**Juan Diego Gómez Triana**  
**Patrones de software - Corte 2**  
**Documentación patrón Singleton.**

El patrón Singleton asegura que una clase tenga una única instancia y proporciona un punto de acceso global a ella. Se utiliza cuando se necesita exactamente un objeto para coordinar acciones a lo largo del sistema.

Se implementa normalmente de la siguiente forma:

1. Se declara un campo privado estático que almacena la instancia única.
2. Se crea un método estático (generalmente llamado getInstance) que devuelve la instancia, creándola si aún no existe.
3. Se oculta el constructor haciendo que sea private, para evitar que se creen más objetos desde fuera de la clase.

Ejemplo clásico: una clase de configuración global en una aplicación o una clase que maneje el acceso a una base de datos. Esto garantiza que no haya conflictos o duplicidad de recursos.

En C# o Java, el patrón Singleton se usa ampliamente para manejar recursos compartidos o para implementar caches, gestores de conexiones o manejadores de logs.