

# ECLIPSE

Iván, Noam, Jorge

1º DAW

# TABLA DE CONTENIDO

• Introducción	
• Índice	02
• Historia y contexto	03
• Instalación de una máquina	04
• Partes y uso del IDE	05
• Partes y uso del IDE	06
• Partes y uso del IDE	07
• Valoración personal	08
• Recomendacion de uso	09

# HISTORÍA Y CONTEXTO

01

## Historia

El proyecto Eclipse fue iniciado por IBM en noviembre de 2001. En 2004, se estableció la Fundación Eclipse como una organización independiente

02

## ¿Para qué sirve?

Eclipse es conocido principalmente por su robusto soporte para Java. Sin embargo soporta una gama amplia de lenguajes y tecnologías. Incluye C/C++, PHP, Python, JavaScript, y herramientas para desarrollo web, bases de datos y servidores de aplicaciones

03

## Destacable

- 2001: Lanzamiento inicial por parte de IBM como proyecto de código abierto [1].
- 2004: Fundación de la Fundación Eclipse independiente.
- Arquitectura modular
- Ciclos de lanzamiento coordinados

# REQUISITOS

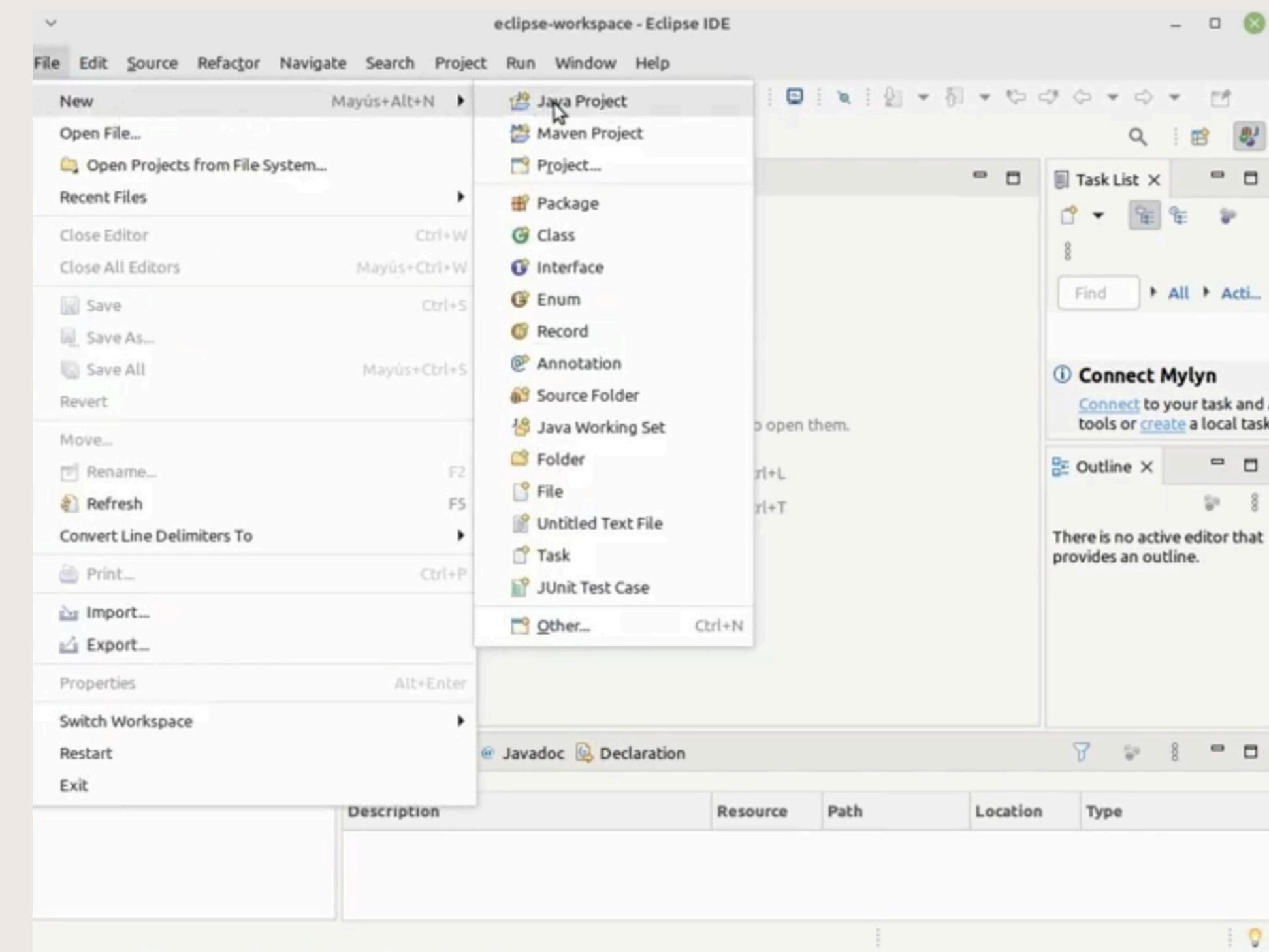
Ubuntu Server 24.04 LTS

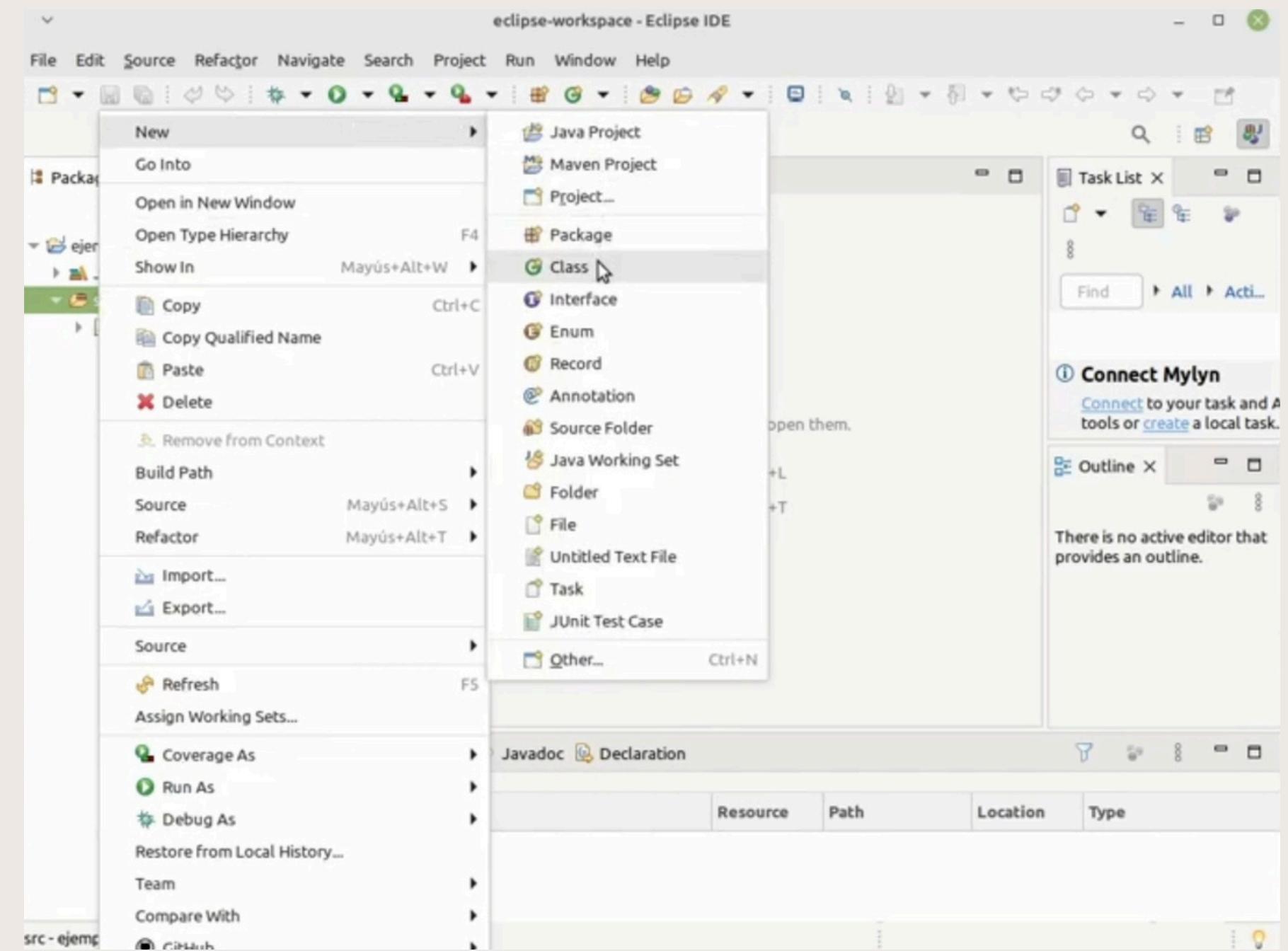
- Sistema Operativo: Un sistema operativo moderno de 64 bits (Windows, macOS o Linux).
- Memoria RAM: Mínimo 2 GB, aunque se recomiendan 4 GB o más para un rendimiento fluido.
- Espacio en Disco: Mínimo 250 MB para el IDE, más espacio adicional para el JDK y los proyectos de trabajo.

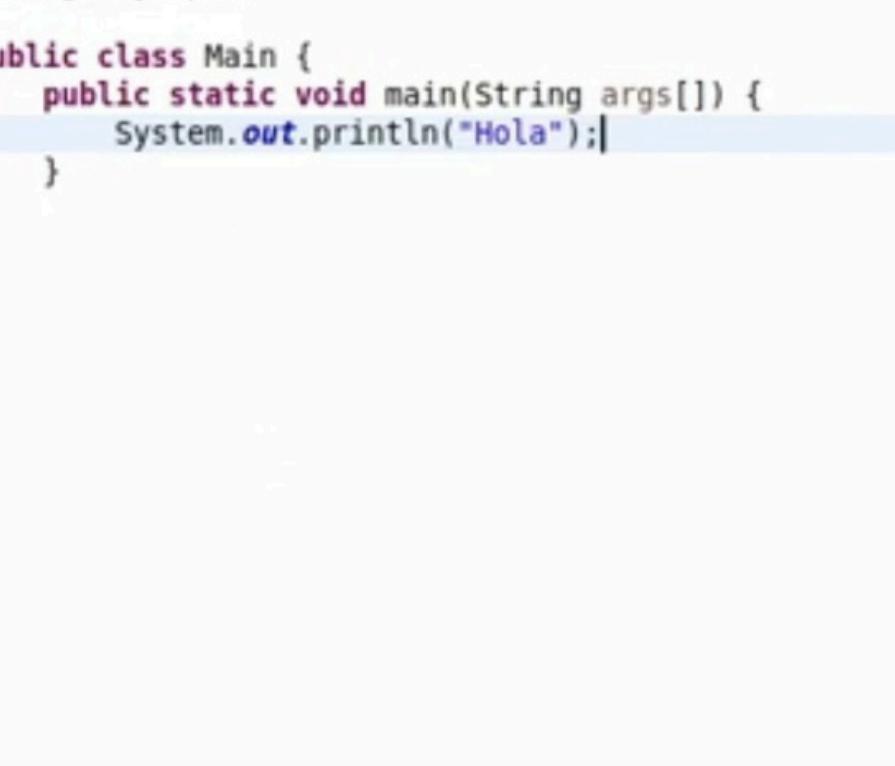
Java: Un Java Development Kit (JDK) instalado y configurado correctamente. Las versiones actuales del instalador de Eclipse requieren al menos JDK 17 o superior para funcionar.

# INSTALACIÓN EN UNA MÁQUINA

- Instalación de Eclipse
- Descarga: Busca "Eclipse" en Google, ve a la página oficial de descargas y descarga el instalador para Linux.
- Extracción y Ejecución: Descomprime el archivo .tar.gz. Entra en la carpeta generada y ejecuta el instalador, acepta los términos.
- Finalizar: Elige la versión deseada (ej. Eclipse IDE for Java Developers) y completa la instalación.
- Crear Proyecto: Ve a File > New > Java Project (o el tipo que necesites). Nómbralo y haz clic en Finish.
- Crear Archivo: En el Package Explorer, haz clic derecho sobre el proyecto o la carpeta src.
- Seleccionar Tipo: Elige New > Class (para Java) o File (para genérico).
- Nombrar y Terminar: Escribe un nombre adecuado (con la extensión si es File) y estarás listo para programar.







```
1 package ejemplo;
2
3 public class Main {
4     public static void main(String args[]) {
5         System.out.println("Hola");
6     }
7 }
8
```

The screenshot shows the Eclipse IDE interface. At the top, there's a toolbar with icons for file operations like New, Open, Save, and Print. Below the toolbar is a menu bar with 'File', 'Edit', 'Search', 'Project', 'Run', 'Help', and 'Window'. The main workspace contains two open perspectives: 'Task List' on the left and 'Outline' on the right. The 'Task List' perspective has a 'Find' input field and buttons for 'All' and 'Active'. The 'Outline' perspective shows a tree view with nodes 'ejemplo' and 'Main'. Under 'Main', there's a node 'main(String[]) : void' which is highlighted in green, indicating it is the currently selected element.

A screenshot of the Eclipse IDE interface. The top menu bar shows tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console X'. Below the menu bar is a toolbar with various icons. The main area displays the output of a Java application named 'Main'. The output text reads: '<terminated> Main [Java Application] /home/eliud/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86\_64\_11 Hola'.

# PARTES Y USO DEL IDE

## Editor del código

**Resaltado de Sintaxis:** Colorea diferentes partes del código (variables, comentarios) para mejorar la legibilidad y ayudar a identificar errores tipográficos.

**Autocompletado:** Sugiere automáticamente nombres de variables, funciones y clases a medida que se escribe, lo que acelera la codificación y reduce errores.

## Compilador

Compilación: Para lenguajes compilados (como C++, Java), el IDE ejecuta el compilador para crear un archivo ejecutable a partir del código fuente.

Para lenguajes interpretados como Python ejecuta el programa en una consola que muestra la salida del programa.

# PARTES Y USO DEL IDE

## Depurador

**Puntos de Interrupción:** El usuario puede establecer puntos específicos en el código donde quiere que la ejecución del programa se pause.

**Inspección de Variables:** Una vez pausado, el depurador permite ver los valores actuales de las variables, lo que ayuda a entender el estado del programa. Permite la ejecución por linea observando como cambia el valor de las variables.

## Control de versiones

Eclipse incluye la herramienta **EGit** que proporciona integración con el sistema de versiones Git. Esto permite realizar operaciones comunes de Git desde el propio IDE.

**Subversion:** Es uno de los sistemas más populares compatibles con Eclipse. Requiere la instalación de un plugin, como Subversive y Subclipse. Permite una gestión completa de repositorios SVN dentro del IDE.

# PARTES Y USO DEL IDE

## Refactorización

La refactorización es el proceso de reestructurar el código existente sin cambiar su comportamiento externo

**Renombrar:** Cambiar el nombre de una variable.

**Seleccionar un bloque de código y convertirlo automáticamente en una nueva función o método.**

## Plugins

**Soporte de Lenguajes Adicionales:**  
Extensiones para dar soporte completo a lenguajes no incluidos por defecto

**Temas e Iconos:** Opciones de personalización visual.

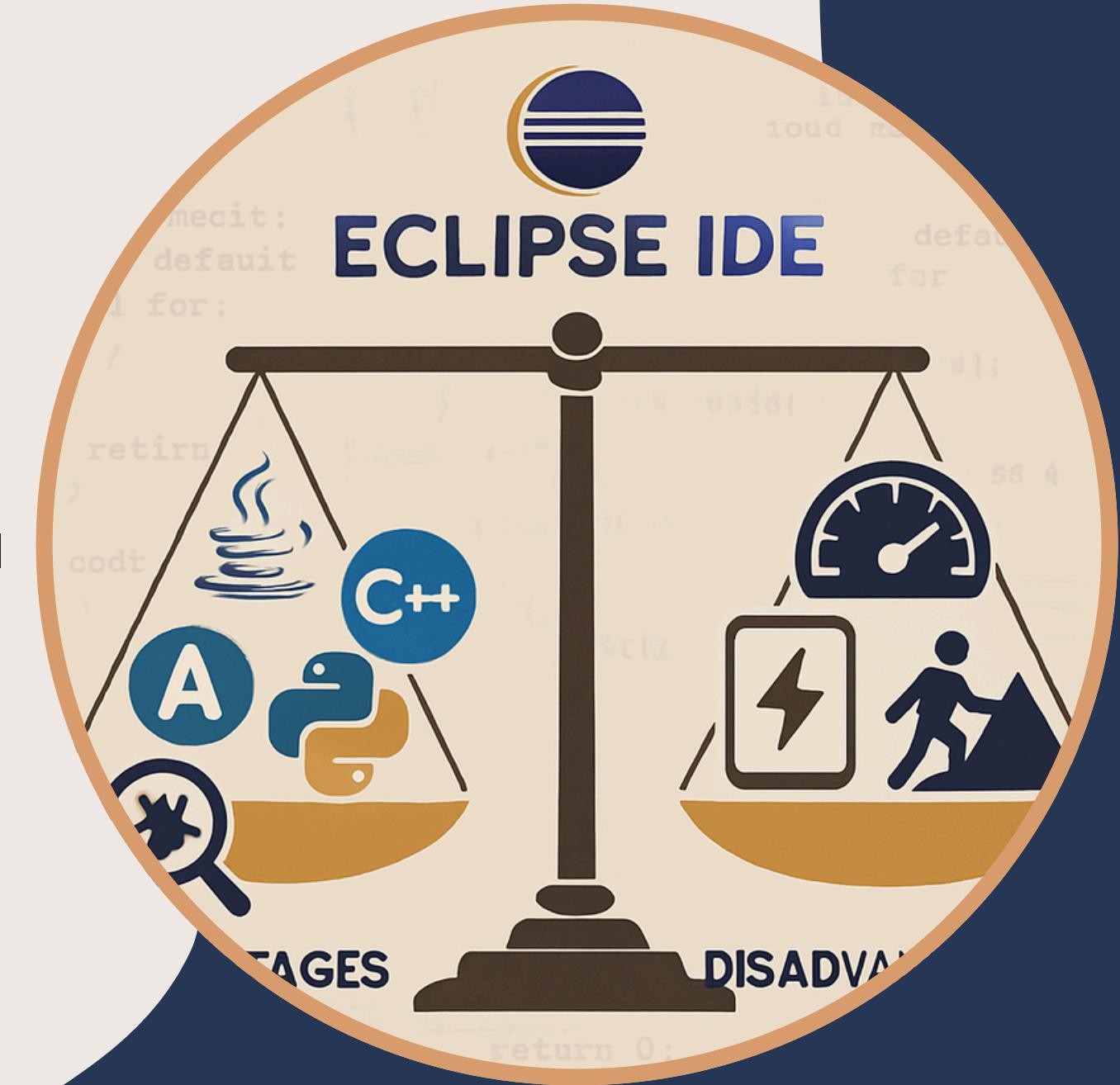
**Integración con Servicios en la Nube:**  
Plugins para interactuar con AWS, Azure o Google Cloud.

# VALORACION PERSONAL

## VENTAJAS

- Extensibilidad: Personalizable con plugins para añadir funcionalidades.
- Soporte Multilenguaje: Compatible con Java, C++, PHP, Python, etc., mediante plugins.
- Herramientas Integradas: Incluye depuradores, perfiles y control de versiones.
- Comunidad Activa: Fuerte apoyo de usuarios y desarrolladores.
- Integración Sencilla: Facilita la conexión con servidores de aplicaciones y tecnologías web.
- Uso intensivo de recursos
- Curva de aprendizaje
- Interfaz compleja
- Problemas de rendimiento

## INCOVENIENTES



# RECOMENDACIÓN DE USO

- Ideal para: Desarrollo Java empresarial (Java EE, servidores de aplicaciones).
- Fortaleza: Proyectos que exigen alta personalización e integración de herramientas específicas.
- Versatilidad: Útil para equipos que gestionan múltiples lenguajes bajo una misma plataforma.
- Accesibilidad: Excelente opción gratuita para proyectos académicos o de software libre.

