

## **Aplicación Del Paradigma Orientado A Objetos**

**Planear Actividades De Construcción Del Software De Acuerdo Con El Diseño Establecido**

**Aprendiz:**

**Santiago Carranza Carrillo**

**Ficha:**

**2848530-A**

**Instructor:**

**Julio Roberto Galvis Cardozo**

**SENA**

**Centro de Diseño y Metrología**

**2 De diciembre 2024**

## Índice

Índice	2
Introducción	3
Objetivos	4
Estandares de codificacion	5

### Introducción

En el presente documento se estará presentando los estándares de codificación de acuerdo a la plataforma de desarrollo seleccionada, brindando las buenas prácticas en los aspectos de declaración de variables, declaración de clases y declaración de métodos.

### Objetivos

- Comprender los entornos de desarrollo orientado a la programación de objetos.
- Comprender los estándares de codificación de declarar variables, métodos y clases.
- Comprender los modificadores de visibilidad en las clases.
- Seleccionar un software adecuado para la gestión de versiones del código en el proyecto.

## **Desarrollo orientado a objetos**

El modelo de programación en el que diseño de un software se organiza alrededor de datos u objetivos, en vez de utilizar funciones lógicas, la definición de un objeto puede darse como un campo que contiene datos con atributos y comportamiento únicos.

La principal característica de este tipo de programación es que soporta objetos que tiene un tipo o clase asociado, dichas clases heredan atributos de una superclase o clase superior por lo dicha metodología de programación es mayormente utilizada en programa complejos que deben actualizarse regularmente.

## **Principales Lenguajes POO**

Los principales lenguajes de programación orientada a objetos con: C + +, Ruby, Java, Python, PHP.

Java y C + + son los dos lenguajes de programación más usados, debido a que estos son los más comunes y lo que en el mercado laboral más se pueden presentar y utilizar.

Asimismo, PHP, Python y Ruby son unos de los lenguajes más populares enfocados en la programación, desarrollo web y aplicaciones móviles, por lo que, si la intención es crear un aplicativo web robusto, es conveniente optar por el lenguaje POO PHP y Python para diseñar dicho aplicativo.

## **Estándares de codificación**

Los estándares de codificación incorporan ciertos principios sólidos para la programación en cada respectivo lenguaje, sienten estas reglas y pautas que ayudan a mantener un estándar

garantizado, ayudando a mejorar la colaboración, agilizar el mantenimiento y garantizar la fiabilidad general.

Además, dichos estándares de programación previenen errores y problemas de vulnerabilidad, mejorando la calidad del código.

El objetivo de los estándares de codificación de software es inculcar buenas prácticas a la hora de programar conduciendo a un código seguro, confiable y mantenible.

Para el estándar de codificación para las clases se dice lo siguiente:

- La primera letra debe ser mayúscula.
- Utilizar una nomenclatura camelCase.
- Los nombres deben ser sustantivos.

Para el estándar de codificaciones de variables se comenta que:

- Por convención, en Java, los nombres de las variables empiezan con una letra minúscula y si es una palabra compuesta la primera letra en mayúscula con nomenclatura camelCase.
- No pueden tener espacios ni iniciar con un número, ni caracteres especiales.
- No puede ser el mismo que una palabra clave
- No deben tener el mismo nombre que otras variables cuyas declaraciones aparezcan en el mismo ámbito.

Para los casos de un constructor posee las siguientes características especiales:

- El nombre del constructor es igual que el nombre de la clase.
- No tiene ningún retorno.
- Se pueden crear varios, siempre y cuando tengan diferentes atributos.

### **Modificadores de acceso a un conjunto de datos**

También, se puede encontrar los modificadores de acceso, estos permiten condicionar si se puede acceder a la información de una determinada clase o si de lo contrario, no es posible o esta misma está protegida. Encontramos cuatro tipos de modificadores de acceso:

1. **Default:** No se especifica ningún modificador de acceso para la clase, por ende, esta solo será accesible dentro del mismo paquete de datos.
2. **Private:** Permite que ninguna de las otras clases del mismo paquete pueda acceder a los demás conjuntos o miembros de datos dentro de las clases.
3. **Protected:** Todos los conjuntos de datos que hayan sido declarados como Protected solo serán accesibles dentro del mismo paquete o subclases en paquetes diferentes.
4. **Public:** Permite que estos miembros de datos sean de acceso público y que tengan un alcance hacia cualquier otro lugar del programa y no tiene ninguna restricción a la hora de utilizar estos mismos.

### **Referencias**

<https://colombia.universidadeuropea.com/blog/programacion-orientada-objetos/> (2024,

Universidad Europea)

[https://drive.google.com/drive/folders/1D\\_YdGEYZmhq2eMTDQ42fniOZpwTrs8S-](https://drive.google.com/drive/folders/1D_YdGEYZmhq2eMTDQ42fniOZpwTrs8S-?usp=drive_link)

[?usp=drive\\_link](https://drive.google.com/drive/folders/1D_YdGEYZmhq2eMTDQ42fniOZpwTrs8S-?usp=drive_link) (2024, Aplicación del Paradigma orientado a objetos)