

PROGRAMACION ORIENTADA A OBJETOS

GRUPO N°3

Integrantes:

- Lady Ipiales
- Gordon Josselyn
- Dilan Sanmartín
- Ariel Zavala

NRC: 8758

CODIGO LIMPIO

1. ¿Qué es el código limpio en POO?

- a) Un código que no tiene errores de sintaxis.
- b) Un código que es fácil de entender y mantener.**
- c) Un código que se ejecuta más rápido que otros códigos.
- d) Un código que utiliza muchos comentarios.

2. ¿Cuál de las siguientes no es una práctica recomendada para mantener un código limpio en POO?

- a) Aplicar principios SOLID.
- b) Utilizar nombres descriptivos para las variables y funciones.
- c) Evitar la reutilización de código.**
- d) Mantener las funciones y métodos cortos y con una única responsabilidad.

3. ¿Cuál de los siguientes principios SOLID promueve la separación de responsabilidades?

- a) Principio de responsabilidad única (SRP).**
- b) Principio de sustitución de Liskov (LSP).
- c) Principio de inversión de dependencia (DIP).
- d) Principio de segregación de interfaces (ISP).

4. ¿Qué significa el principio de responsabilidad única (SRP)?

- a) Cada clase debe tener una única responsabilidad.**
- b) Cada método debe tener una única responsabilidad.
- c) Cada objeto debe tener una única responsabilidad.
- d) Cada línea de código debe tener una única responsabilidad.

5. ¿Cuál de los siguientes no es un beneficio de seguir el principio de responsabilidad única (SRP)?

- a) Facilita la reutilización de código.
- b) Mejora la legibilidad y comprensión del código.

c) Permite un mantenimiento más sencillo del código.

d) Aumenta la complejidad del código.

6. ¿Qué es el acoplamiento en POO?

a) La forma en que los objetos se comunican entre sí.

b) La forma en que se agrupan las clases en un proyecto.

c) La dependencia entre las clases y sus módulos.

d) La medida de complejidad de un código.

7. ¿Cuál de los siguientes es un principio que ayuda a reducir el acoplamiento en POO?

a) Principio de sustitución de Liskov (LSP).

b) Principio de inversión de dependencia (DIP).

c) Principio de segregación de interfaces (ISP).

d) Principio de abstracción (ABP).

8. ¿Qué significa el principio de inversión de dependencia (DIP)?

a) Las clases de alto nivel no deben depender de las clases de bajo nivel.

b) Las clases de bajo nivel no deben depender de las clases de alto nivel.

c) Las clases no deben depender de interfaces concretas, sino de abstracciones.

d) Las interfaces concretas deben depender de las clases.

9. ¿Cuál de los siguientes no es una práctica recomendada para mejorar la legibilidad del código en POO?

a) Utilizar nombres descriptivos para las variables y métodos.

b) Comentar cada línea de código para explicar su funcionamiento.

c) Mantener el código bien estructurado y organizado.

d) Aplicar indentación y espaciados adecuados.

10. ¿Qué se entiende por robustez en el código limpio?

a) La capacidad del código para ejecutarse sin errores de sintaxis.

b) La capacidad del código para adaptarse a diferentes plataformas y entornos.

c) La capacidad del código para manejar situaciones inesperadas y errores de manera adecuada.

d) La capacidad del código para ejecutarse más rápido que otros códigos.