

# PATRONES DE DISEÑO: SINGLETON

---

Ingeniería de Software II

Luis Gonzalez



# BREVE INTRODUCCIÓN A PATRÓN DE DISEÑO

- Forma reutilizable de resolver un problema común.
  - Ahorran tiempo
  - Seguridad en la validez del código
  - Establecen un lenguaje común
- 
- Tenemos distintos grupos y tipos: Grupo de los Creacionales, estructurales y de comportamiento.
  - **Singleton** pertenece al grupo de los patrones creacionales.

# SINGLETON

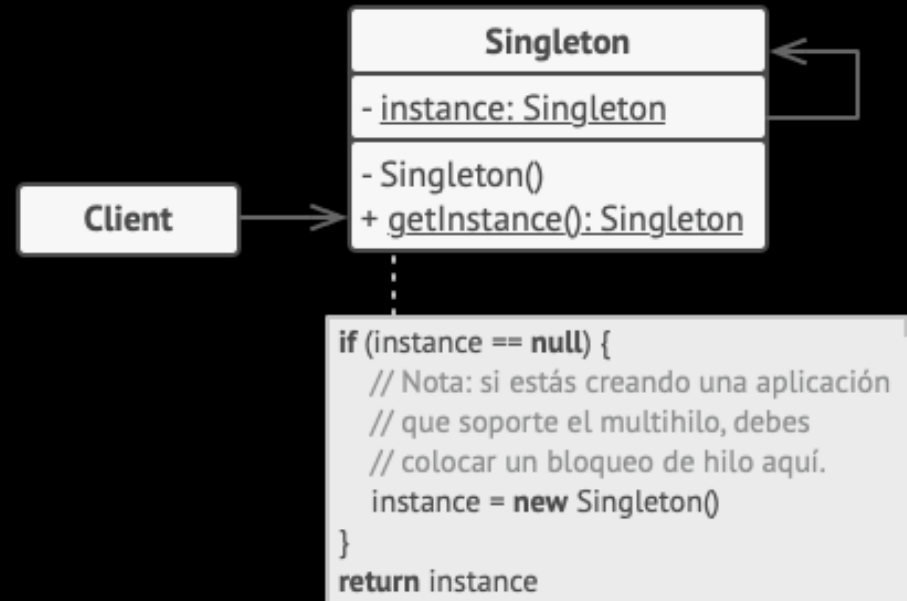
- Patrón de diseño creacional.
- Importante: Tener el control de las instancias de clases que estemos usando en un proyecto.
- ¿Qué pasaría si pudiéramos limitar a una única instancia en toda la aplicación a ciertas clases de nuestro proyecto?
- Singleton limita el número de instancias posibles de una clase en nuestro programa y proporciona un acceso global al mismo.

# ¿CUÁNDO USAR SINGLETON?

- Cuando nuestra clase o código tiene lógica que nunca (o casi nunca) va a cambiar y es utilizada muchas veces dentro del código.
- Pierde el sentido instanciar clases constantemente si siempre hacen lo mismo.
- Cuando necesitamos coordinar acciones entre uno o varios sistemas.

# OBJETIVO Y CONSTRUCCIÓN

- Mejora la Performance del sistema
- Ejemplo: Calendario.



## CUANDO NO UTILIZARLO

- Base de datos de alta concurrencia
- Anti patrón
- Cuellos de botella

**GRACIAS...**