**Hoja de Trabajo 4**

**Parte 7**

El patrón Singleton es un patrón de diseño de software que se utiliza para garantizar que una clase tenga una única instancia, y proporciona un punto de acceso global a esa instancia. A continuación, se presentan algunas ventajas y desventajas generales del patrón Singleton:

**Ventajas:**

Garantiza que solo haya una instancia de una clase en todo el sistema.

Proporciona un punto de acceso global a la instancia única.

Controla el acceso concurrente a la instancia única.

Permite una fácil extensión para la gestión de recursos compartidos.

**Desventajas:**

Puede aumentar la complejidad del código y la dependencia en el patrón.

El patrón puede hacer que el código sea menos flexible y menos escalable.

La implementación incorrecta del patrón Singleton puede ocasionar problemas de rendimiento y cuestiones de concurrencia.

El patrón Singleton puede ser difícil de probar, ya que no se pueden crear instancias múltiples para realizar pruebas unitarias.

Es importante tener en cuenta que el patrón Singleton no es la solución para todos los problemas de diseño de software. Debe ser utilizado con precaución y solo cuando sea necesario para evitar problemas de diseño y rendimiento.

**¿Cree que su uso es adecuado en este programa?**

Ya que el programa es algo complejo y requiere varias instancias el patron Singleton es una buena opción ya que este nos garantiza que solo haya una instancia de la clase en todo el sistema y ayuda a proporcionar un punto de acceso global a la instancia. Además, ya que se utilizan stacks y listas en el programa Singleton ayuda a que se maneje de forma segura y controlada la instancia.