EVIDENCIA DE APRENDIZAJE – DEFINIR ESTÁNDARES DE CODIFICACIÓN DE ACUERDO A PLATAFORMA DE DESARROLLO ELEGIDA

GA7-220501096-AA1-EV02

FASE – EJECUCION

Por:

GEFFRY SANTIAGO LEON JIMENEZ

CENTRO DE LA TECNOLOGÍA DEL DISEÑO Y LA PRODUCTIVIDAD EMPRESARIAL

REGIONAL CUNDINAMARCA – SENA – GIRARDOT

ANÁLISIS Y DESARROLLO DE SOFTWARE

FICHA 2977481

INSTRUCTOR: MILTON IVAN BARBOSA GAONA

30/07/2025

Tabla de Contenido

[**1.** **INTRODUCCIÓN** 4](#_Toc204778631)

[**2.** **OBJETIVOS** 5](#_Toc204778632)

[**2.1 Objetivo general** 5](#_Toc204778633)

[***2.1.1 Objetivos especificos*** 5](#_Toc204778634)

[**3.** **ESTANDARES DE CODIFICACION** 6](#_Toc204778635)

[**3.1 ¿Qué son los estandares de codificacion?** 6](#_Toc204778636)

[**3.2 Nombramiento de variables** 6](#_Toc204778637)

[**3.3 Declaracion de Clases** 8](#_Toc204778638)

[**3.4 Declaracion de metodo** 9](#_Toc204778639)

[**3.5 Terminacion de sentencias** 10](#_Toc204778640)

[**4.** **CONCLUSIONES** 11](#_Toc204778641)

[**5.** **REFERENCIAS BIBLIOGRÁFICAS** 12](#_Toc204778642)

# **INTRODUCCIÓN**

En los lenguajes de programacion existen estandares de codificacion, en este caso hablaremos de esto para asi poder reforzar estos conocimientos que mas adelante nos seran muy utiles a la hora de estar programando, en este caso hablare de los estandares del lenguaje Java, ya que es el que se explica en el material de aprendizaje.

# **OBJETIVOS**

## **2.1 Objetivo general**

Poder aprender en los estandares de codificacion de java para asi tener un mejor desarrollo a la hora del aprendizaje en las evidencias de programacion.

### ***2.1.1 Objetivos especificos***

* Aprender estandares de codificacion
* Memorizar datos
* Avanzar en mi aprendizaje

# **ESTANDARES DE CODIFICACION**

## **3.1 ¿Qué son los estandares de codificacion?**

Los estandares de codificacion son las reglas y convencios que los desarrolladores siguen al momento de escribir el codigo. Estas reglas definen las caracteristicas del codigo que se necesitan para poder tener una base de codigo uniforme y que pueda facilitar la colaboracion entre los desarrolladores. (Codacy, 2025)

## **3.2 Nombramiento de variables**

Para poder nombrar correctamente una variable y poder tener un codigo mucho mas comodo se recomienda que cumpla con las siguientes convenciones: (SENA, Aplicacion del paradigma orientado a objetos, 2021)

* En java, por convencion las variables empieza con una letra minuscula, si esta es una palabra compuesta la primera letra en mayuscula con la nomenclatura camelCase (SENA, Aplicacion del paradigma orientado a objetos, 2021)
* Las variables no pueden tener espacios ni iniciar con un numero, y tampoco puede contener caracteres especiales (SENA, Aplicacion del paradigma orientado a objetos, 2021)
* No puede ser el mismo que una palabra clave (SENA, Aplicacion del paradigma orientado a objetos, 2021)
* No deben tener el mismo nombre que otras variables cuyas declaraciones llegue a aparecer en el mismo ambito (SENA, Aplicacion del paradigma orientado a objetos, 2021)

Por ejemplo:

* edadCliente
* segundoNombre
* sangreTipo

para poder nombrar una variable tambien tenemos que pensar en la sintaxis, para esto tenemos varios datos que se complementan de los nombre de las variables: (SENA, Aplicacion del paradigma orientado a objetos, 2021)

* byte: este representa un tipo de dato de 8 bits con signo. De tal manera que este puede almacenar los valores numericos de -128 a 127 (Web, 2025)
* short: representa un tipo de dato de 16 bits. Almacena valores numericos desde -32.768 a 32.767 (Web, 2025)
* int: tipo de dato de 32 bits con signo para almacenar valores numericos. Valor minimo es -231 y el valor maximo 231-1 (Web, 2025)
* long: tipo de datos 64 bits con signo que almacena valores numericos entre -263 a 263-1 (Web, 2025)
* float: tipo de dato para almacenar numeros en coma flotante con precision simple de 32 bits (Web, 2025)
* double: dato para almacenar numero en coma flotante con doble precision de 64 bits (Web, 2025)
* char: tipo de dato representa carácter unicode sencillo de 16 bits
* string: texto (Web, 2025)
* boolean: define tipo de datos booleanos. Es decir, los que tienes valor de true o false. Ocupa 1 bit de informacion (Web, 2025)

## **3.3 Declaracion de Clases**

Las clases son los principales elementos de un lenguaje de programacion, estos es debiado a que se usan para todas las operaciones logicas en un desarrollo. Se podria decir que son estructuras o plantillas que representan cosas que existen de verdad. (SENA, Aplicacion del paradigma orientado a objetos, 2021)

Una clase en java esta conformada por estas declaraciones:

* paquete
* comentarios
* definicion de clases
* atributos
* constructores
* metodos

Java determina en sus estandares de codificacion estas conveciones para usarlas en los nombres de clases, metodos y atributos. En este caso se esa lo siguiente (SENA, Aplicacion del paradigma orientado a objetos, 2021):

* la primera letra deber ser en mayuscula
* se debe utilizar la nomenclatura camelCase, si en alguna caso el nombre esta compuesto por favorias palabras se pondria asi (NombreCliente).

Java determino esto ya que hace que podamos identificar de forma rapida de que un elemento es una clase.

* los nombres deben de ser sustantivos (se debe escribir “Producto” no “Productos”) (SENA, Aplicacion del paradigma orientado a objetos, 2021)

escribir asi los nombres hace que se pueda reflejar una mayor precision de que la clase es un molde para una unica instancia de ese tipo. (SENA, Aplicacion del paradigma orientado a objetos, 2021)

Este es un ejemplo de la estructura de una clase:

Public class

NombreDeLaClase {}

La palabra public indica que el alcance de la clase sera publico, la palabra class indica que es una clase y las llaves {} se usan para agrupar los bloques de codigo (SENA, Aplicacion del paradigma orientado a objetos, 2021)

## **3.4 Declaracion de metodo**

Para empezar la definicion de un metodo se compone de dos partes (SARREPLEC, s.f.):

* cabecera del metodo: es la que contiene el nombre del metodo junto con el tipo devuelto, cun conjunto que contiene posibles modificadores y una lista de parametros. (SARREPLEC, s.f.)
* Cuerpo del metodo: el cuerpo en pocas palabras llega a ser el bloque de codigo que esta encerrado entre las llaves (SARREPLEC, s.f.)

Para hacer una declaracion de un metodo debe tener unos elementros minimos para que funcione: (SARREPLEC, s.f.)

* El tipo devuelto por el metodo
* El nombre del metodo
* Los parentesis
* El cuerpo del metodo entre llaves: { }

Podremos encontrar dentro del cuerpo del metodo declaraciones de variables, sentencias y todo tipo de estructuras de control (bucles, condiciones, etc.). (SARREPLEC, s.f.)

## **3.5 Terminacion de sentencias**

En toda linea que se escriba al estar programando se debe cerrar, para poder cerrarla se debe insertar el punto y coma (;) .

Esto es obligatorio ya que si no cerramos la linea de codigo con el punto y coma no la leera y nos votara un error a la hora de querer ejecutar todo el programa hecho.

# **CONCLUSIONES**

Aunque haya podido terminar este trabajo la verdad me parecio muy confuso ya que no lo pude entender, incluso ahora que lo termine no se si me quedaria bien, pero de todo lo que investigue se me quedo pegado muchas cosas que sinceramente me van a servir demasiado en un futuro, inclusive aprendi un poco de sintaxis por investigar algo que no era, pero me parecio divertido poder realizar este trabajo.

# **REFERENCIAS BIBLIOGRÁFICAS**

Java: Nomenclatura de variables: <https://basicos.codica.la/languages/java/lessons/naming-style#:~:text=En%20Java%20se%20utiliza%20CamelCase,%2C%20se%20ver%C3%ADa%20as%C3%AD:%20miSuperVariable%20>.

Estándares de codificación: ¿qué son y por qué son importantes? : <https://blog.codacy.com/coding-standards>

Manual Web Tipos de datos primitivos en java : <https://www.manualweb.net/java/tipos-datos-primitivos-java/>

Declaracion de un metodo: <https://sarreplec.caib.es/pluginfile.php/11616/mod_resource/content/2/41_declaracin_de__un_mtodo.html>

Material de formacion: <https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF28/index.html#/curso/tema2>