

SERVICIO NACIONAL DE APRENDIZAJE – SENA



CENTRO DE COMERCIO REGIONAL ANTIOQUIA

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE - 2675805

**Evidencias de desempeño: definir estándares de codificación de acuerdo a
plataforma de desarrollo elegida. GA7-220501096-AA1-EV02**

DANIEL FELIPE ARIAS CORREDOR

2023

Introducción

Este documento tiene como objetivo presentar una noción general acerca de la Programación Orientada a Objetos o POO y su uso en entornos de programación. Primero se explicarán algunas nociones generales de POO mediante ejemplos en lenguaje *Java*. Posteriormente se presentará el paso a paso de la instalación del entorno de desarrollo *Eclipse*.

¿Qué es la programación orientada a objetos?

La programación orientada a objetos o POO es un paradigma para la programación de software, el cual hace especial énfasis en trasladar la naturaleza de los objetos de la vida real a las entidades que se crean en código. Por tanto, los objetos del POO tienen un estado, un comportamiento (que pueden hacer) y unas propiedades. Como ejemplo, un objeto creado en código denominado 'automovil_1' puede tener atributos similares a uno de la vida real (color, peso, modelo), así como comportamientos (arrancar, frenar) y un estado (detenido, circulando). Los programas se modelan en torno a objetos.

¿Cuáles son las características de la programación orientada a objetos?

La programación orientada a objetos trabaja con ciertos conceptos que son sus componentes principales a la hora de construir el software, ellos son: clases, objetos y atributos.

El objeto es el núcleo o la razón de ser de este paradigma. Un objeto es una entidad en código con el que el programa puede interactuar y trata de imitar a uno de la vida real. Un objeto consta de estados, atributos y comportamiento, además de pertenecer a una clase. Ej. 'automovil_1' y 'automóvil_2' son objetos diferentes, aunque compartan una clase y unas propiedades.

Una clase es un modelo o plantilla que contiene características comunes a un grupo de objetos. Al crear un objeto se debe llamar a esta clase o instanciarla para que el objeto herede sus atributos. Por ejemplo, se ha creado la clase 'automovil' con atributos como 'numero_ruedas', 'peso' y 'motor' y comportamientos como

Arrancar(), al crear un objeto como automóvil_1 perteneciente a esta clase, el objeto heredará los comportamientos y atributos con un valor predefinido.

Un atributo son las propiedades que vienen a poseer los objetos dentro de una clase. Los objetos pertenecientes a una clase tienen los mismos atributos, pero sus valores pueden diferir. Ej. automovil_1 tiene los atributos 'numero_ruedas', 'peso' y 'motor', al igual que automovil_2, aunque para este último el valor de 'motor' es de 1800 frente a 2100 de automovil_1.

Pilares de la Programación Orientada a Objetos

La POO tiene cuatro principios o pilares necesarios a la hora de crear un software basado en este paradigma. Los pilares son: herencia, abstracción, encapsulamiento y polimorfismo.

La herencia permite que una clase herede o traiga para sí características provenientes de otras clases. Esta herencia se da en un orden jerárquico, lo que quiere decir que una clase hereda de una clase padre o superclase. Siguiendo el ejemplo, si a la clase automóvil le es creada una subclase llamada 'camioneta', esta última puede heredar los atributos de su superclase, por lo que camioneta tendrá los atributos 'numero_ruedas', 'peso' y 'motor' y el comportamiento Arrancar(). A la hora de asignar esta herencia se debe apelar a la lógica y considerar la regla que un objeto de la subclase pueda pertenecer a la superclase (una camioneta sí es un automóvil).

El encapsulamiento consiste en la posibilidad de aislar ciertos comportamientos o propiedades dentro de una clase para que solo sean visibles y modificados dentro de la misma clase, consiguiendo que no puedan ser accedidos por otras clase o métodos. Ej. La clase automóvil solo tiene como público Arrancar() y 'numero_ruedas', siendo 'peso' y 'motor' atributos privados e inalterables.

La abstracción permite representar características de una clase al exterior, pero ocultado su complejidad interna. No sólo controla el acceso a la información como el encapsulamiento, sino también oculta la complejidad de los métodos. La abstracción junto con el encapsulamiento proporciona modularidad al proyecto de

software, permitiendo que otro módulo o usuario solo pueda hacer uso de las propiedades o métodos habilitados para un módulo, siendo el resto de la clase una especie de caja negra.

Por último, el polimorfismo permite que un objeto se pueda comportar de forma diferente dependiendo del contexto. Un objeto de una subclase puede reemplazar a uno de la superclase, por lo que se puede usar de manera indistinta siempre y cuando tenga una base en común. Ej. Se pueden usar un objeto automóvil_1 y el objeto motocicleta_1 indistintamente para la superclase automóvil, aunque tengan sus peculiaridades internas.

Cómo declarar una variable

Una variable es un espacio de memoria en el sistema donde se guarda determinado valor o dato. La declarar de una variable en Java tiene la siguiente estructura:

```
>int numero;
```

‘int’ es una palabra reservada del lenguaje que indica el tipo de dato que se almacenará, mientras que ‘numero’ es el nombre de la variable. De esta manera la máquina sabe que debe reservar un espacio en memoria de nombre numero para almacenar un dato de tipo numérico.

Cómo definir una variable

Definir una variable es asignarle un valor. En Java la definición de una variable se logra con la siguiente sintaxis:

```
>numero=5;
```

De esta manera la máquina sabe que a la variable ‘numero’ se le está asignando un valor y lo debe almacenar. En Java es posible declarar y definir una variable en una sola línea de código, tal como se muestra a continuación:

```
>int numero = 5;
```

Cabe resaltar que a las variables se le puede cambiar el valor en cualquier momento.

¿Cómo se declara un método?

Un método una función creada para una clase y viene a representar el comportamiento de sus objetos. Determinan las tareas básicas que realizará el objeto. Ej. La clase automóvil contiene el método Arrancar(), por lo que al ser llamado para un objeto como parámetro: ">Arrancar(automóvil_1)", el objeto iniciará este comportamiento.

Antes de hacer el llamado a un método, este debe haber sido creado. Para crear un método en Java se hace uso de la siguiente sintaxis.

```
>public class automovil {  
  
    Public static void Arrancar(auto) {  
  
        En este espacio se anotan el código con las funciones deseadas  
  
    }  
  
}
```

Cómo se acaba de ver, dentro de la clase automovil se ha creado un método denominado 'Arrancar', de tal manera que cuando sea llamado y se le pase como parámetro un objeto ej: Arrancar(objeto), el programa realizará las funciones deseadas dentro del método solo para este objeto.

¿Qué es eclipse?

Eclipse es un entorno de desarrollo para la creación de código en diferentes lenguajes, usado especialmente en el trabajo con lenguajes de programación orientada a objetos. Uno de estos lenguajes orientados a objetos es Java, por lo que eclipse al ser usado para construir software con este lenguaje permite la elaboración de código y su ejecución, así como la organización de clases, paquetes y módulos.



Figura 1. Logotipo de eclipse

¿Cómo se instala eclipse?

Para instalar eclipse como entorno de desarrollo o IDE se recomienda de antemano haber instalado Java. Como primer paso se debe acceder al sitio web de eclipse: eclipse.org/downloads, se abrirá una interfaz como la de la figura 2.

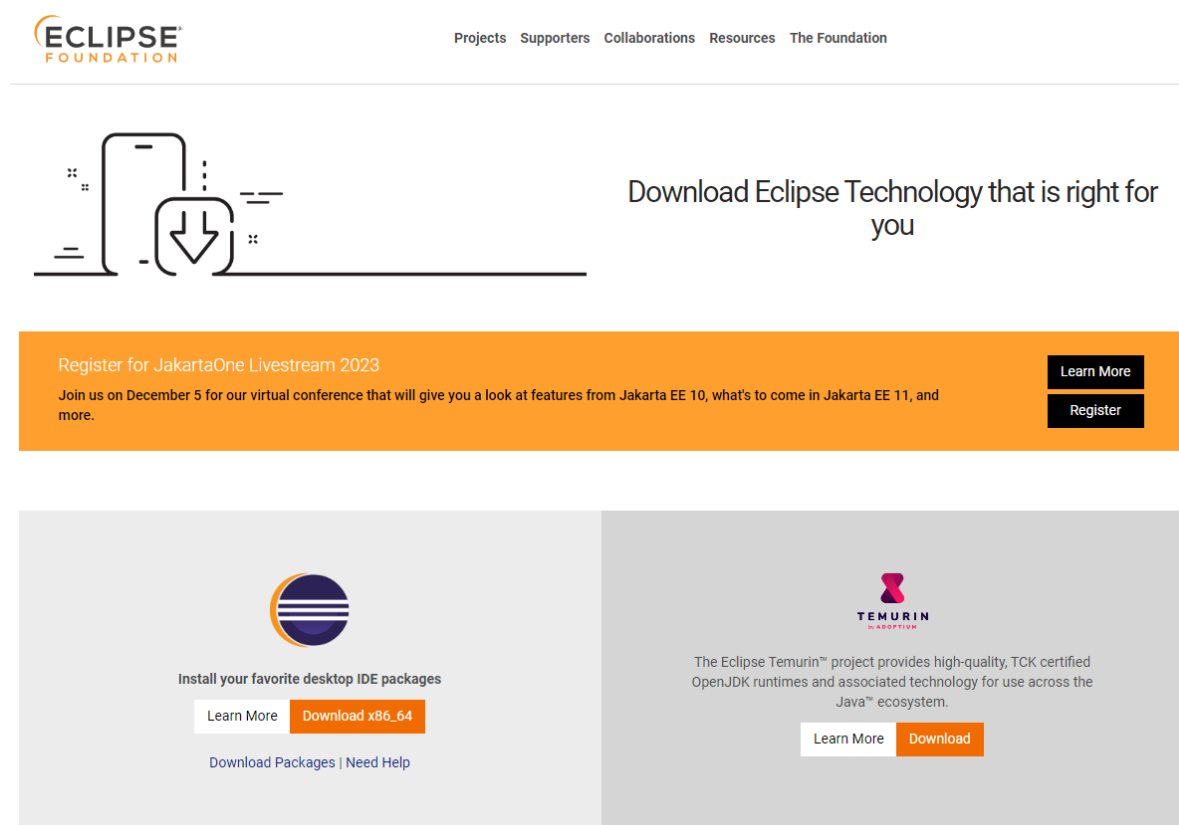


Figura 2

Luego de su descarga se seleccionará una carpeta como directorio de trabajo (figura 3)

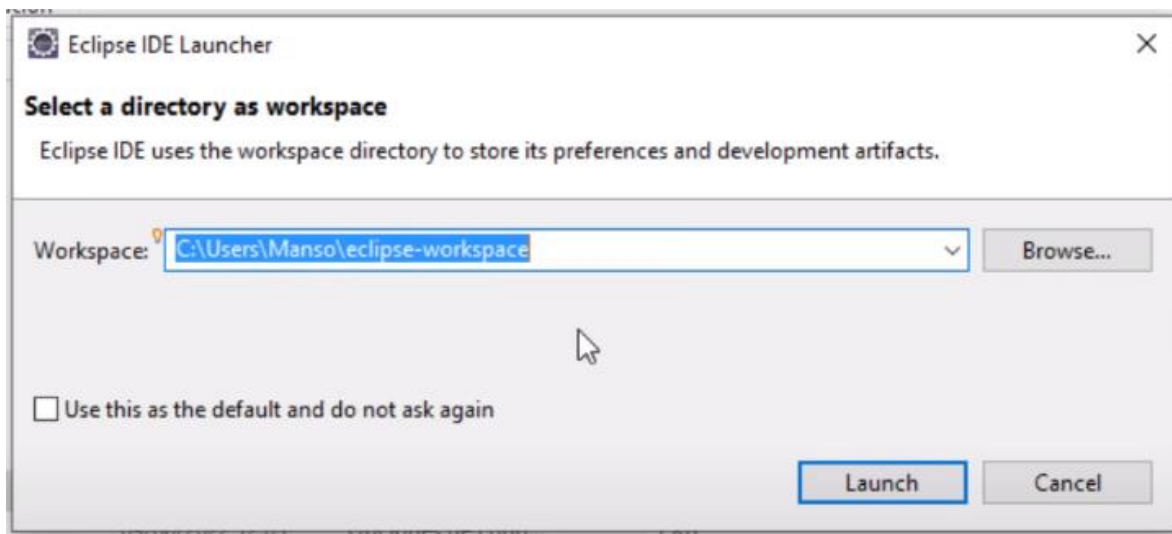


Figura 3. Recuperado de: <https://www.pildorasinformaticas.es>

Se espera entonces a que se instale (figura 4).



Figura 4. Recuperado de: <https://www.pildorasinformaticas.es>

Una vez instalado, su página de inicio se asemeja a la de la figura 5.

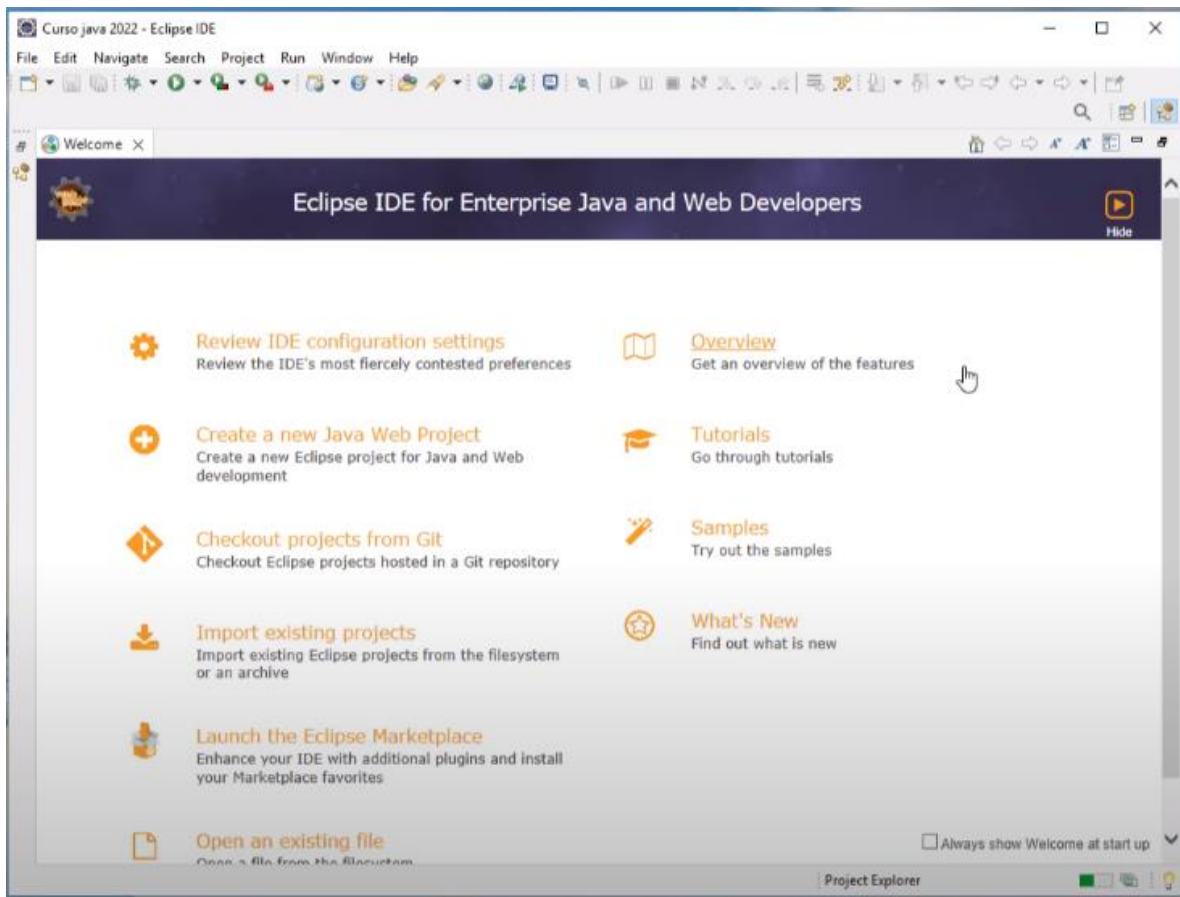


Figura 5. Recuperado de: <https://www.pildorasinformaticas.es>

Para iniciar un proyecto Java, basta con dirigirse a la pestaña de archivo, luego nuevo y finalmente nuevo proyecto (figura 6).

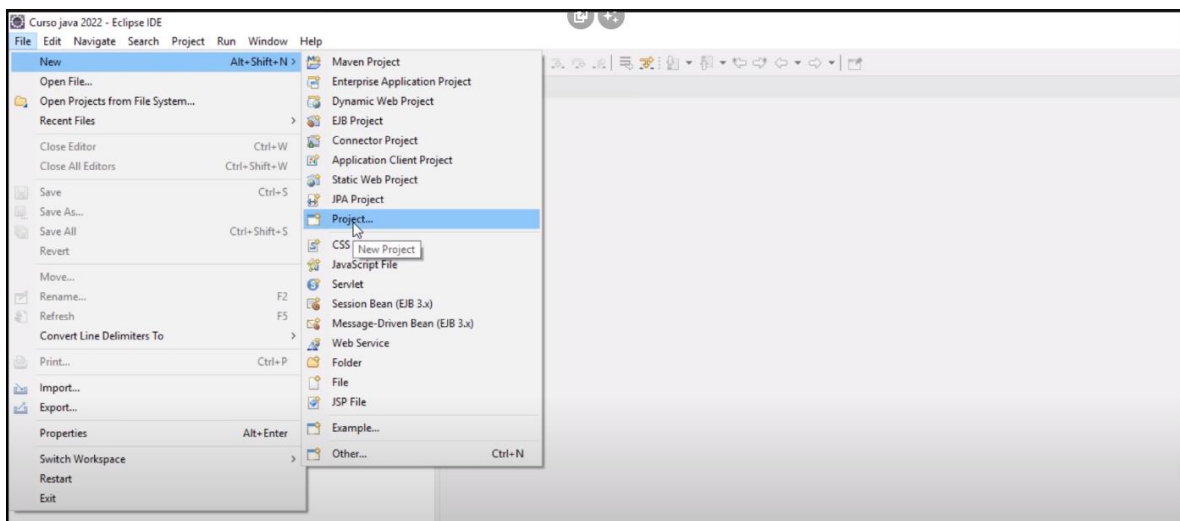


Figura 6. Recuperado de: <https://www.pildorasinformaticas.es>

En la ventana emergente (figura 7) se selecciona Java Project y se da clic en aceptar.

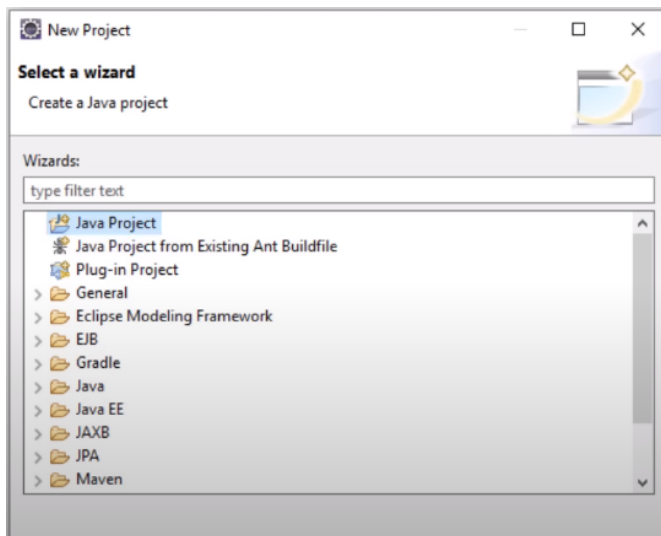


Figura 7. Recuperado de: <https://www.pildorasinformaticas.es>

La figura 8 ejemplifica la ventana que sigue donde se debe dar un nombre al proyecto y seleccionar la versión de Java a usar.

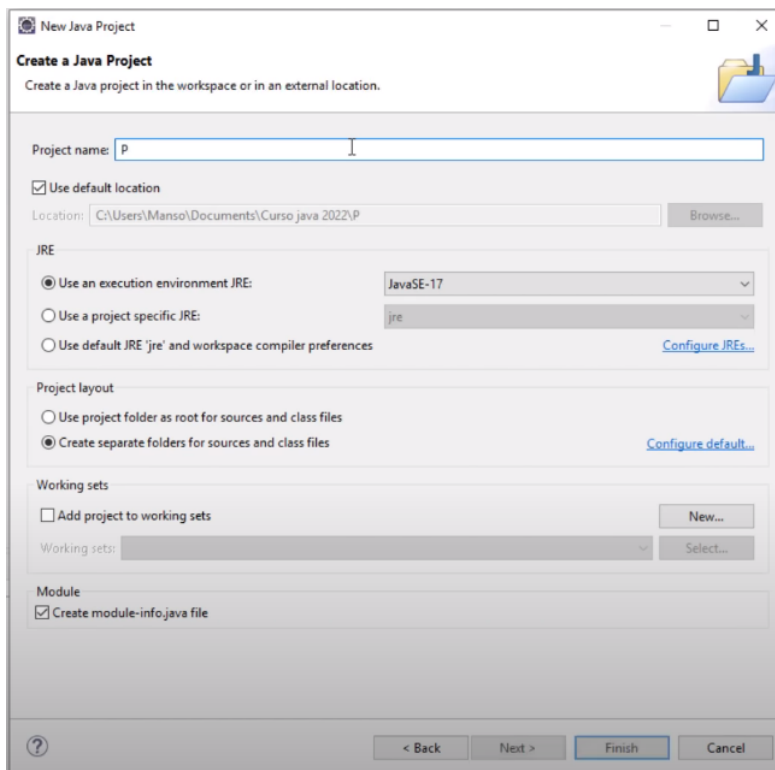


Figura 8. Recuperado de: <https://www.pildorasinformaticas.es>

Una vez creado el proyecto, la interfaz de eclipse nos permitirá crear la primera clase del proyecto siguiendo los pasos de la figura 9 y añadiendo la información necesaria para su creación (figura 10).

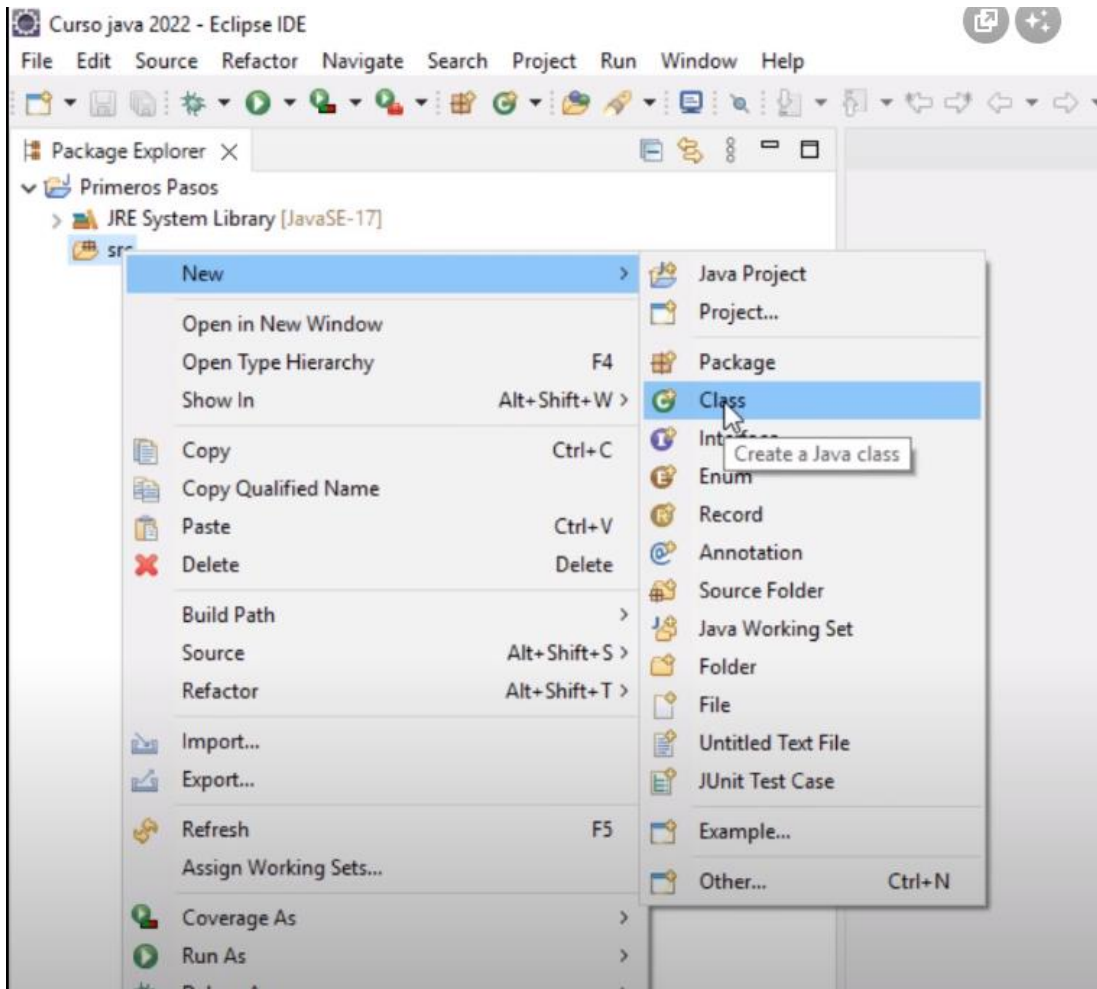


Figura 9. Recuperado de: <https://www.pildorasinformaticas.es>

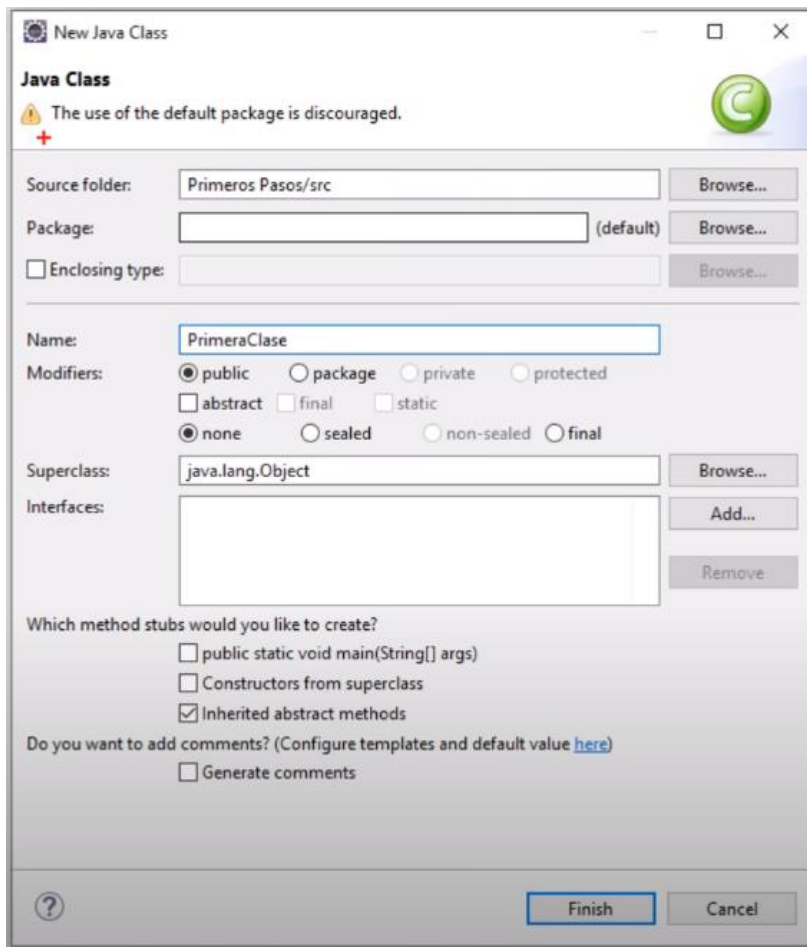
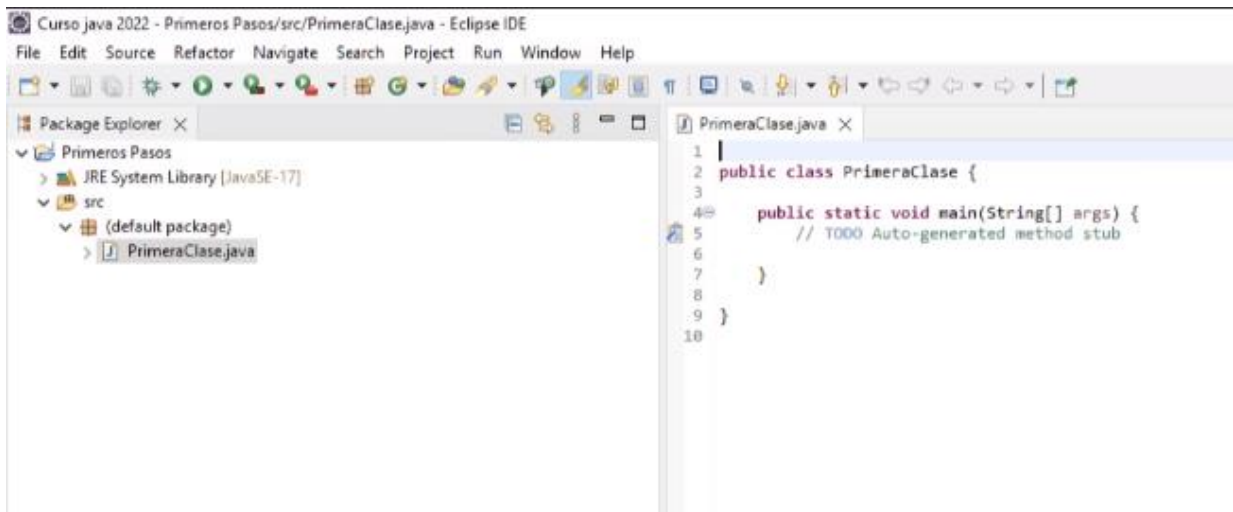


Figura 10. Recuperado de: <https://www.pildorasinformaticas.es>

Una vez definida la clase, eclipse nos presentará su editor de código (figura 11), con el código para la definición de la clase previamente creada.



Referencias:

¿Qué son las variables en Java? Recuperado de:

<https://www.manualweb.net/java/variables-java/>

Expresiones y operadores. Recuperado de:

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Expressions_and_Operators#aritmeticos

Cómo llamar a un método en Java. Recuperado de:

<https://codegym.cc/es/groups/posts/es.642.como-llamar-a-un-metodo-en-java>

Java desde 0. Módulo 1. Recuperado de:

<https://www.pildorasinformaticas.es/course/java-desde-0/>