

**DEFINIR ESTÁNDARES DE CODIFICACIÓN DE ACUERDO A
PLATAFORMA DE DESARROLLO ELEGIDA**

Laura Daniela Gómez González

**CENTRO DE FORMACIÓN PARA EL DESARROLLO RURAL Y MINERO
SENA**

2834894: ANALISIS Y DESARROLLO DE SOFTWARE

Instructora. LUZ KARIME CASTELLANOS JOYA.

FEBRERO 2025.

INTRODUCCION

La programación orientada a objetos es un paradigma de programación que permite modelar objetos del mundo real, sus propiedades y sus interacciones en un programa de computadora. En la actualidad, este paradigma se ha vuelto cada vez más popular debido a su capacidad para facilitar la creación de programas más complejos y escalables, lo que ha llevado a su aplicación en una amplia variedad de campos, desde el desarrollo de software hasta la inteligencia artificial y la robótica. En la programación orientada a objetos, los objetos se modelan como entidades que tienen un estado, un comportamiento y una identidad. Estos objetos interactúan entre sí mediante el intercambio de mensajes, lo que permite la creación de programas más modulares y reutilizables. Además, la programación orientada a objetos proporciona herramientas para la abstracción, encapsulamiento, herencia y polimorfismo, que son conceptos fundamentales para la creación de programas más complejos.

OBJETIVO

La aplicación del paradigma orientado a objetos tiene como objetivo principal modelar objetos del mundo real en un programa de computadora y permitir su interacción de manera más natural y sencilla. Entre sus objetivos específicos se encuentran la facilitación de la creación de programas más complejos, la mejora de la escalabilidad, la eficiencia en el desarrollo, y la reutilización del código.

ESTÁNDARES DE CODIFICACIÓN:

Nombramiento de variables:

Una clase en java se compone por las siguientes declaraciones:

- Paquete
- Comentarios
- Definición de clases
- Atributos
- Constructores
- Métodos

Java determina en sus estándares de codificación convenciones para nombres de clases, métodos y atributos. Para las clases se define lo siguiente:

1. La primera letra debe ser mayúscula.
2. Utilizar nomenclatura camelCase (por ejemplo: un nombre compuesto por varias palabras: clase "CuentaAhorros").
3. Los nombres deben ser sustantivos.

Ejemplo:

Estructura:

```
public class NombreDeLaClase {}
```

Donde la palabra 'public' indica que el alcance de la clase será público.

La palabra 'class' indica que es una clase.

Y las llaves {} se usan para agrupar los bloques de código.

Todos los identificadores Java presentan unas condiciones en su nomenclatura que se resumen en los siguientes aspectos:

Deben tener nombre claro y conciso

Se permite el uso del carácter subrayado (_), exceptuando la posición

final e inicial del identificador

No se permiten dos caracteres subrayados consecutivos.

El nombre de la clase, Publicacion debe ser un identificador válido en Java y por convención, empieza por letra mayúscula. Una declaración de clase crea un nombre de tipo en Java. De esta forma las referencias a objetos Publicacion se pueden declarar con un simple:

Nombre de la clase:

Las variables de una clase se llaman campos. La clase Publicación tiene tres campos, idPublicacion que identifica de forma unívoca cada publicación, título que es el título de la publicación y autor que ha firmado la publicación. La creación de un objeto en Java es equivalente a decir que se está instanciando una clase. Cuando se hace la instancia se comparten los atributos y métodos de esta. Una sola clase puede tener varias instancias.

Declaración de un método:

Un método consta de un encabezado y un cuerpo. Para declarar el encabezado de un método, basta con escribir el tipo que retorna, seguido del nombre del método y entre paréntesis la lista de parámetros.

Ejemplos:

void imprimir(); este método no retorna nada y no tiene parámetros.

CONCLUSION

En conclusión, la programación orientada a objetos es un paradigma de programación poderoso que permite modelar objetos del mundo real en un programa de computadora y facilita la creación de programas más complejos y escalables. En la actualidad, su aplicación se ha vuelto cada vez más popular debido a sus ventajas en la creación de programas más modulares, reutilizables y mantenibles.