



**PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL  
GUÍA DE APRENDIZAJE – JAVA POO.**

JUAN ESTEBAN BOTIA GOMEZ

ANALISIS Y DESARROLLO DE SOFTWARE(ADSO)  
2827167

OSMAN ALONSO ARANGUREN ESCOBAR

SENA  
SERVICIO NACIONAL DE APRENDIZAJ CIMA  
SOGAMOSO-BOYCA  
2024

- Responder la pregunta planteada y subirla en la carpeta como evidencia No 1.
- Observar el video: “¿Qué es la programación orientada a objetos?” link:
- [https://www.youtube.com/watch?v=DlphYPc\\_HKk](https://www.youtube.com/watch?v=DlphYPc_HKk)

**Respuesta:** El video explica el paradigma de la programación orientada a objetos (POO) de una manera clara y sencilla. El ponente, Álvaro Felipe, comienza definiendo los objetos como entidades del mundo real que pueden ser representados en código. Luego, explica que la programación tradicional, basada en instrucciones secuenciales, puede ser compleja para proyectos grandes.

### **3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje. TIEMPO 8 HORAS**

#### **3.2.1 Consulte utilizando la ayuda de la biblioteca del SENA link:**

<http://biblioteca.sena.edu.co/paginas/bases.html>, los siguientes conceptos:

#### **¿Qué es la Programación Orientada a Objetos (POO)?**

La POO es una forma de programar que se basa en la idea de "objetos". Un objeto es como una representación de algo del mundo real, como un carro, una persona o un número. Estos objetos tienen características (atributos) y acciones que pueden realizar (métodos). La POO nos ayuda a organizar nuestro código de una manera más intuitiva y fácil de mantener.

#### **¿Cuál es la historia de la Programación Orientada a Objetos?**

Los orígenes de la POO se remontan a los años 60, pero fue en los 70 cuando se consolidó con lenguajes como Simula. La idea era crear programas más cercanos al pensamiento humano, modelando el mundo real a través de objetos.

#### **¿Cuáles son los lenguajes de programación que soportan la POO?**

La mayoría de los lenguajes de programación modernos soportan la POO, algunos de los más populares son: Java, C++, Python, C#, JavaScript, Ruby, entre otros.

#### **¿Qué es un objeto?**

Un objeto es una instancia de una clase. Es como una plantilla que define las características y comportamientos de un conjunto de objetos similares.

#### **¿Qué es un atributo?**

Un atributo es una característica de un objeto. Por ejemplo, si tenemos un objeto "Carro", sus atributos podrían ser: color, marca, modelo, año.

#### **¿Qué es un método?**

Un método es una acción que un objeto puede realizar. Siguiendo el ejemplo del "Carro", los métodos podrían ser: acelerar, frenar, girar.

### ¿Qué es un método void?

Un método void es un método que no devuelve ningún valor. Simplemente realiza una acción.

### ¿Cuáles son los tipos de método?

Existen diferentes tipos de métodos, pero los más comunes son:

**Métodos de instancia:** Actúan sobre un objeto en particular.

**Métodos de clase:** Actúan sobre la clase en sí, sin necesidad de crear una instancia.

### ¿Qué es una clase, describa sus características?

Una clase es como un plano o molde para crear objetos. Define los atributos y métodos que compartirán todos los objetos de ese tipo.

### ¿Para qué sirve Getter y Setter?

Los getters y setters son métodos especiales que se utilizan para acceder y modificar los atributos de un objeto de manera controlada.

### ¿Para qué sirve this?

La palabra clave `this` se refiere al objeto actual dentro de un método. Se utiliza para distinguir entre los atributos de un objeto y los parámetros de un método.

### ¿Para qué sirve null?

`null` es un valor especial que indica que una variable no referencia a ningún objeto.

### ¿Qué es instanciar?

Instanciar es crear un objeto a partir de una clase. Es como tomar el molde (clase) y crear una copia concreta (objeto).

### ¿Cuáles son los pilares de la POO?

#### Los pilares fundamentales de la POO son:

**Abstracción:** Modelar el mundo real en términos de objetos.

**Encapsulamiento:** Ocultar la implementación interna de un objeto.

**Herencia:** Crear nuevas clases a partir de clases existentes.

**Polimorfismo:** La capacidad de que objetos de diferentes clases respondan de manera diferente al mismo mensaje.

### ¿Qué es abstracción?

La abstracción consiste en identificar las características esenciales de un objeto y omitir los detalles innecesarios.

### **¿Qué es polimorfismo?**

El polimorfismo permite que objetos de diferentes clases puedan ser tratados como si fueran del mismo tipo, siempre y cuando compartan un método común.

### **¿Qué es encapsulamiento?**

El encapsulamiento consiste en proteger los datos internos de un objeto y controlar el acceso a ellos a través de métodos.

### **¿Cuáles son los tipos de modificadores de acceso en JAVA?**

En Java, los modificadores de acceso controlan la visibilidad de los miembros de una clase. Los principales son:

**public:** Accesible desde cualquier parte.

**private:** Accesible solo dentro de la clase.

**protected:** Accesible dentro de la clase y sus subclases.

**default:** Accesible dentro del mismo paquete.

### **¿Qué es herencia?**

La herencia es un mecanismo que permite crear nuevas clases (subclases) a partir de clases existentes (superclases), heredando sus atributos y métodos.

### **¿Qué es un constructor?**

Un constructor es un método especial que se utiliza para inicializar los objetos cuando se crean.

### **¿Qué es un destructor?**

Un destructor es un método que se ejecuta automáticamente cuando un objeto se destruye. En algunos lenguajes, como C++, se utiliza para liberar la memoria ocupada por el objeto.

### **¿En qué consiste el concepto de memoria stack y heap?**

**Stack:** Se utiliza para almacenar variables locales y parámetros de métodos. Tiene un tamaño fijo y se gestiona de forma automática.

**Heap:** Se utiliza para almacenar objetos. Tiene un tamaño dinámico y se gestiona manualmente (o por el garbage collector).

### **¿Qué es una clase anidada?**

**Clase anidada:** Una clase que se define dentro de otra clase.

### **¿Qué es una clase abstracta?**

**Clase abstracta:** Una clase que no puede ser instanciada directamente y sirve como base para otras clases.

### **¿Qué es un mensaje en POO?**

Un mensaje es una solicitud que se envía a un objeto para que ejecute uno de sus métodos.

### **¿En qué consiste la sobrecarga de constructores y métodos?**

La sobrecarga consiste en tener múltiples métodos o constructores con el mismo nombre, pero con diferentes parámetros.

### **¿En qué consiste el uso de la palabra Final?**

La palabra clave `final` se utiliza para indicar que algo no puede ser modificado después de su inicialización. Puede aplicarse a variables, métodos y clases.

### **¿Qué es una colección?**

Una colección es un objeto que agrupa otros objetos. Sirve para almacenar y organizar múltiples elementos.

### **¿Qué es arraylist?**

Un ArrayList es una implementación de una lista dinámica en Java. Permite almacenar un número variable de elementos de un mismo tipo.

### **¿Qué es una interfaz?**

Una interfaz define un contrato que las clases deben cumplir. Especifica los métodos que una clase debe implementar, sin proporcionar una implementación.

### **¿Qué es una excepción?**

Una excepción es un evento que interrumpe el flujo normal de ejecución de un programa. Se utilizan para manejar errores y situaciones inesperadas.

### **Evidencia requerida No 2**

- Modalidad de trabajo: Grupal (dos aprendices)
- Elaborar un resumen con los conceptos anteriores y la sintaxis y subirlos a Territorium.
- Observar los siguientes videos:

<https://www.youtube.com/watch?v=cfGrtrdDGRk&list=PLGfF3KgbxaiwzDR3ijzdX3HTCNYToj3CE>.

[https://www.youtube.com/watch?v=oBO01Cx\\_YwQ&list=PLGfF3KgbxaiwzDR3ijzdX3HTCNYToj3CE&index=2](https://www.youtube.com/watch?v=oBO01Cx_YwQ&list=PLGfF3KgbxaiwzDR3ijzdX3HTCNYToj3CE&index=2)

<https://www.youtube.com/watch?v=gR0EssHrl24&list=PLGfF3KgbxaiwzDR3ijzdX3HTCNYToj3CE&index=3>

<https://www.youtube.com/watch?v=9NynVRpZzv4&list=PLGfF3KgbxaiwzDR3ijzdX3HTCNYToj3CE&index=4>

<https://www.youtube.com/watch?v=tjyecfz9Cvk&list=PLGfF3KgbxaiwzDR3ijzdX3HTCNYToj3CE&index=5>

**Sopa de letras:** <https://learningapps.org/view26522192>

**Respuesta:**

- **Video 1: "¿Qué es la Abstracción? - Programación Orientada a Objetos"**

**Resumen:** La abstracción consiste en tomar un concepto complejo del mundo real y representarlo en un programa de manera simplificada, enfocándose en sus características más importantes y omitiendo los detalles innecesarios. Esto permite crear modelos más manejables y comprensibles.

**Ejemplo:** Un automóvil puede abstraerse a un objeto con atributos como color, marca y modelo, sin detallar componentes internos como el motor.

- **Video 2: "¿Qué es el Encapsulamiento? - Programación Orientada a Objetos"**

**Resumen:** El encapsulamiento es el mecanismo que permite ocultar los detalles internos de un objeto, exponiendo solo una interfaz para interactuar con él. Esto protege los datos y evita modificaciones no deseadas.

- **Sintaxis (Java):**

```
java
public class Persona {
    private String nombre;
    private int edad;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nuevoNombre) {
        nombre = nuevoNombre;
    }
    // ...
}
```

- **Video 3: "Herencia en programación orientada a objetos"**

**Resumen:** La herencia es un mecanismo que permite crear nuevas clases (subclases) a partir de una clase existente (superclase), heredando sus atributos y métodos. Esto fomenta la reutilización de código y la creación de jerarquías de clases.

- **Sintaxis (Java):**

```
java
public class Animal {
    // ...
}

public class Perro extends Animal {
    // ...
}
```

- **Video 4: "Polimorfismo en Java"**

**Resumen:** El polimorfismo es la capacidad de que objetos de diferentes tipos puedan ser tratados como si fueran del mismo tipo. Se manifiesta en la sobrecarga de métodos (distintos métodos con el mismo nombre, pero diferentes parámetros) y en la sobreescritura de métodos (un método en una subclase que re implementa un método de la superclase).

- **Sintaxis (Java):**

```
java
public class Animal {
    public void hacerSonido() {
        System.out.println("Hace un sonido");
    }
}

public class Perro extends Animal {
    @Override
    public void hacerSonido() {
        System.out.println("Ladra");
    }
}
```

## **SUBACTIVIDADES**