TECNOLOGO EN ANILISIS Y DESARROLLO DE SOFTWARE

FICHA 3223899

LOGICA DE PROGRAMACIÓN.

TALLER: Clases y Métodos Abstractos en Java (NetBeans)

Resultado de Aprendizaje

Codificar el software empleando el lenguaje de programación

seleccionado, aplicando principios de la programación

orientada a objetos.

1. Actividad de Reflexión Inicial

Propósito: Activar conocimientos previos sobre herencia y el

concepto de abstracción.

Reflexiona y responde en tu cuaderno o archivo de texto:

• ¿Qué diferencia existe entre una clase concreta y una clase

abstracta?

rst/ Una clase concreta es la que uno puede usar directamente para crear objetos, porque ya tiene todo listo: sus métodos, sus atributos y su comportamiento bien definido. En cambio, una clase abstracta es más como un modelo o una guía; no se puede usar tal cual para crear objetos, sino que sirve para que otras clases hereden de ella y completen los detalles que faltan.

• ¿Por qué crees que algunos métodos no deben tener

implementación en una clase padre?

rst/ hay métodos que no deberían tener implementación en la clase padre porque cada subclase necesita comportarse de manera distinta por ejemplo en la clase animal,no se podría colocar habla porque todo animal tiene distinto sonido y manera de hablar sería mejor como implementarlo para cada subclase y que decida cómo hacerlo.

• ¿Qué ventajas trae definir una estructura base que obligue

a las subclases a implementar ciertos comportamientos?

rst/ yo diría que tener una estructura que obligue a la subclase a implementar esos comportamientos es útil porque asegura que todas sigan una misma lógica y cumplan con las cierta regla establecida y hace que el código esté más ordenado y fácil de mantener.

2. Actividades de Contextualización

Objetivo: Comprender el uso y necesidad de clases y métodos

abstractos.

Situación problema: Eres parte de un equipo que desarrolla un

sistema de diseño gráfico. Todas las figuras deben tener un

método para calcular el área, pero cada figura la calcula de

forma diferente. El analista te sugiere crear una estructura base

común, sin definir exactamente cómo se calcula el área.

Conceptos clave a repasar:

• Una clase abstracta es aquella que no puede instanciarse

directamente.

• Puede tener métodos concretos y abstractos.

• Los métodos abstractos no tienen cuerpo y deben ser

implementados por las subclases.

• Las subclases que heredan de una clase abstracta deben

definir el comportamiento específico.

RST/ En un sistema de diseño gráfico, todas las figuras deben tener un método para calcular el área, pero cada una lo hace de forma diferente. Por eso se usa una clase abstracta llamada, por ejemplo, “Figura”, que define la estructura común (como el método calcularÁrea), pero sin implementar ese método. Las subclases como “Círculo” o “Cuadrado” heredan de “Figura” y escriben su propia forma de calcular el área.

Esto permite mantener el código ordenado, facilitar la expansión del sistema y asegurar que todas las figuras sigan la misma estructura, aunque cada una tenga su propio comportamiento.

3. Actividades de Apropiación del Conocimiento

Ejercicio 1: Clase Abstracta Básica

Antes de codificar, elabore el diagrama de clases, que

represente la relación entre `FiguraGeometrica`, `Circulo`,

`Rectangulo` y `Triangulo`. Incluya atributos, métodos y

relaciones de herencia.

Objetivo: Implementar una jerarquía de clases que

representen figuras geométricas.

Pasos:

1. Crea un proyecto en NetBeans llamado:

TallerClasesAbstractas

2. Dentro del paquete modelo, crea una clase abstracta

FiguraGeometrica con métodos abstractos y concretos.

3. Crea las subclases Circulo, Rectangulo y luego Triangulo

implementando calcularArea().

4. Crea la clase Main para ejecutar las pruebas.