# Temario Proyecto POO

## **Proyecto: Tutorial de POO**

1. **Introducción a la programación orientada a objetos**
   1. **Antecedentes de la programación orientada a objetos**
   2. **Enfoque orientado a objetos**
   3. **Conceptos de la programación orientada a objetos**
   4. **Ventajas de la programación orientada a objetos**
   5. **Lenguajes orientados a la programación orientada a objetos**
2. **Implementación de clases y objetos en Java**

**2.1 Clases  
2.2 Objetos e instanciación   
2.3 Protección  
2.4 Constructores y destructores**

1. **Pilares de POO en Java**

**3.1 Encapsulamiento  
3.2 Sobrecarga de funciones (métodos)  
3.3 Herencia  
3.4 Polimorfismo**

**Introducción a la programación orientada a objetos**

* 1. **Antecedentes de la programación orientada a objetos**

**Los conceptos de la POO tienen origen en Simula 67, un lenguaje diseñado para hacer simulaciones, creado por Ole-Johan Dahl y Kristen Nygaard, del Centro de Cómputo Noruego en Oslo. En este lenguaje se introdujeron los conceptos de clases, rutinas y subclases.**

**Mas tardé, estos conceptos fueron refinados en Smalltalk, desarrollado en Simula en Xerox PARC (cuya primera versión fue escrita sobre Basic) pero diseñado para ser un sistema completamente dinámico en el cual los objetos se podrían crear y modificar "sobre la marcha" (en tiempo de ejecución) en lugar de tener un sistema basado en programas estáticos.**

* 1. **Enfoque orientado a objetos**

**La programación orientada a objetos (siglas POO en español y en ingles OOP) es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.**

**La POO se fue convirtiendo en el estilo de programación dominante a mediados de los años 1980, en gran parte debido a la influencia de C++, una extensión del lenguaje de programación C. Su dominación fue consolidada gracias al auge de las interfaces gráficas de usuario, para las cuales la POO está particularmente bien adaptada. En este caso, se habla también de programación dirigida por eventos.**

**Las características de orientación a objetos fueron agregadas a muchos lenguajes existentes durante ese tiempo, incluyendo Ada, BASIC, Lisp más Pascal, entre otros.**

* 1. **Conceptos de la programación orientada a objetos**

**La POO es una forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:**

**Clase**

**Una clase se puede definir de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ella.**

**Herencia**

**Por ejemplo, herencia de la clase C a la clase D, es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables registrados como "públicos" (public) en C. Los componentes registrados como "privados" (private) también se heredan pero se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Para poder acceder a un atributo u operación de una clase en cualquiera de sus subclases pero mantenerla oculta para otras clases es necesario registrar los componentes como "protegidos" (protected), de esta manera serán visibles en C y en D pero no en otras clases.**

**Objeto**

**Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales del mundo que nos rodea, o con objetos internos del sistema (del programa).**

**Método**

**Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.**

**Evento**

**Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento la reacción que puede desencadenar un objeto; es decir, la acción que genera.**

**Atributos**

**Características que tiene la clase.**

**Mensaje**

**Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.**

**Propiedad o atributo**

**Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.**

**Estado interno**

**Es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.**

**Componentes de un objeto**

**Atributos, identidad, relaciones y métodos.**

**Identificación de un objeto**

**Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.**

**En comparación con un lenguaje imperativo, una "variable" no es más que un contenedor interno del atributo del objeto o de un estado interno, así como la "función" es un procedimiento interno del método del objeto.**

* 1. **Ventajas de la programación orientada a objetos**

**Estas ventajas están dadas por las características de POO, las cuales permiten de alguna manera que los programas sean mas versátiles a la hora de escribirse y ejecutarse.**

**Una lista de ventajas es la siguiente:**

* 1. **Lenguajes orientados a la programación orientada a objetos**

**-Java**