

UNIVERSIDAD DE LAS FUERZAS ARMAS ESPE

Nombre: Richard Alban

Materia: Desarrollo Web Avanzado

NRC: 14956

Docente: Diego Medardo Saavedra Garcia

Preguntas de Reflexión

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

En C#, las variables son contenedores para almacenar datos y una propiedad es una forma de encapsular el acceso a los campos de una clase.

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

Es útil en programación porque permite elegir cuál de las dos rutas de acceso al código se deben seguir en función de una expresión booleana.

Funciona primeramente evaluando la condición a la que esta basada donde si la condición se cumple el programa ejecuta todas las instrucciones dentro del if, sino continua el flujo del programa

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

Reutilización de código donde se puede encapsular funcionalidades en clases y luego reutilizar esas clases en diferentes partes de tu programa o en diferentes proyectos.

Herencia donde me permite crear nuevas clases basadas en clases ya existentes donde se puede heredas todos sus atributos y métodos

Escalabilidad es decir el código orientado a objetos tiende a ser más escalable ya que nuevas funcionalidades se pueden agregar extendiendo clases existentes o creando nuevas clases.

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

La herencia es un concepto que permite crear clases nuevas a partir de otras existentes, reutilizando su código y añadiendo funcionalidades específicas, además la podemos usar cuando se quiere modelar una relación de tipo “es un” o “es una” entre clases.

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

La encapsulación consiste en ocultar los detalles de implementación de un objeto, es decir, sus atributos y métodos internos, y ofrecer una interfaz pública para acceder a ellos, además la encapsulación es importante porque permite proteger, modularizar y mantener el código de una clase, de esta forma, se evita que otros objetos puedan modificar los datos de forma directa o indebida, y se garantiza su integridad y coherencia.