Albeño, Fátima - 18060
De León, Diana – 18607
Algoritmos y Estructura de Datos, sección 10
Hoja de trabajo #4

Singleton

Este patrón de diseño asegura que una case solo tiene una instancia y proporciona un punto global de acceso a esta.

Usos:

- **Logger:** se tiene un archivo de log(recurso compartido) que solo debe ser accesado por un proceso, además de no permitir la creación de nuevos objetos cada vez que realiza una operación de logging.
- **Configuración:** permite tener un punto de acceso único a los parámetros de configuración de una aplicación manteniendo el estado de estos durante la ejecución, independientemente de donde se cargaron los valores al iniciar el Sistema, si de una base de datos o un archivo.
- **Pool y Factory:** usan Singleton como base ya que un Pool o Factory son objetos únicos qie tienen un estado que ofrece como servicio, permitiendo que la aplicación acceda a este estado siempre que lo necesite.

Desventajas:

- **Dificultad para realizar tiesting:** un Singleton es ocmo un valor de variable global. Lo global y la orientación a objetos no son buenos amigos ya que introduce un “estado persistente”, es decir, valores que se mantienen siempre dificultando el uso del objeto de reemplazo(mock en test).
- **Promociona el alto acoplamiento:** Singleton es instanciado directamente desde su propia clase promocionando el uso de métodos privados y estáticos. Esto acopla la clase que los use además de impedir el uso adecuado de inyección de dependencias.
- **Restricción de ejecuciones paralelas:** aunque un objetivo del Singleton sea la gestión de un recurso compartido esto restringe operar de forma paralela a la aplicación y lo transforma en un cuello de botella de operaciones seriales que no es recomendable cuando la demanda es alta.

El uso de Singleton fue adecuado ya que en algunas ocasiones solo se instanciaba un objeto.