

MATERIA	Desarrollo Web Avanzado	NRC	14761	PRUEBA No.	1
CARRERA	Ing. de Software	Docente		Ing. Diego Saveedra	
PERIODO ACADÉMICO	Nov2023 – Mar2024	FECHA		13/12/2023	
TEMA	Preguntas de reflexión				
ESTUDIANTE(S)	Michelle Yánez				

A. DESARROLLO

Preguntas de Reflexión:

1. ¿Cuál es la diferencia entre una variable y una propiedad en C#?

Variable: Es un contenedor que almacena un valor en la memoria. Puede ser de diferentes tipos de datos, como int, string, etc.

Propiedad: Es una abstracción de una variable que permite controlar el acceso a los datos. Las propiedades en C# suelen tener un campo de respaldo (backing field) y proporcionan un nivel de encapsulación, permitiendo la lectura y escritura controlada de los datos asociados.

2. Explica cómo funciona la estructura if y por qué es útil en programación.

- La estructura if es una sentencia condicional en C# que permite ejecutar un bloque de código si una condición es verdadera.
- Sintaxis: `if (condicion) { // código a ejecutar si la condición es verdadera }`.
- La utilidad principal radica en controlar el flujo de ejecución del programa. Permite que ciertos bloques de código se ejecuten solo cuando se cumple una condición específica, lo que ayuda a tomar decisiones y realizar acciones específicas en función de las circunstancias.

3. ¿Qué ventajas ofrece la programación orientada a objetos en comparación con otros paradigmas de programación?

- Reutilización de código: Los objetos y las clases permiten la reutilización de código, ya que puedes crear clases que representen conceptos genéricos y luego heredar o instanciar esas clases en otros lugares del código.

- **Abstracción:** La POO permite modelar el mundo real de manera más natural mediante la abstracción de entidades y sus interacciones.
- **Encapsulación:** Oculta la implementación interna de los objetos y expone solo la interfaz necesaria para interactuar con ellos.
- **Herencia y Polimorfismo:** Facilita la creación de jerarquías de clases y el uso de polimorfismo, lo que simplifica la estructura del código y facilita su mantenimiento.

4. ¿Cuándo usarías la herencia en un diseño de clases?

- La herencia se utiliza cuando hay una relación lógica del tipo "es un" entre dos clases. Una clase derivada (subclase) hereda características y comportamientos de una clase base (superclase).
- Se utiliza para reutilizar y extender el código existente, evitando la duplicación y facilitando el mantenimiento.
- Es útil cuando varias clases comparten propiedades y comportamientos comunes, pero también tienen características únicas.

5. ¿Por qué es importante la encapsulación en programación orientada a objetos?

- **Seguridad y control de acceso:** La encapsulación limita el acceso directo a los detalles internos de un objeto, protegiendo la integridad de los datos y evitando modificaciones no deseadas.
- **Abstracción:** Permite ocultar los detalles de implementación, exponiendo solo la interfaz necesaria para interactuar con el objeto. Esto facilita el cambio interno sin afectar el código externo.
- **Mantenimiento del código:** Mejora la modularidad y facilita el mantenimiento del código al limitar el impacto de los cambios en una parte específica del programa.