Definir estándares de codificación de acuerdo a plataforma de desarrollo elegida.

GA7-220501096-AA1-EV02.

PRESENTADO POR:

LUIS EDUARDO CHALA SERRADA

**Ficha programa de formación: 2455277**



INSTRUCTOR:

[HENRY TORRES PERTUZ](https://sena.territorio.la/perfil.php?persona=1770103&grupo=1423502)

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

SERVICIO NACIONAL DE APRENDIZAJE – SENA

CENTRO DE LOGÍSTICA Y PROMOCIÓN ECOTURÍSTICA DEL MAGDALENA​

2023

Introducción: La programación orientada a objetos es un paradigma de programación que permite modelar objetos del mundo real, sus propiedades e interacciones en un programa informático. En la actualidad, este paradigma se ha hecho 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 gran 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 crear programas más modulares y reutilizables. Además, la programación orientada a objetos proporciona herramientas de abstracción, encapsulación, herencia y polimorfismo, que son conceptos fundamentales para crear programas más complejos.

Objetivo: El objetivo principal de la aplicación del paradigma orientado a objetos es modelar objetos del mundo real en un programa informático y permitir su interacción de forma más natural y sencilla.

**Pautas del Estándar de Codificación Java**

A continuación se presentan algunas pautas básicas del estándar de codificación Java:

**Nomenclatura**

* Las clases y las interfaces deben seguir la convención de nomenclatura de UpperCamelCase.
* Los métodos y variables deben seguir la convención de nomenclatura de lowerCamelCase.
* Las constantes deben escribirse en mayúsculas y separarse con guiones bajos.

**Indentación y Espaciado**

* Utilizar una indentación de 4 espacios para cada nivel de bloque de código.
* Utilizar espacios para separar operadores y elementos en expresiones.
* No dejar espacios en blanco al final de las líneas de código.

**Comentarios**

* Utilizar comentarios para explicar el propósito y la lógica del código.
* Los comentarios deben estar escritos en inglés, a menos que se especifique lo contrario.

**Manejo de Excepciones**

* Utilizar bloques try-catch para manejar excepciones de manera adecuada.
* Evitar el uso de bloques catch vacíos o genéricos.

**Convenciones de Estilo**

* Utilizar llaves en la misma línea que las declaraciones de control de flujo.
* Limitar la longitud de las líneas de código a 80 caracteres.
* Utilizar espacios en blanco para mejorar la legibilidad del código.

**Conclusiones**

El estándar de codificación Java es una herramienta fundamental para el desarrollo de software de calidad. Siguiendo estas pautas, se mejora la legibilidad, la consistencia y la calidad del código Java, lo que facilita su mantenimiento y colaboración en proyectos de desarrollo de software.

Referencias bibliográficas:

# Obtener Java para aplicaciones de escritorio

[Java | Oracle](https://www.java.com/es/)

Mejores practicas para implementar java

[10 Best practices to design and implement a Java class | by lahlali issam | Javarevisited | Medium](https://medium.com/javarevisited/10-best-practices-to-design-and-implement-a-java-class-489f72a7099a)

[Un breve resumen de las mejores prácticas de codificación Java | por Rafiullah Hamedy | Medio (medium.com)](https://rhamedy.medium.com/a-short-summary-of-java-coding-best-practices-31283d0167d3)

Estándares de codificacion

[Java Foundations: Java - Estándares de programación](http://javafoundations.blogspot.com/2010/07/java-estandares-de-programacion.html?m=1)

variables

[Variables (The Java™ Tutorials > Learning the Java Language > Language Basics) (oracle.com)](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html)