

Universidad Autónoma de Santo Domingo
Primada de América / Fundada el 28 de octubre de 1538

Foro Unidad I
100738271 Jeffry Gilma

INF5120-3 Lenguaje de programacion

Victor Nu*ez Vicente

Peravia: República Dominicana

Septiembre2026

Temas obligatorios del foro

1. Programación Orientada a Objetos (POO)

1. LA DIFERENCIA ENTRE CLASE Y OBJETO, UTILIZANDO UN EJEMPLO PROPIO:

La clase es como el modelo o el diseño que define las características y las acciones que tendrá algo. No ocupa ninguna memoria del programa como entidad física. Por el otro lado tenemos el Objeto, que es la realidad, la instancia de la clase. Es como si tomas un plano y construyes una casa. Cada objeto tiene su propia identidad.

LA DIFERENCIA ENTRE EL CONCEPTO DE ESTADO Y COMPORTAMIENTO EN UN OBJETO.

El estado de un objeto es mutable, osea que se puede cambiar, pero define la identidad actual del objeto. Representa los datos o las características del objeto en un momento determinado, mientras que el comportamiento del objeto es lo que hace el objeto, define las acciones que el objeto puede realizar o las operaciones que se pueden ejecutar sobre el.

Seleccione dos pilares de la POO (abstracción, encapsulamiento, herencia o polimorfismo) y explique:

ABSTRACCIÓN

La abstracción es relevante porque se trata de el proceso de remover características de algo, para definir sus atributos y métodos que va a tener. Ejemplo para definir que es una canción, debemos primeramente abstraerlo añadiendo solo los atributos necesarios que tenemos planeado. Si el objetivo es hacer una biblioteca de música los atributos pueden ser; Nombre, artista, duración y género. Como método podemos poner escuchar y el reproductor.

QUE ERRORES COMUNES SE COMETEN AL NO APLICARLOS CORRECTAMENTE.

Código Frágil: Esto sucede cuando la abstracción es pobre, al estar pobre una parte del software provoca errores inesperados en lugares totalmente remotos.

Dificultad para crear pruebas unitarias: Bueno la abstracción correcta lo que hace es aislar componentes, y si no está bien abstracta, es casi imposible probarla de forma aislada.

Duplicación de lógica: Si no se abstracta una funcionalidad común en una clase, diferentes desarrolladores crean sus propias versiones de esa lógica en varias partes del software.

ENCAPSULACION

Es relevante porque a través de la modularidad podemos detectar los fallos individuales de nuestro programa más fácil, ya que cada clase debe tener control propio de lo que sucede dentro de ella, y tener los métodos adecuados para poder ser utilizados desde fuera.

QUE ERRORES COMUNES SE COMETEN AL NO APLICARLOS CORRECTAMENTE.

Corrupción de datos: Cualquier parte del sistema puede modificar el estado interno de un objeto de forma ilegal.

Acoplamiento de datos: Si se expone los detalles interno, se obliga a que todo el software dependa de esa decisión específica.

Control de versiones (Git / GitHub / GitLab)

Por que el control de versiones es indispensable en desarrollo de software, incluso en proyectos pequeños.

El error está en creer que que GIT solo sirve para compartir códigos. Pero sin GIT, si cometes un error catastrófico (borras una carpeta o rompes una lógica que funciona), tu única opción es confiar en tu memoria o ctrl + Z infinito. Además que permite crear líneas temporales alternativas llamadas Branches o Ramas.

Que ventajas aporta al trabajo colaborativo y al mantenimiento del código.

En el mantenimiento representa cerca del 80% del costo de la vida de un software. Git es la herramienta principal para reducir el costo. El mantenimiento moderno está autorizado. Herramienta de CI/CD se conectan a tu repositorio de Gitlab o Github.