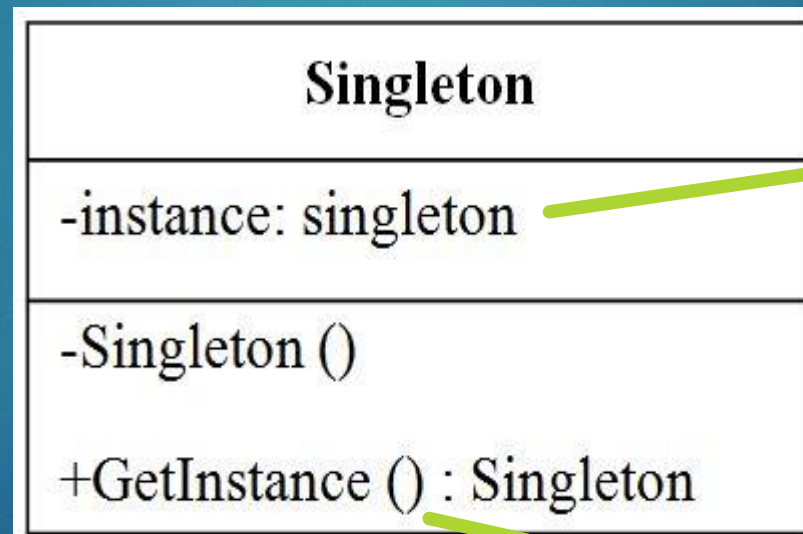
Propósito y Usabilidad

- ▶ Asegurar que una clase tiene una sola instancia y proporcionar un punto de acceso global a ella.
- ▶ Este método debe usarse cuando:
 - Cuando hay exactamente una instancia de una clase, y ésta debe ser accesible a los clientes desde un punto de acceso conocido.
 - Cuando la única instancia debería ser extensible mediante herencia, y los clientes deberían ser capaces de usar una instancia extendida sin modificar su código.

Estructura

- ▶ El constructor de la clase debe ser PRIVADO.
- ▶ Se proporciona un método ESTATICO en la clase que devuelve la UNICA INSTANCIA de la clase: GetInstance().

- Indica que es **private**
+ indica que es **public**



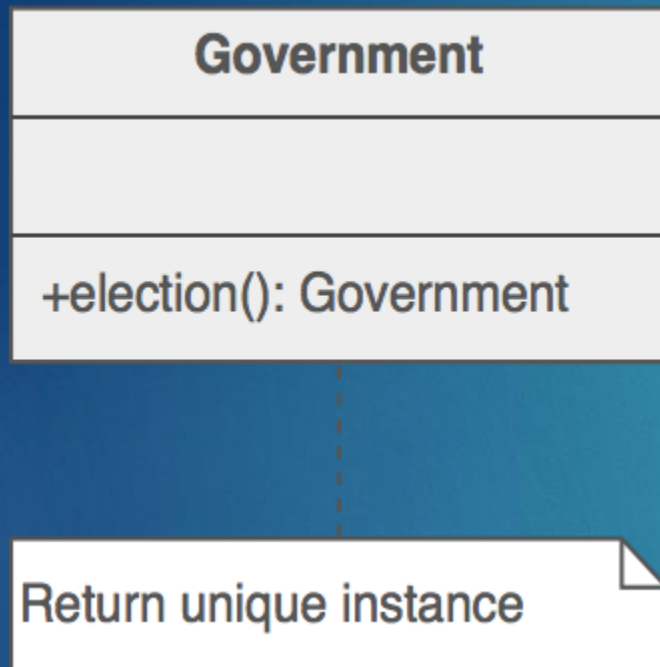
Atributos

Métodos

Devuelve una única instancia

Método estático

Ejemplo



La oficina del Presidente de Estados Unidos es un Singleton. La Constitución de norteamericana define los medios en los que un Presidente puede ser electo y sus limitaciones. Como resultado solo un Presidente puede estar activo al mismo tiempo. Independientemente de la persona que ocupe el cargo, la function `election()` deberia ser el acceso global al jefe del Estado.

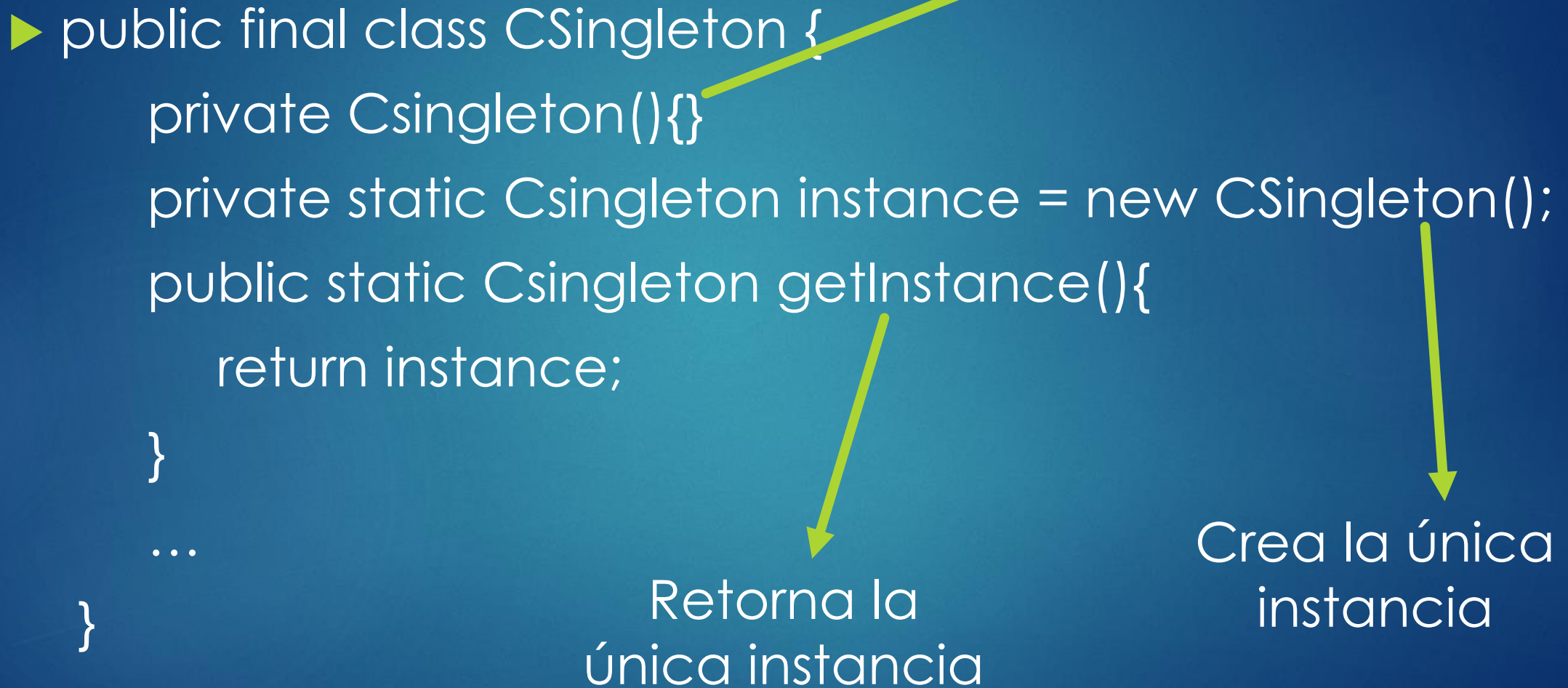
Esquema

Constructor

```
► public final class CSingleton {  
    private Csingleton(){}  
    private static Csingleton instance = new CSingleton();  
    public static Csingleton getInstance(){  
        return instance;  
    }  
    ...  
}
```

Retorna la
única instancia

Crea la única
instancia

A diagram illustrating the Singleton pattern implementation. The code is shown on the left, and three annotations with arrows point to specific parts of it. The word 'Constructor' is at the top right, with an arrow pointing to the private constructor 'private Csingleton(){}'. The text 'Retorna la única instancia' is at the bottom center, with an arrow pointing to the 'return instance;' line in the 'getInstance()' method. The text 'Crea la única instancia' is at the bottom right, with an arrow pointing to the 'new CSingleton()' line in the static initialization of 'instance'.

Consecuencias

- ▶ Acceso controlado a la única instancia: Encapsula su única instancia, puede tener un control estricto sobre como y cuando acceden a ella los clientes.
- ▶ Permite el refinamiento de las operaciones y la representación: Se puede crear una subclase de la clase Singleton, y es fácil configurar una aplicación con una instancia de esta clase extendida.
- ▶ Permite un número variable de instancias: Hace que sea fácil permitir mas de una instancia de la clase. Solo se necesitaría cambiar la operación que otorga acceso a la instancia del Singleton