



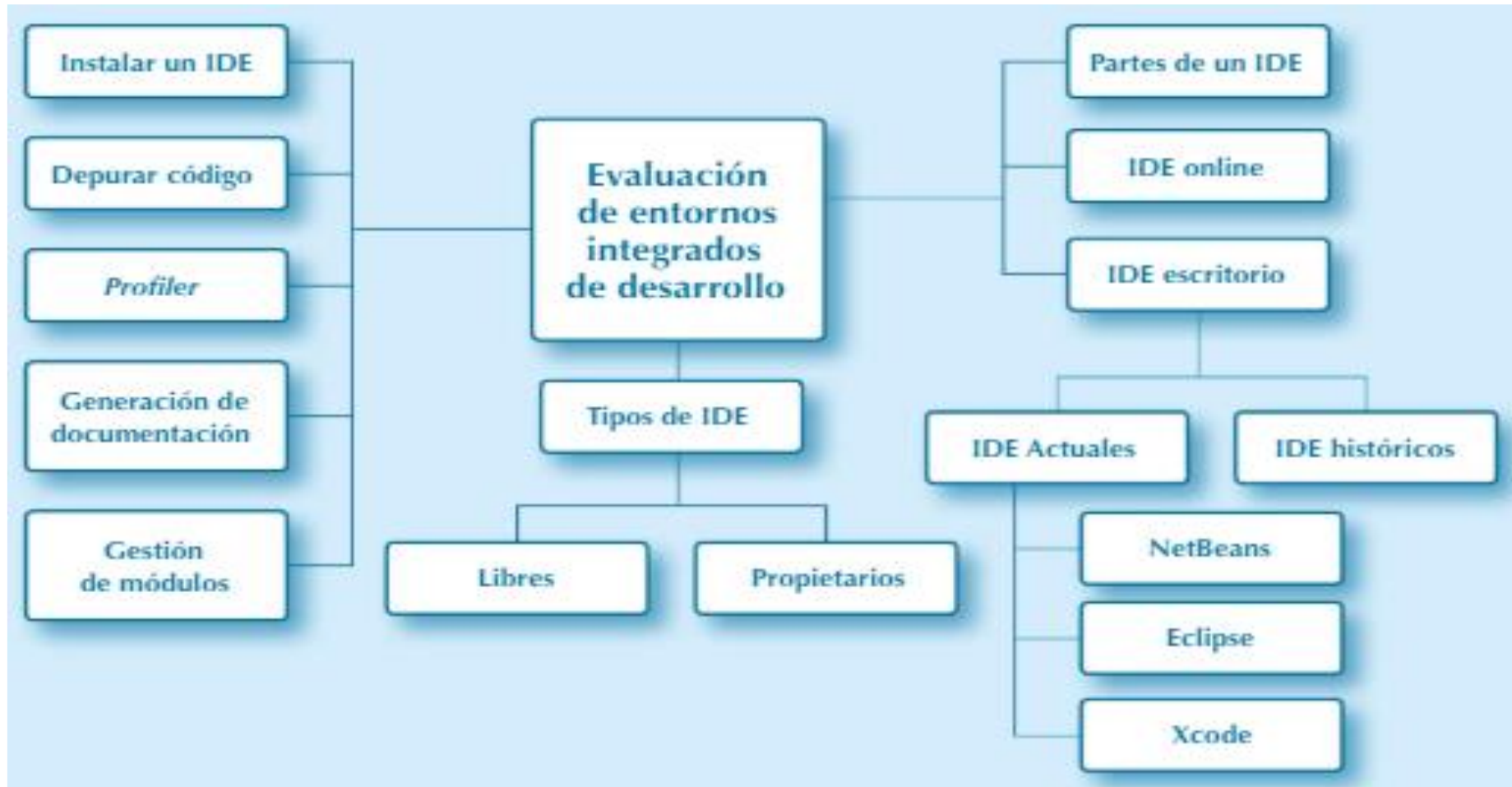
UNIDAD 2. EVALUACIÓN DE ENTORNOS DE DESARROLLO INTEGRADOS

VICENT MARTÍ

OBJETIVOS

- Aprender a instalar y configurar un IDE.
- Conocer los IDE más utilizados y como han ido evolucionando.
- Aprender a depurar el código utilizando un IDE.
- Conocer como generar documentación de forma automática y ampliar las capacidades del IDE mediante plugins.

MAPA CONCEPTUAL



CONTENIDO

1. Introducción
2. Evolución de los entornos de desarrollo
3. Entornos de desarrollo actuales
4. Entornos de desarrollo en la nube
5. Entornos de desarrollo libres y propietarios
6. Funciones de un IDE
7. Estructura de un IDE
8. Instalar un IDE
9. Configurar un IDE

1. INTRODUCCIÓN

Los entornos de desarrollo son las herramientas con las cuales los programadores crean aplicaciones. En entornos profesionales suelen utilizarse un IDE (Entorno de Desarrollo Integrado). Los IDE están diseñados para maximizar la productividad del programador proporcionando componentes muy unidos con interfaces de usuario similares. Los IDE presentan un único programa en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de software.

Un IDE consta de las siguientes herramientas:

- **Editor:** editores que colorean la sintaxis para ayudar al programador a detectar errores más fácilmente.
- **Compilador e intérprete:** Dependiendo del tipo de lenguaje utilizado para generar el código ejecutable.
- **Depurador (intérprete):** Un buen depurador requiere tener un intérprete detrás para ejecutar ordenes paso a paso, inspeccionar las variables, etc.
- **Constructor de interfaces gráficos:** Para que el desarrollador pueda crear ventanas, botones, campos de texto, tablas, pestañas, etc.

2. EVOLUCIÓN DE LOS ENTORNOS DE DESARROLLO

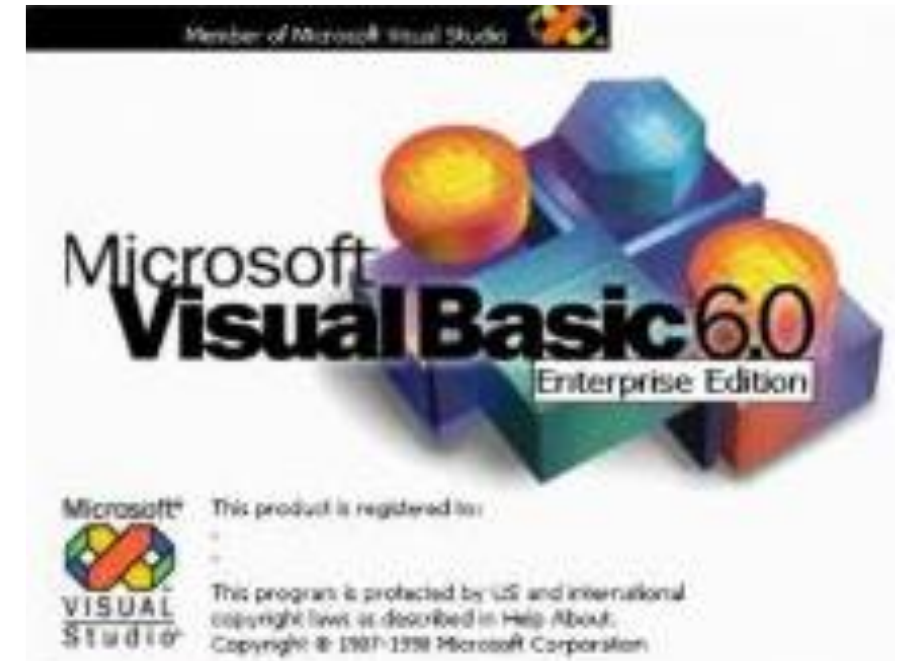
Los primeros sistemas no podían utilizar IDE, porque los programas eran preparados usando diagramas de flujo, introduciendo programas con tarjetas agujeradas (o papel cartón, etcétera) antes de enviarlos a un compilador. Dartmouth BASIC fue el primer lenguaje en ser creado con un IDE (también fue el primero en ser diseñado para ser utilizado enfrente de la consola o la terminal). Este IDE (parte de *Dartmouth Time Sharing System*) fue basado en código y basado en comandos, y por esto no se parecía mucho a los IDE gráficos actuales. Sin embargo, la edición integrada, manejo de archivos, compilación, depurador y ejecutable en una manera consistente con los IDE modernos.

- **Maestro I:** es un producto de Softlab Múnich y fue el primer sistema de desarrollo integrado IDE, para software, creado en 1975. Maestro I fue instalado por 22.000 programadores en todo el mundo.
- **Turbo Pascal:** lo lanzó la empresa Borland en 1983, fue el más potente en su época y se adaptó a diferentes plataformas como MS-DOS, CP/M, Macintosh y WINDOWS. Tras este éxito Borland lanzó nuevas herramientas como Delphi basadas en Pascal.



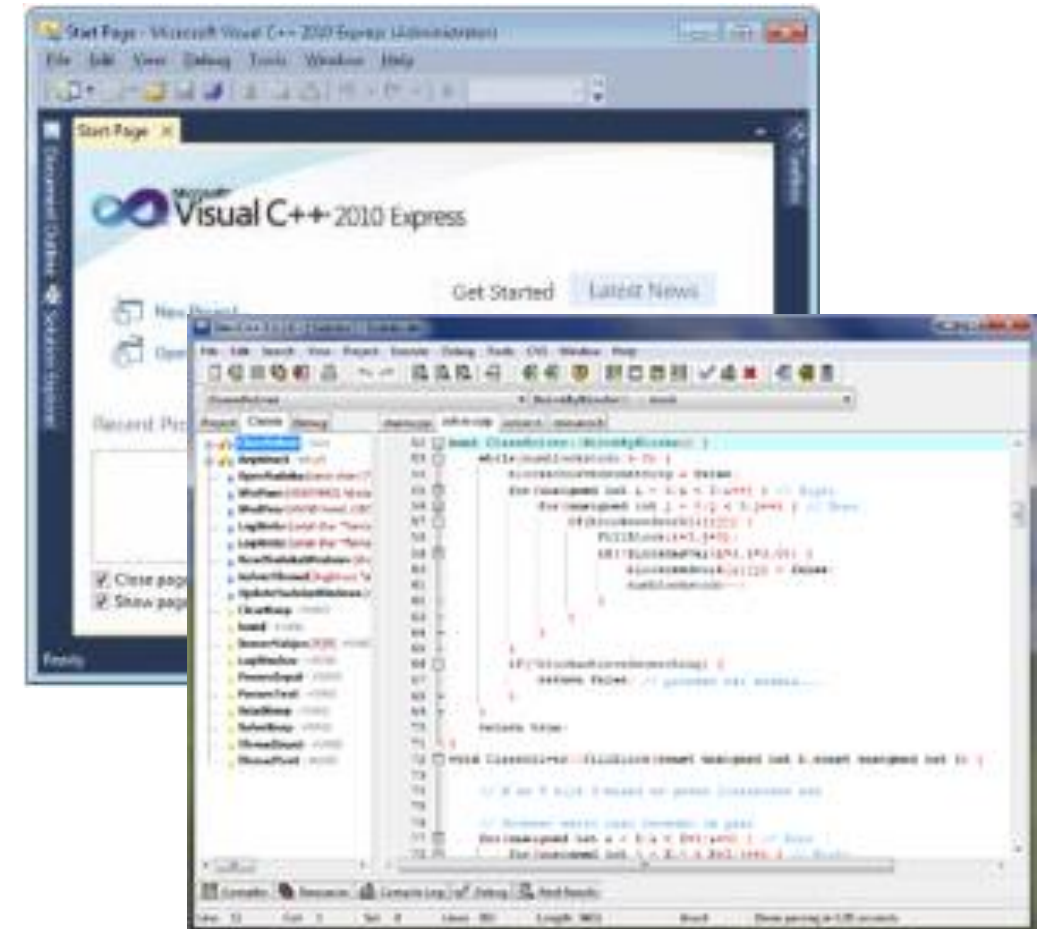
2. EVOLUCIÓN DE LOS ENTORNOS DE DESARROLLO II

- **Visual Basic 6:** Creado por en los años 90 por Microsoft, se estableció un nuevo paradigma de desarrollo llamado RAD (Rapid Application Development), Desarrollo Rápido de Aplicaciones. Primero se desarrollan los interfaces que se consensuan con el usuario y posteriormente se desarrolla el código y la base de datos. En la actualidad se utiliza un dialecto suyo VBA (Visual Basic for Applications) en las macros de Office. Visual Basic evolucionó para integrar la plataforma .NET; allí perdió su propia identidad como lenguaje único adquirible, pasando a integrar un paquete de productos, llamado precisamente Microsoft .NET; dentro de ese paquete o framework se encuentra el nuevo y llamado Visual Basic .NET, que trabaja sobre el entorno Microsoft Visual Studio. Esta nueva versión del lenguaje posee profundas diferencias en la forma de programar respecto de Visual Basic 6, pero gran semejanza en su sintaxis básica.



2. EVOLUCIÓN DE LOS ENTORNOS DE DESARROLLO II

- **Visual C++:** Es un IDE de Microsoft, para desarrollar en C y C++, su potencia radica en incluir las librerías de Windows, las Microsoft Foundation Classes (MFC) y framework .NET. Además permite añadir otras librerías externas como: DirectX, wxWidgets o Simple DirectMedia Layer (SDL). Su última versión estable es Visual Studio 2013.
- **Dev-C++** es un IDE para programar en C y C++, Usa MinGW, que es una versión de GCC (GNU Compiler Collection) como su compilador. Dev-C++ puede además ser usado en combinación con Cygwin y cualquier compilador basado en GCC. El Entorno está desarrollado en el lenguaje Delphi de Borland. Tiene una página de paquetes opcionales para instalar, con diferentes bibliotecas de código abierto. La última versión disponible es de 2015.



3. ENTORNOS DE DESARROLLO ACTUALES

El uso de los entornos integrados de desarrollo se ratifica y afianza en los 90 y hoy en día contamos con infinidad de IDE, tanto de licencia libre como no.

No hay unos entornos de desarrollo más importantes que otros. La elección del IDE más adecuado dependerá del lenguaje de programación que vayamos a utilizar para la codificación de las aplicaciones y el tipo de licencia con la que queramos trabajar.

Tabla de los IDE más relevantes hoy en día:

Entorno de desarrollo	Lenguajes que soporta	Tipo de licencia
NetBeans	C/C++, Java, JavaScript, PHP, Python.	De uso público.
Eclipse	Ada, C/C++, Java, JavaScript, PHP.	De uso público.
Microsoft Visual Studio.	Basic, C/C++, C#.	Propietario.
C++ Builder.	C/C++.	Propietario.
JBuilder.	Java.	Propietario.

NETBEANS

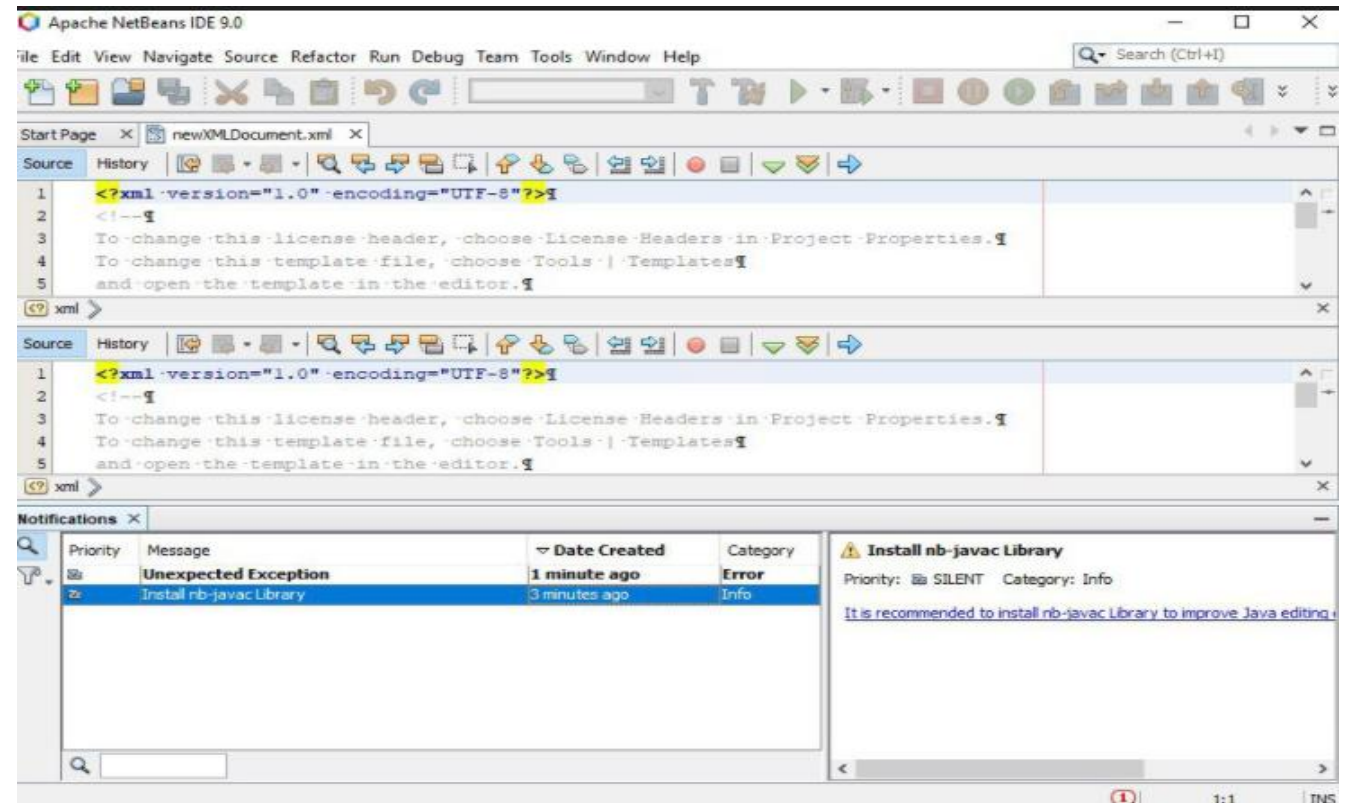


Está escrito en Java, es multiplataforma, aunque se creó para desarrollos Java, puede utilizarse para Python, PHP, HTML5, C/C++, etc.

Es open source y es modular, módulos que pueden añadirse según se necesiten.

Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation). Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia

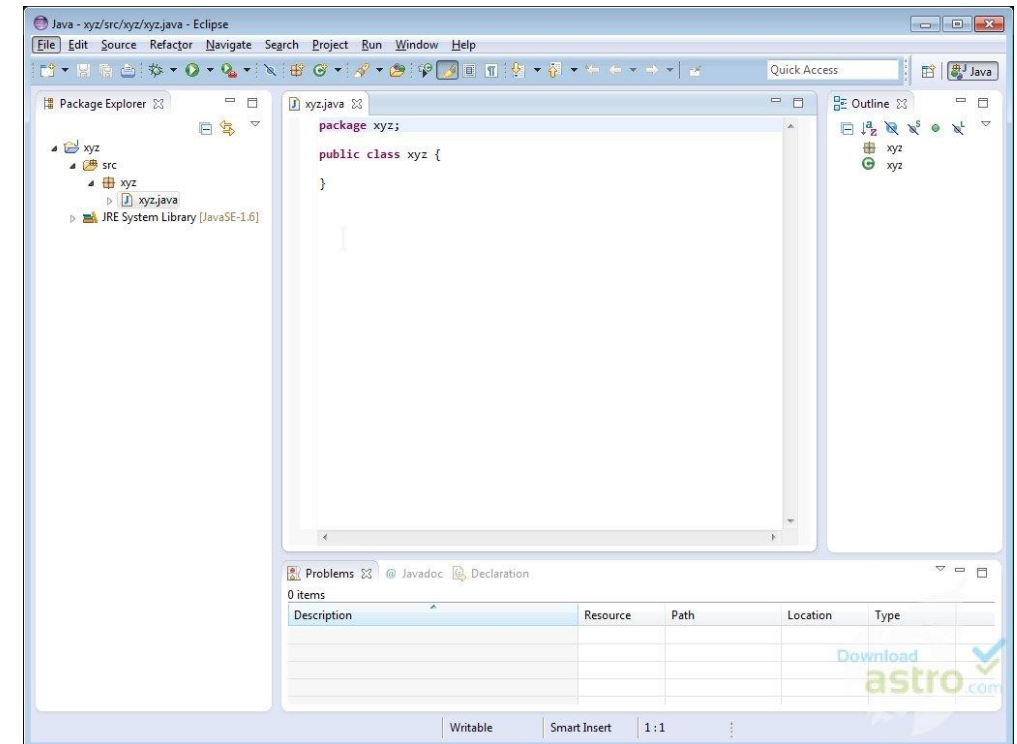
basada en la Mozilla Public License (MPL). En octubre de 2007, Sun anunció que NetBeans desde entonces se ofrecerá bajo licenciamiento dual de Licencia CDDL y la GPL versión 2.



ECLIPSE



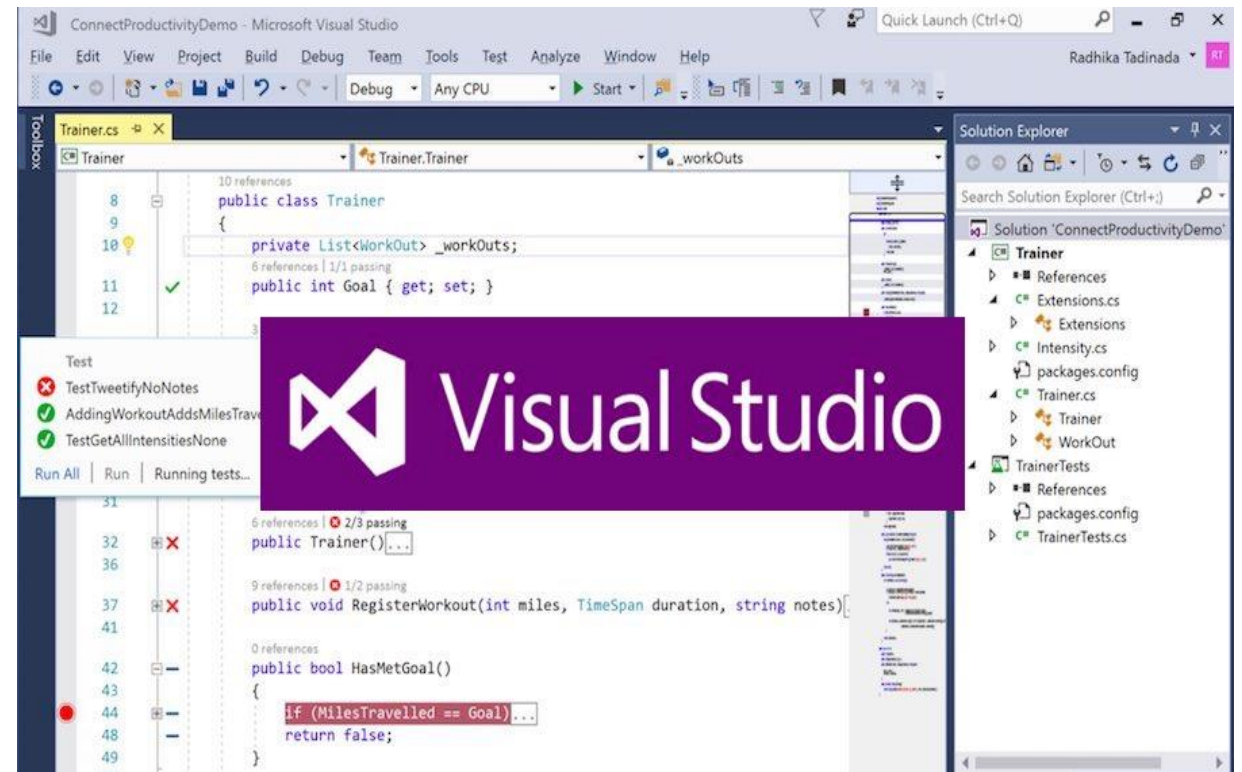
Es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar IDE, como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse. Fue desarrollado por IBM como el sucesor de su familia de herramientas para Visual Age. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto. Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero incompatibles con GNU GPL.



MICROSOFT VISUAL STUDIO

IDE para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, .NET, F#, Java, Python, Ruby, PHP, Visual Basic, etc.

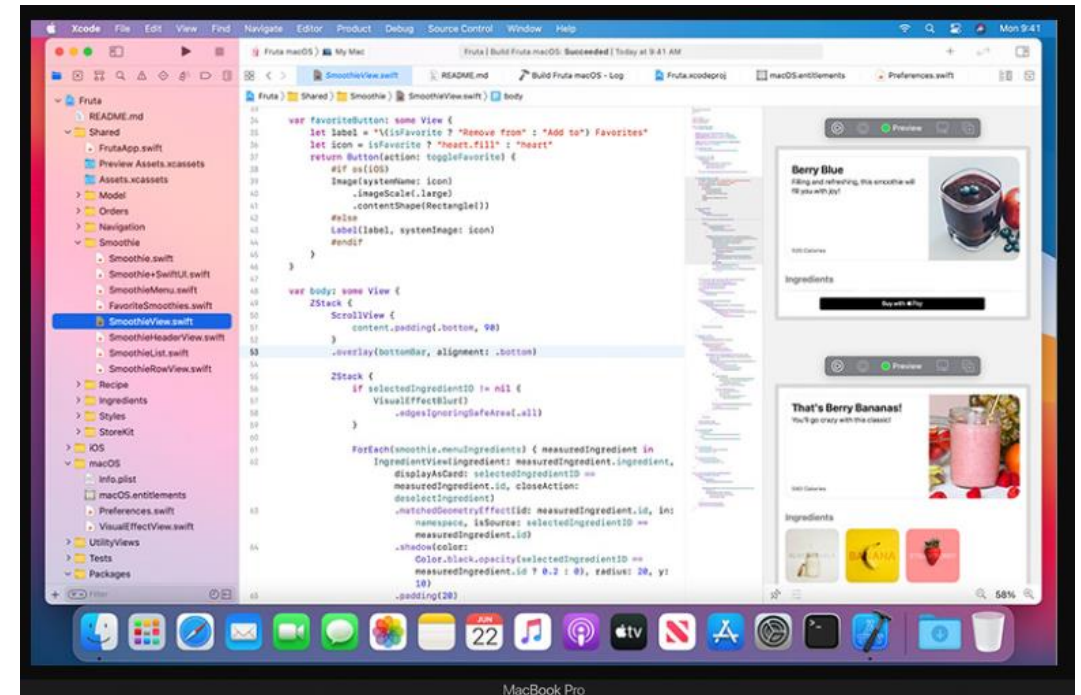
También para entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Monaco. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.



XCODE

Es un IDE para macOS que contiene un conjunto de herramientas creadas por Apple destinadas al desarrollo de software para macOS, iOS, watchOs y tvOS. El entorno gráfico se complementa con herramientas de línea de comando. Su primera versión tiene origen en el año 2003 y actualmente su versión número 11 se encuentra disponible de manera gratuita en el Mac App Store o mediante descarga directa desde la página para desarrolladores de Apple. Xcode trabaja conjuntamente con Interface Builder, una herencia de NeXT, una herramienta gráfica para la creación de interfaces de usuario. Incluye la colección de compiladores del proyecto GNU (GCC), y puede compilar C, C++, Swift, Objective-C, ObjectiveC++, Java y AppleScript.

Entre las características más apreciadas de Xcode está la tecnología para distribuir el proceso de construcción a partir de código fuente entre varios ordenadores, utilizando Bonjour.



4. ENTORNOS DE DESARROLLO EN LA NUBE

Los entornos de desarrollo en la nube o online se extienden cada vez mas, su principal desventaja es la potencia, pero facilitan el trabajo colaborativo, los repositorios comunes y facilidades para trabajar con cualquier dispositivo.

- **Cloud9**
- **Codeanywhere**
- **Eclipse Che**
- **Koding**
- **ShiftEdit**
- **Orión**
- **Python Fiddle**



EJERCICIO 1



Escribe un documento y súbelo al aula virtual (AV)

- Analiza las diferentes características de los IDE en la NUBE, indica para que uso es más adecuado cada uno.

5. ENTORNOS DE DESARROLLO LIBRES Y PROPIETARIOS

Entornos Integrados Libres: Son aquellos con licencia de uso público. No hay que pagar por ellos, y aunque los más conocidos y utilizados son Eclipse y NetBeans, hay bastantes más.

IDE	Lenguajes que soporta	Sistema Operativo
NetBeans.	C/C++, Java, JavaScript, PHP, Python.	Windows, Linux, Mac OS X.
Eclipse.	Ada, C/C++, Java, JavaScript, PHP.	Windows, Linux, Mac OS X.
Gambas.	Basic.	Linux.
Anjuta.	C/C++, Python, Javascript.	Linux.
Geany.	C/C++, Java.	Windows, Linux, Mac OS X.
GNAT Studio.	Fortran.	Windows, Linux, Mac OS X.

Entornos Integrados Proprietarios: Son aquellos entornos integrados de desarrollo que necesitan licencia. No son free software, hay que pagar por ellos. El más conocido y utilizado es Microsoft Visual Studio, que usa el framework .NET y es desarrollado por Microsoft.

IDE	Lenguajes que soporta	Sistema Operativo
Microsoft Visual Studio.	Basic, C/C++, C#.	Windows.
FlashBuilder.	ActionScript.	Windows, Mac OS X.
C++ Builder.	C/C++.	Windows.
Turbo C++ profesional.	C/C++.	Windows.
JBuilder.	Java.	Windows.
JCreator.	Java.	Windows, Linux, Mac OS X.
Xcode.	C/C++, Java.	Mac OS X.

5. ENTORNOS DE DESARROLLO LIBRES Y PROPIETARIOS II

- El aspecto de la licencia del IDE que se elija para el desarrollo de un proyecto es una cuestión de vital importancia. En su elección prevalecerá la decisión de los supervisores del proyecto y de la dirección de la empresa.
- En el siguiente enlace encontrarás un documento muy interesante, en inglés, donde se detallan los principales entornos de desarrollo existentes en la actualidad con todas sus características: licencias, sistemas operativos donde pueden ser instalados y configurados, lenguajes que soporta, desarrolladores y última versión estable.

http://en.wikipedia.org/wiki/Integrated_development_environment

6. FUNCIONES DE UN IDE

- Editor de código: coloración de la sintaxis.
- Auto-completado de código, atributos y métodos de clases.
- Identificación automática de código.
- Herramientas de concepción visual para crear y manipular componentes visuales.
- Asistentes y utilidades de gestión y generación de código.
- Archivos fuente en unas carpetas y compilados a otras.
- Compilación de proyectos complejos en un solo paso.
- Control de versiones: tener un único almacén de archivos compartido por todos los colaboradores de un proyecto.
- Ante un error, mecanismo de auto-recuperación a un estado anterior estable.
- Soporta cambios de varios usuarios de manera simultánea.
- Generador de documentación integrado.
- Detección de errores de sintaxis en tiempo real.

6. FUNCIONES DE UN IDE II

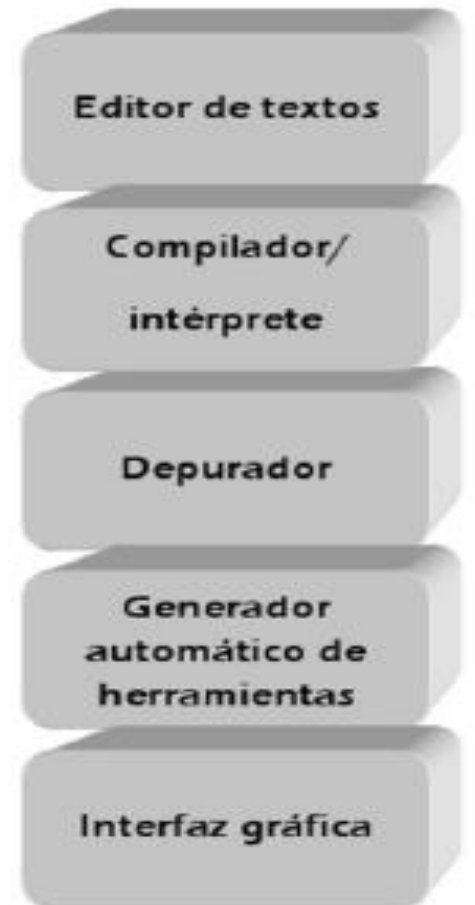
Otras funciones importantes son:

- Ofrece refactorización de código: cambios menores en el código que facilitan su legibilidad sin alterar su funcionalidad (por ejemplo cambiar el nombre a una variable).
- Permite introducir automáticamente tabulaciones y espaciados para aumentar la legibilidad.
- Depuración: seguimiento de variables, puntos de ruptura y mensajes de error del intérprete.
- Aumento de funcionalidades a través de la gestión de sus módulos y plugins.
- Administración de las interfaces de usuario (menús y barras de herramientas).
- Administración de las configuraciones del usuario.

7. ESTRUCTURA DE UN IDE

La funcionalidades descritas anteriormente, quedan estructuradas en los siguientes componentes software:

- **Editor de textos:** Se resalta y colorea la sintaxis, autocompletar código, ayuda y listado de parámetros de funciones y métodos de clase. Inserción automática de paréntesis, corchetes, tabulaciones y espaciados.
- **Compilador/intérprete:** Detección de errores de sintaxis en tiempo real. Características de refactorización.
- **Depurador:** Botón de ejecución y traza, puntos de ruptura y seguimiento de variables. Opción de depurar en servidores remotos.
- **Generador automático de herramientas:** Para la visualización, creación y manipulación de componentes visuales y utilidades de gestión y generación código.
- **Interfaz gráfica:** Nos brinda la oportunidad de programar en varios lenguajes con un mismo IDE. Diversas bibliotecas y plugins.



8. INSTALAR UN IDE

- Vamos a realizar la instalación de NetBeans, en su versión 12 (LTS) (soporte de larga duración) sobre Windows 10. Podemos utilizar otras versiones pero nos aseguramos que sean compatibles.
- Primero requiere la instalación previa del JDK (versión 8 o superior, recomiendo la 11 que es la LTS) que sea compatible con la versión de NetBeans que se quiera instalar. Algunas versiones requieren registrarse en Oracle, pero no supone ningún problema.
- Podemos comprobar si tenemos la versión de la máquina java instalada, para ello en la pantalla de comandos (cmd): `java -version`. Con `javac -version` vemos la versión del JDK.
- JDK (Java development kit o kit de desarrollo de Java) necesario para programar, pero para ejecutar es suficiente con el JRE (Java Runtime Environment o Entorno Java de Ejecución) que está en el JVM (Java Virtual Machine o Máquina Virtual de Java) cada sistema operativo tendrá una JVM diferente.

INSTALACIÓN DE NETBEANS EN WINDOWS

- A continuación vamos a la página web de Oracle para iniciar la descarga del JDK.(1)
<https://www.oracle.com/java/technologies/javase-downloads.html>
- Seleccionamos el sistema operativo, si es de 32 o 64 bits (2). Seguimos el asistente con las opciones por defecto(3). Finalmente comprobamos como queda la versión instalada del JRE y del JDK (4).

1 Java SE 11 (LTS)
Java SE 11.0.8 is the latest release for the Java SE 11 Platform

- Documentation
- Installation Instructions
- Release Notes
- Oracle License
 - Binary License
 - Documentation License
- Java SE Licensing Information User Manual
 - Includes Third Party Licenses
- Certified System Configurations
- Readme

2 Windows x64 Installer
151.73 MB
jdk-11.0.8_windows-x64_bin.exe

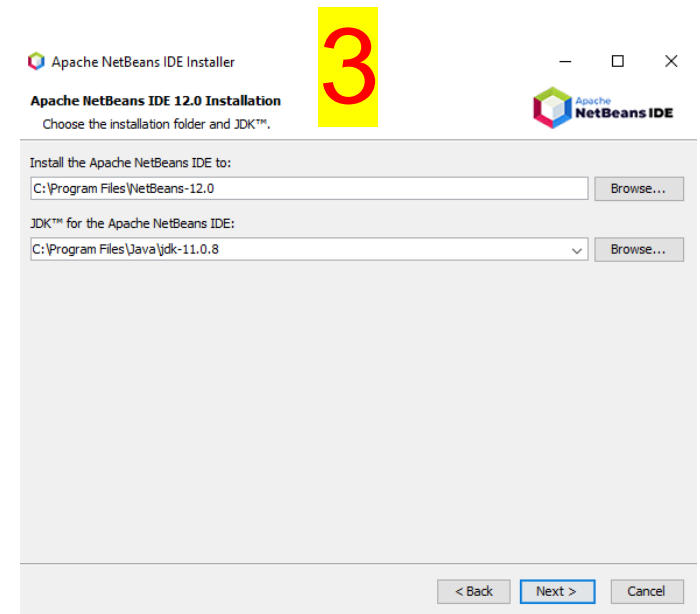
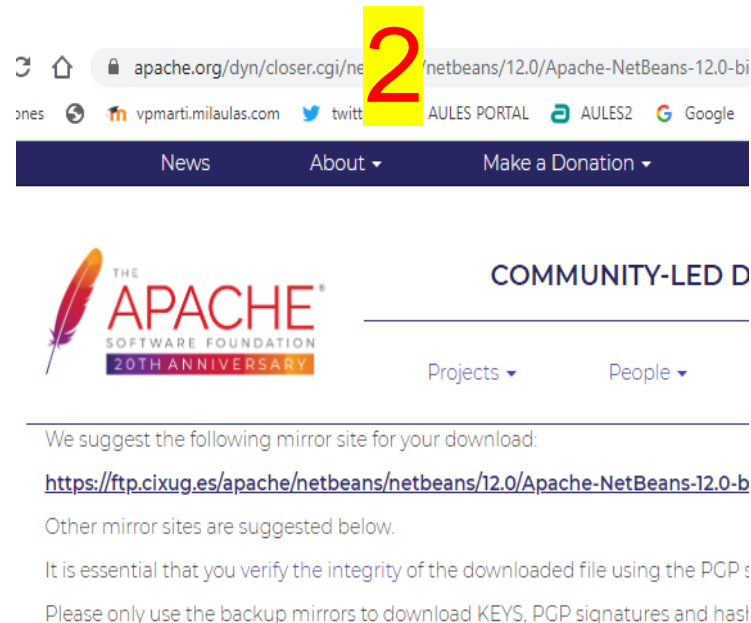
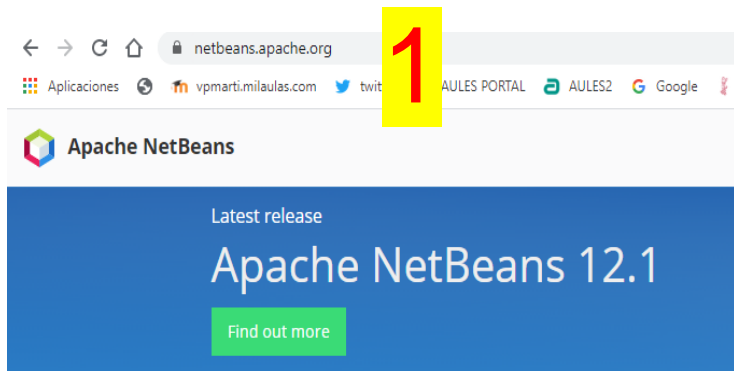
3 Java(TM) SE Development Kit 11.0.8 (64-bit) - Setup
Welcome to the Installation Wizard for Java SE Development Kit 11.0.8
This wizard will guide you through the installation process for the Java SE Development Kit 11.0.8.

4 Símbolo del sistema
Microsoft Windows [Versión 10.0.19041.508]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Vicent>java -version
java version "11.0.8" 2020-07-14 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.8+10-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.8+10-LTS, mixed mode)
C:\Users\Vicent>javac -version
javac 11.0.8

Next > Cancel

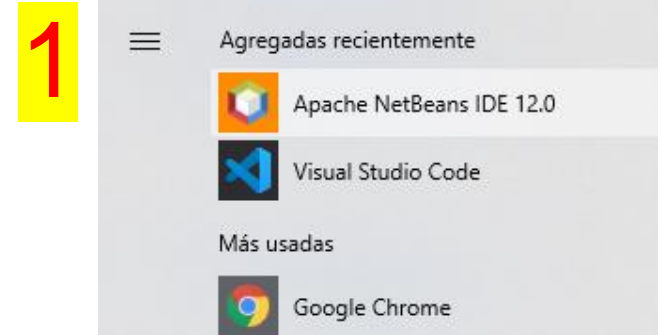
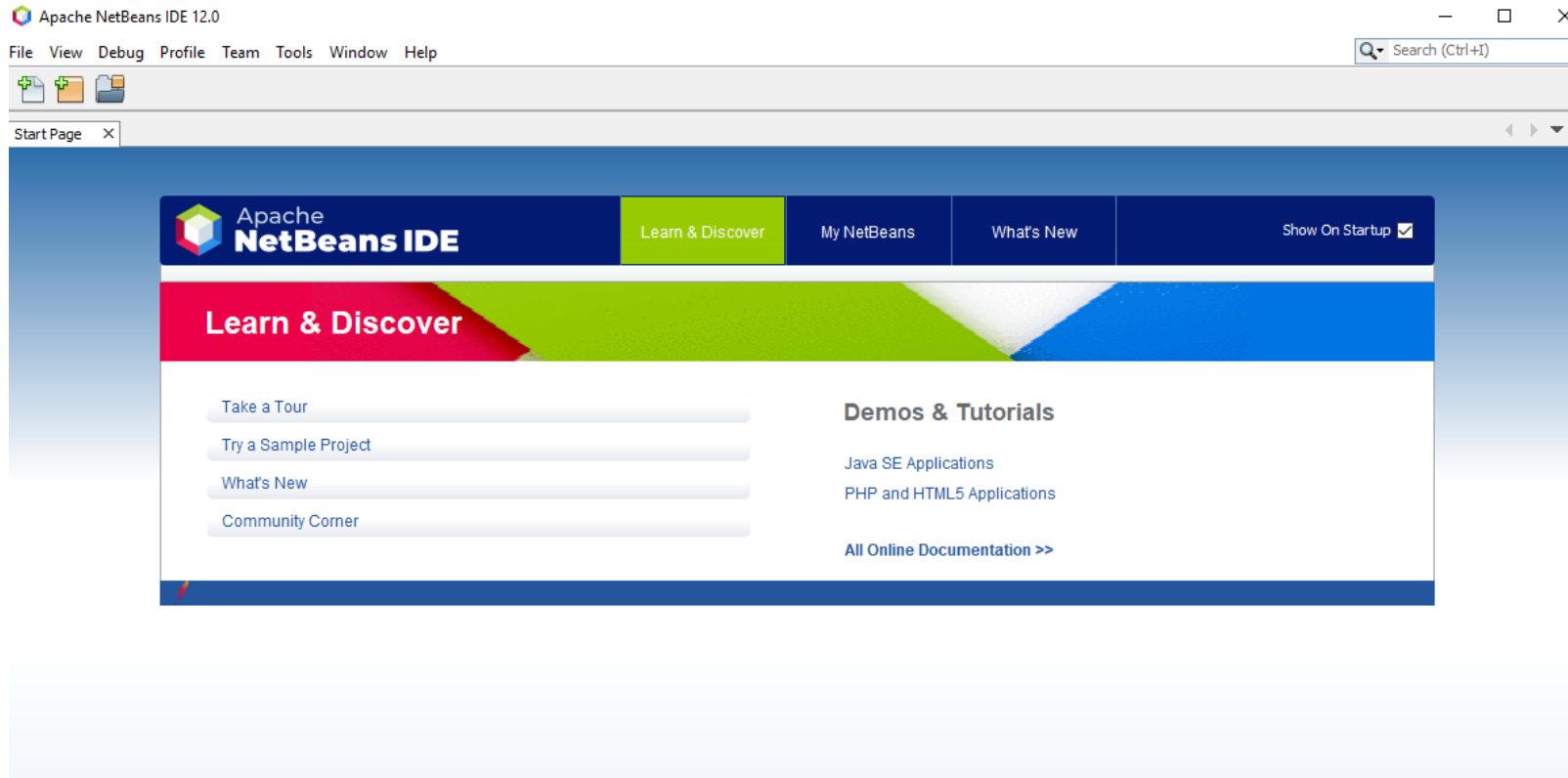
INSTALACIÓN DE NETBEANS EN WINDOWS II

- A continuación la instalación de NetBeans, (1) vamos a la pagina: <https://netbeans.apache.org/>
- Vamos a download y descargamos (2), y vamos a realizar la instalación de NetBeans (3), en su versión 12 (LTS) (soporte de larga duración) sobre Windows 10.



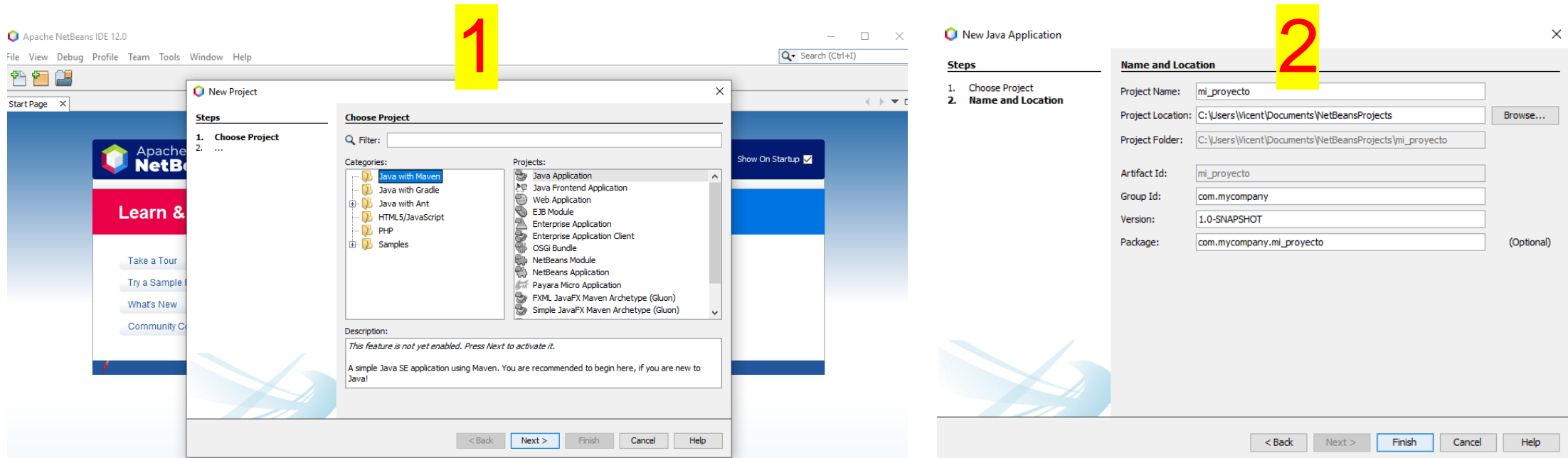
INSTALACIÓN DE NETBEANS EN WINDOWS III

- Una vez instalado podemos buscar en el menú de Windows (1) o el icono en el escritorio y procedemos a entrar:



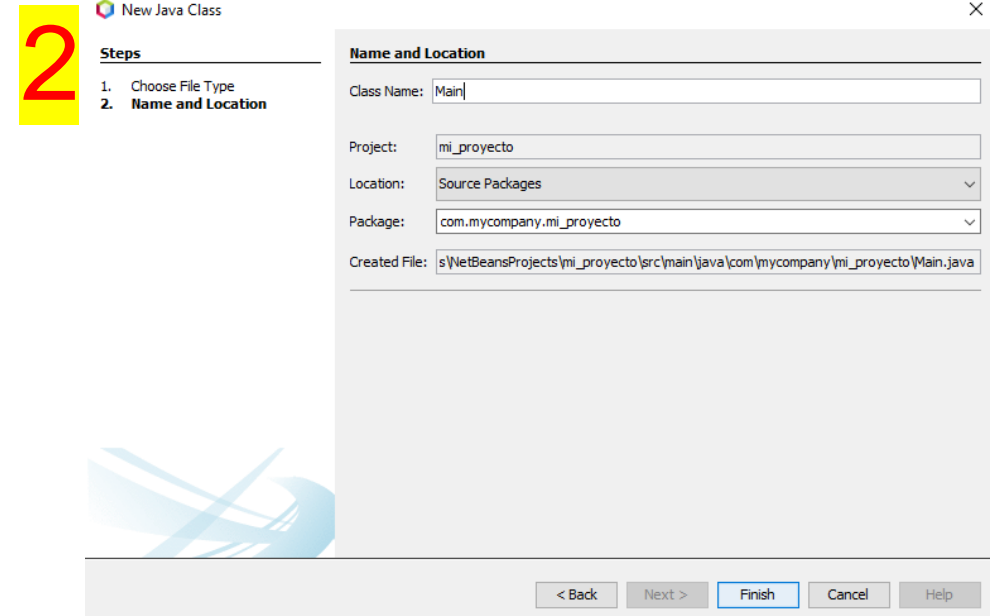
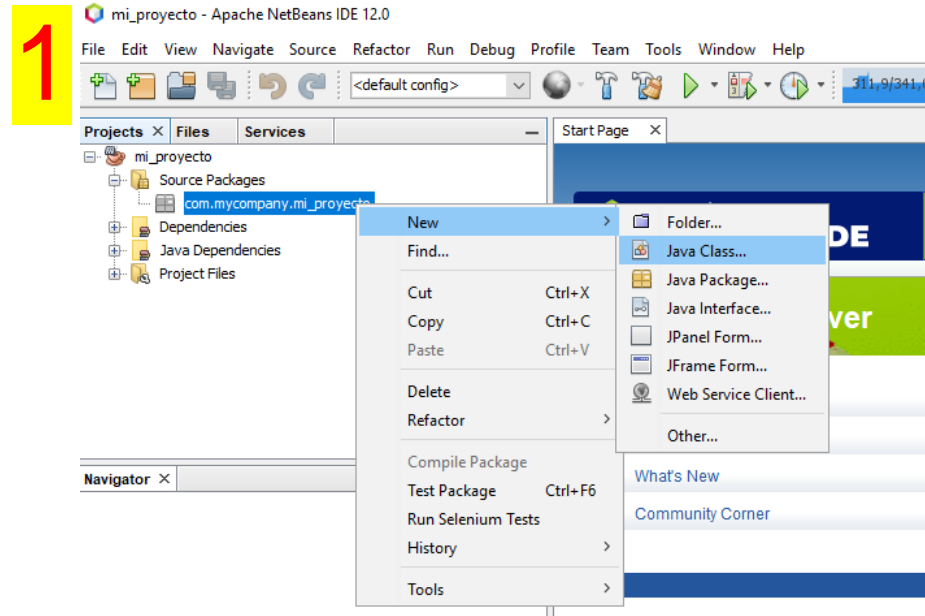
INSTALACIÓN DE NETBEANS EN WINDOWS IV

- Para ver que todo esta correcto crearemos un nuevo proyecto. Entramos a **File>New Project>Java with Maven>Java Application** y pulsamos **<Next>** (1) y aceptamos la licencia y todo por defecto hasta terminar. Hemos activado Java SE.
- Pondremos el nombre del proyecto: (Por ejemplo: Mi_proyecto, como vemos en la pantalla (2):



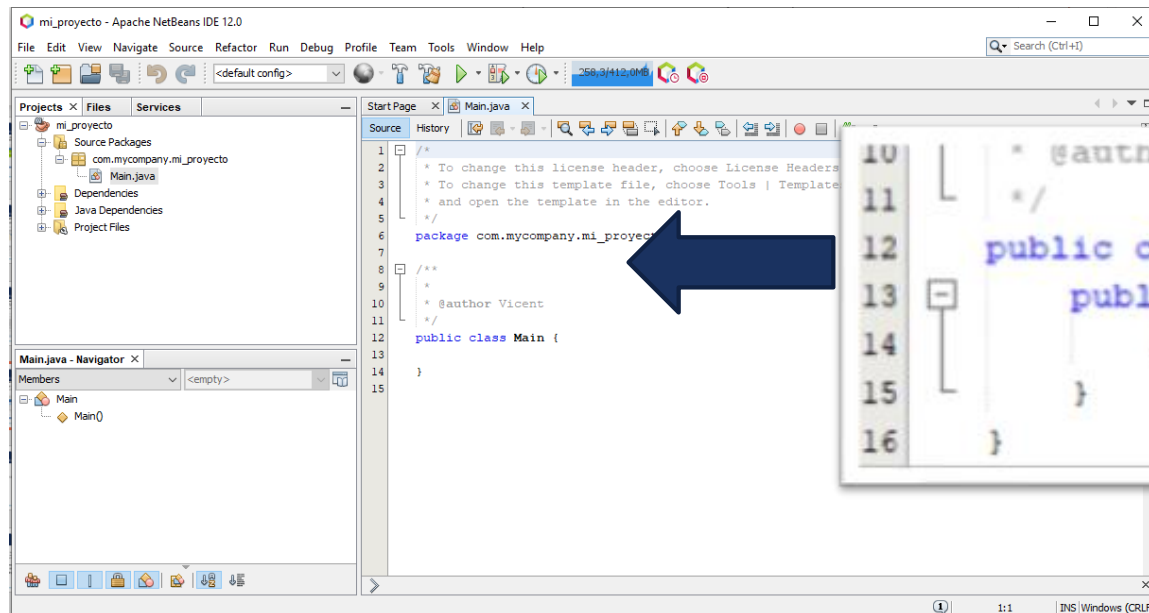
INSTALACIÓN DE NETBEANS EN WINDOWS V

- Una vez finalice el proceso, procederemos a crear una clase Java, llamada Main. Entramos a Mi_proyecto y con **<clic derecho>** seleccionamos **New>Java Class**. (1)
- Pondremos el nombre **Main** y **<Finish>** (2)



INSTALACIÓN DE NETBEANS EN WINDOWS VI

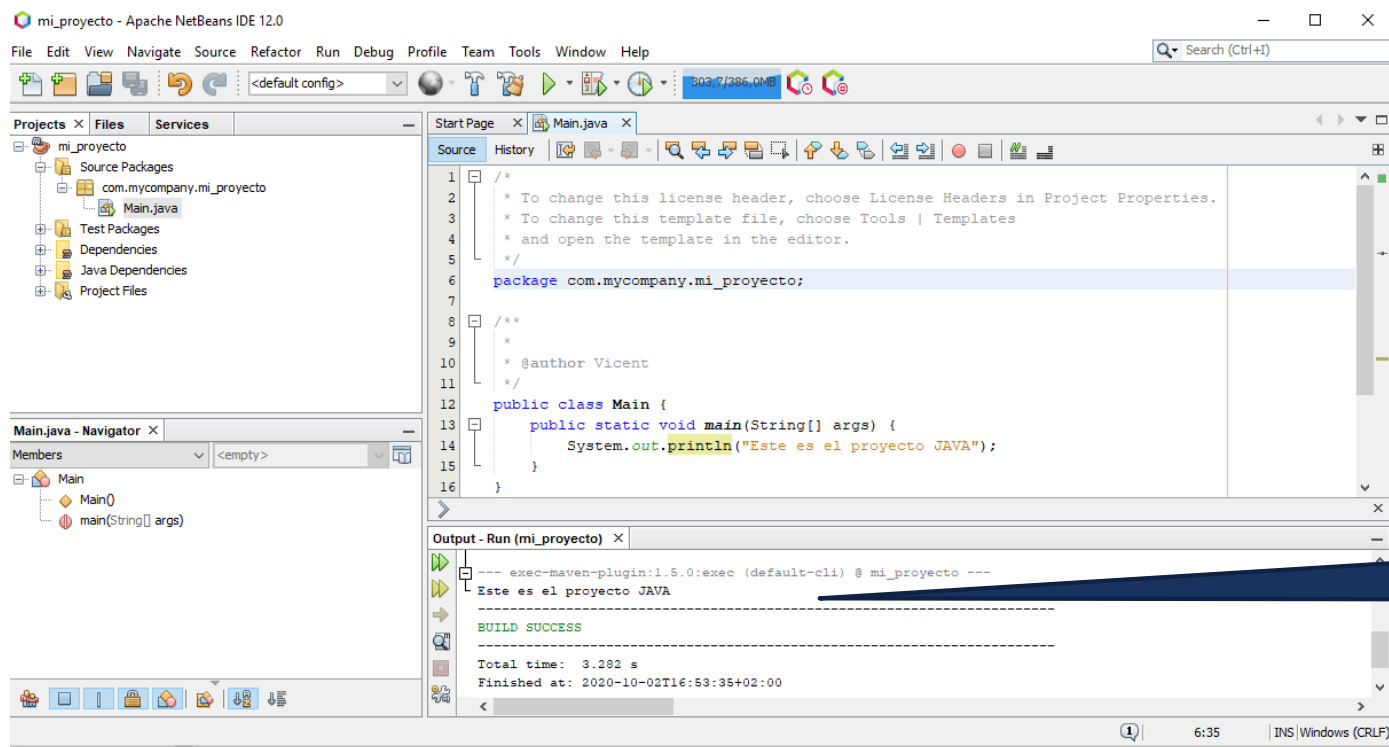
- Tenemos un código fuente preparado para completar con el código Java que queramos.
- Simplemente lo podemos completar el método Main haciendo que escriba por pantalla.
- Hay teclas que nos ayudan a manejarnos mejor, podemos ver como si escribes psvm **<control><space>** te propone el nombre public static void main(String[] args) { }, que es lo que pretendemos escribir, puedes ver en el menu superior **"Source"** con diferentes opciones.



Código fuente que introducimos

INSTALACIÓN DE NETBEANS EN WINDOWS VII

- Guardamos **File>Save**, el programa y los ejecutamos **<clic derecho>, >Run file**. La primera vez tardará mucho, pero la segunda vez que ejecutamos será más rápido.



Texto que escribe en la pantalla al ejecutar

EJERCICIO 2



- Instala en una máquina virtual de Windows 10 el Netbeans y crea un pequeño programa como el del enlace.

<https://www.youtube.com/watch?v=L3wsgXRtuQM>

Escribe un documento con portada, índice, url consultadas, etc.

- Los pasos básicos de la instalación. Paso a paso con pantallazos y la explicación de los comandos utilizados.
- Incluye que funcionalidades del IDE se pueden utilizar para agilizar la escritura del programa.
- Súbelo al aula virtual (AV).

INSTALACIÓN DE NETBEANS EN LINUX(UBUNTU)

- A continuación la instalación de NetBeans en Linux (Ubuntu).
- Vamos a realizar la instalación de NetBeans, en su versión 12 (LTS) (soporte de larga duración) sobre Linux (Ubuntu). Podemos utilizar diferentes distribuciones y versiones pero nos aseguramos que sean compatibles.
- Primero requiere la instalación previa del JDK (versión 8 o superior, recomendando la 11 que es la LTS) que sea compatible con la versión de NetBeans que se quiera instalar. Algunas versiones requieren registrarse en Oracle, pero no supone ningún problema.
- Podemos comprobar si tenemos la versión JDK instalada, para ello en la pantalla de comandos `javac -versión`.
- Podemos seguir diferentes pasos para la instalación.

INSTALACIÓN DE NETBEANS EN LINUX(UBUNTU) II

A continuación, lo vemos paso a paso:

- Primero necesitas instalar Java JDK de los repositorios estándar como se muestra.

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

```
$ sudo apt-get install default-jdk
```

- Si se quiere instalar el runtime sólo, en lugar del desarrollo utilizaremos: `$ sudo apt-get install default-jre`

A continuación, verifique el Java JDK.

```
$ javac -version
```

- En la página de NetBeans descargue el script de instalación (Apache-NetBeans-12.0-bin-linux-x64.sh)
Alternativamente, también puedes descargar utilizando la utilidad `wget` ingresando el siguiente comando:

```
$ wget -c https://downloads.apache.org/netbeans/netbeans/12.0/Apache-NetBeans-12.0-bin-linux-x64.sh
```

INSTALACIÓN DE NETBEANS EN LINUX(UBUNTU) III

- Una vez completada la descarga, navegue al directorio en el que se encuentra el directorio **NetBeans IDE** y de permisos de ejecución.

```
$ chmod +x Apache-NetBeans-12.0-bin-linux-x64.sh
```

```
$ ./Apache-NetBeans-12.0-bin-linux-x64.sh
```

- Después de ejecutar el script de instalación anterior, debe seguir las opciones y acepte la licencia.
- A continuación, habilite las actualizaciones automáticas para los complementos instalados utilizando la casilla de verificación en la siguiente pantalla que muestra el resumen de la instalación y haga clic en **Instalar**.
- Cuando finalice la instalación, reinicie el ordenador.

EJERCICIO 3



- Instala en una máquina virtual de Ubuntu el Netbeans y crea un pequeño programa que pinte por pantalla un texto.

Escribe una guía con portada, índice, url consultadas, etc.

- Paso a paso con pantallazos y la explicación de los comandos utilizados, súbelo al aula virtual (AV).

EJERCICIO 4

- Instala Eclipse en una máquina virtual de Windows 10 y crea el mismo programa del ejercicio 2. Puedes utilizar un tutorial, solo tienes que seguir los pasos.

<https://www.youtube.com/watch?v=lp23wIYgPPQ>

<https://www.youtube.com/watch?v=8c9LHF9iiKQ>

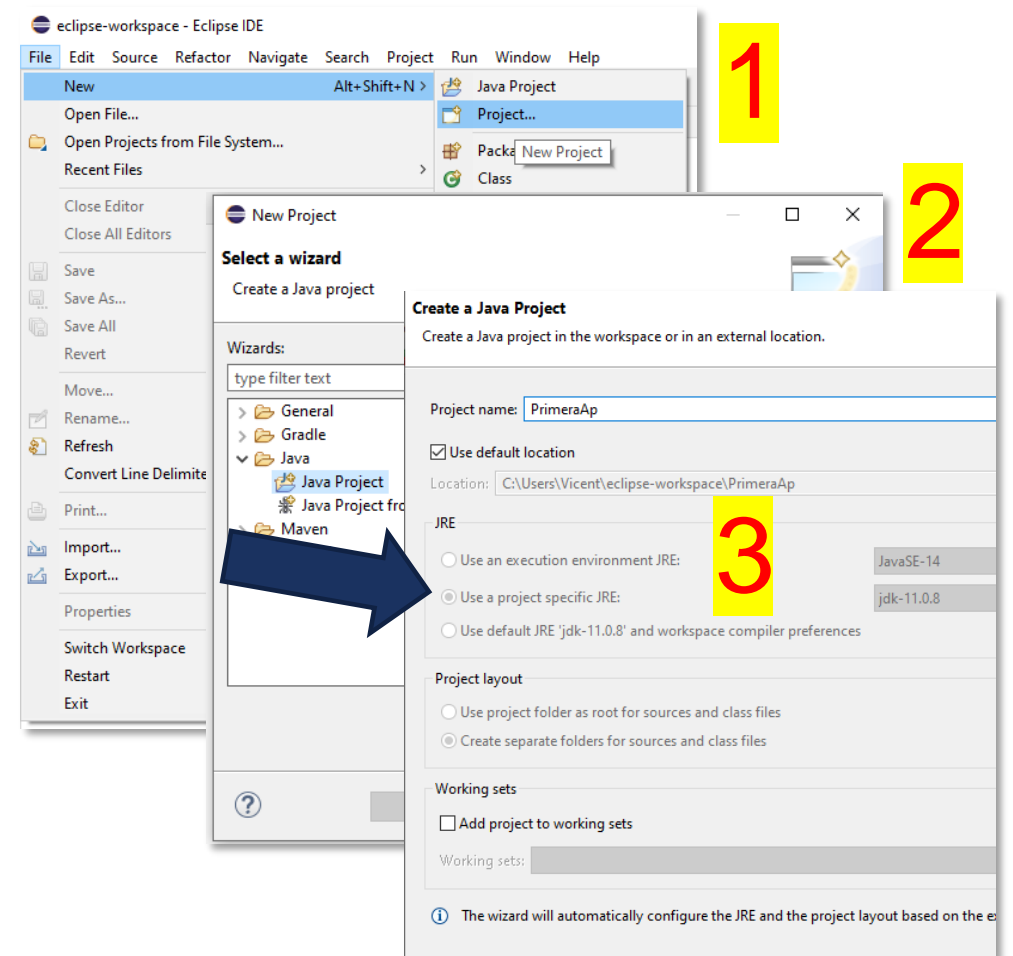
Escribe un documento con portada, índice, url consultadas, etc.

- Los pasos básicos de la instalación. Paso a paso con pantallazos y la explicación de los comandos utilizados. Recuerda que antes debes tener instalado Java JDK.
- Súbelo al aula virtual (AV).

The screenshot shows the Eclipse IDE installation interface. A blue speech bubble points to the 'eclipse.org/downloads/' URL in the browser address bar, containing the text 'La web de Eclipse'. Another blue speech bubble points to the 'Java 11+ VM' dropdown menu, containing the text 'Esta debe ser la versión de java que tienes ya instalada'. The interface includes a 'Download 64 bit' button and an 'INSTALL' button. The 'Installation Folder' is set to 'C:\Users\Vicent\eclipse\java-2020-09'. The 'Java 11+ VM' is set to 'C:\Program Files\Java\jdk-11.0.8'. The 'create start menu entry' and 'create desktop shortcut' options are checked.

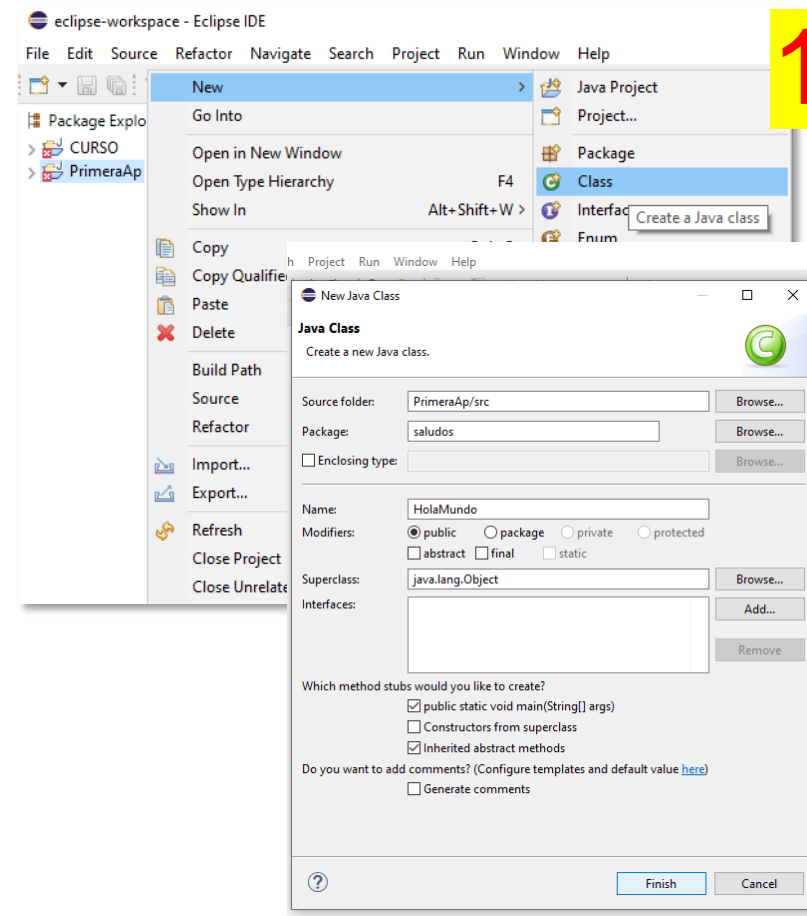
9. CONFIGURAR UN IDE

- Vamos a definir un mínimo proyecto en Java usando Eclipse
- Para crear un nuevo proyecto: aplicación / programa en Java con Eclipse
- pulsaremos en (1) **File>New>Project**
- Seleccionaremos el tipo de proyecto Java que queremos crear (2): “**Java Project**” y **<Next>**.
- Indicaremos el nombre del proyecto Java de Eclipse (3) (en nuestro caso “**PrimeraAp**”), indicaremos la carpeta donde queremos guardar el proyecto Eclipse. Dejamos todo por defecto. Nos aseguramos que el Java es el que tenemos instalado.



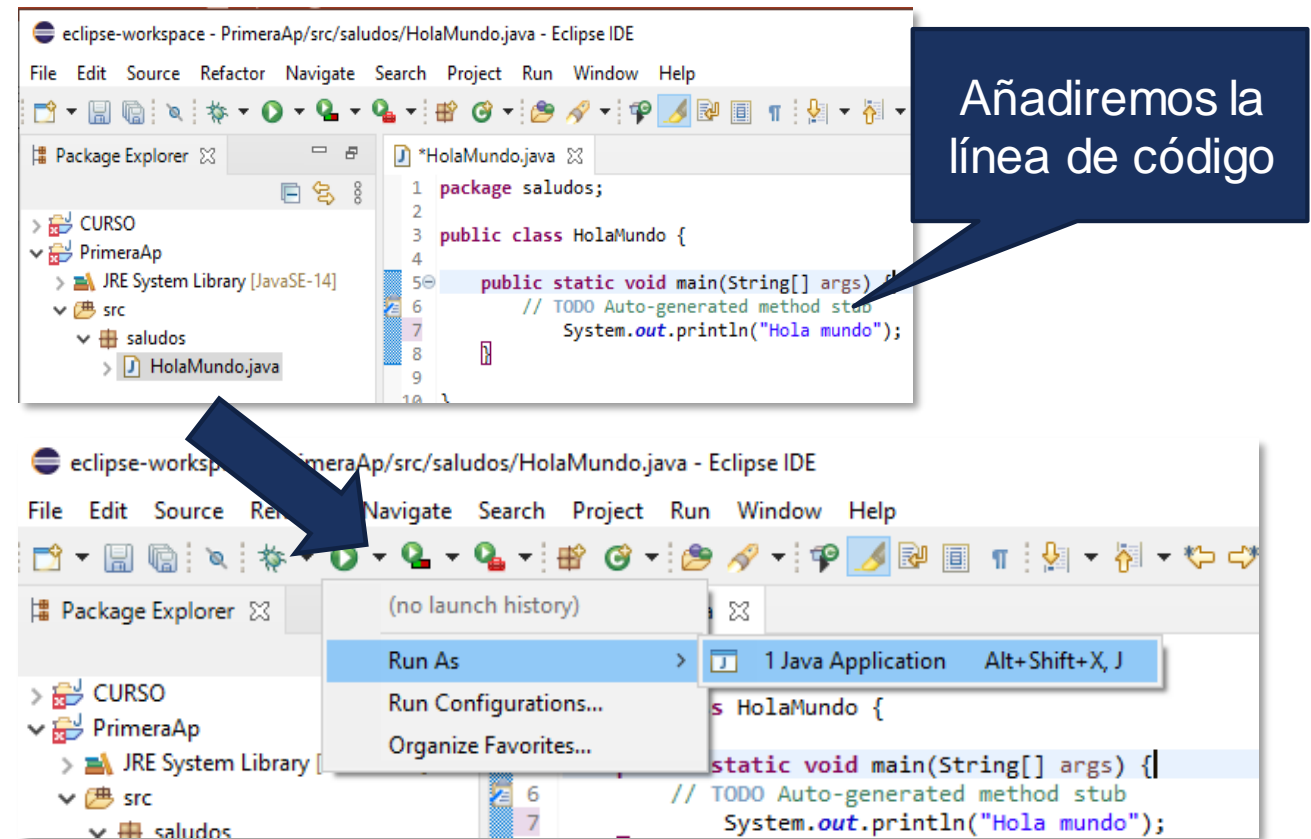
PRIMER PROYECTO EN ECLIPSE

- Una vez creado el proyecto PrimeraAp, crearemos una nueva clase Java (1) para crear el típico mensaje "Hola mundo". Para ello pulsaremos sobre "**PrimeraAp**" con el botón derecho del ratón, y seleccionaremos **New>Class**.
- Indicaremos (2) el nombre del paquete (no se recomienda utilizar el paquete predeterminado), en nuestro caso "**saludos**". Indicaremos el nombre de la clase en nuestro caso "**HolaMundo**".
- Marcaremos "**public static void main(String[] args)**", pues será la clase principal de la aplicación.



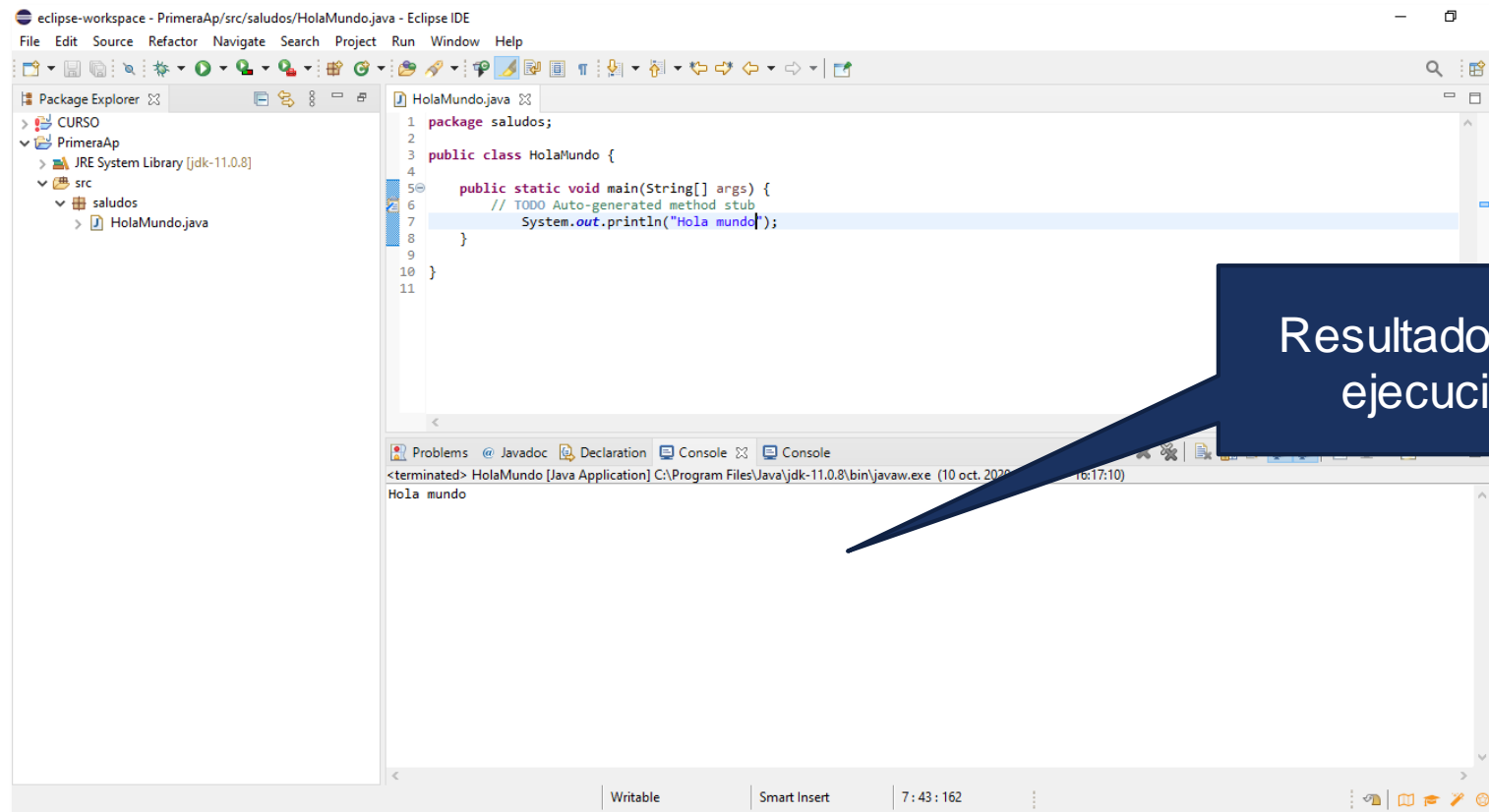
PRIMER PROYECTO EN ECLIPSE II

- Añadiremos la línea de código:
`System.out.println("Hola mundo");`
- Para compilar nuestra aplicación Java Eclipse pulsaremos en la flecha hacia abajo que aparece en la parte derecha del botón "Ejecutar", nos mostrará un menú emergente, seleccionaremos "Run as/Ejecutar como" y pulsaremos en "Java Application/Aplicación Java".



PRIMER PROYECTO EN ECLIPSE III

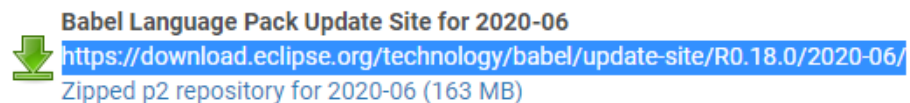
- Veremos en la parte de la consola el resultado de la ejecución.



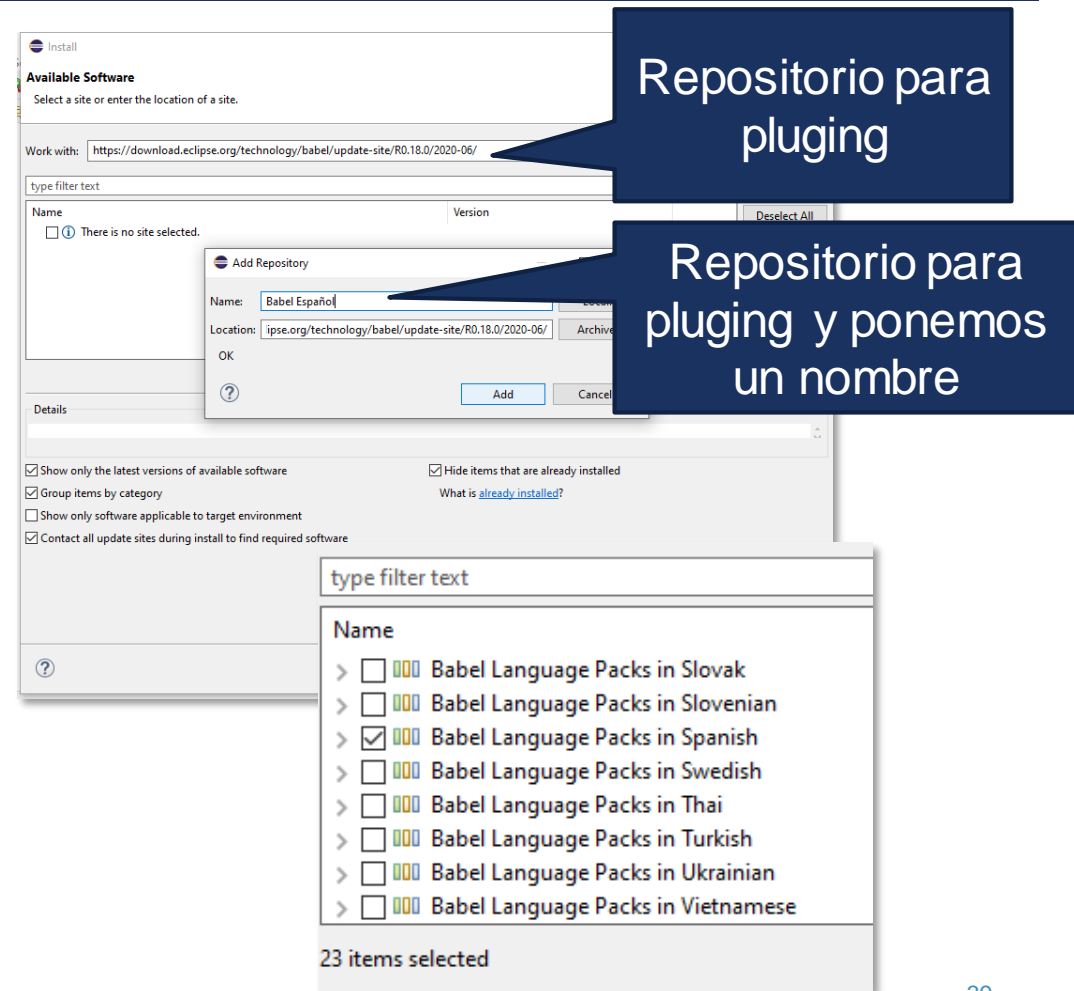
INSTALAR PLUGINS

- Veremos como instalar plugins, en este caso, para ver en castellano las opciones de Eclipse.
- Hacemos click en **>Help>Install New Software**
- Se abre una ventana en la que debemos poner la localización del plugin que queremos instalar. Lo buscamos en:

<https://www.eclipse.org/babel/downloads.php>

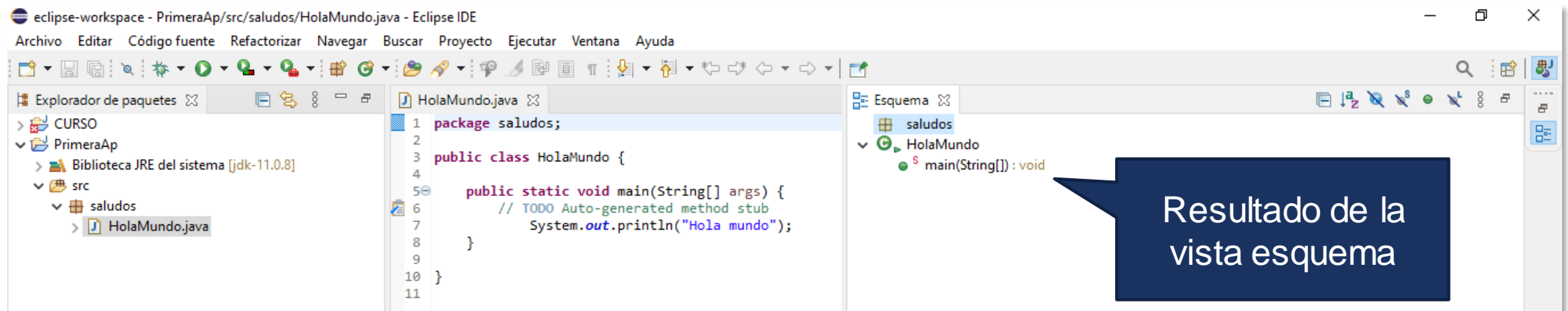


- Lo pegamos en el cuadro de texto de **“Work with”**, hacemos click en **<Add>** y nos sale otro recuadro.
- En el cuadro de texto **“Name”** le daremos un nombre para nuestro repositorio, en este caso **Babel Español**, y luego hacemos click en **<Add>**.
- En la siguiente pantalla seleccionamos el idioma que queremos, en este caso **“Spanish”** y luego hacemos click en **<Next>**, **<Next>**
- Aceptamos las licencias y **<Finish>** y al finalizar la instalación se necesita reiniciar eclipse.



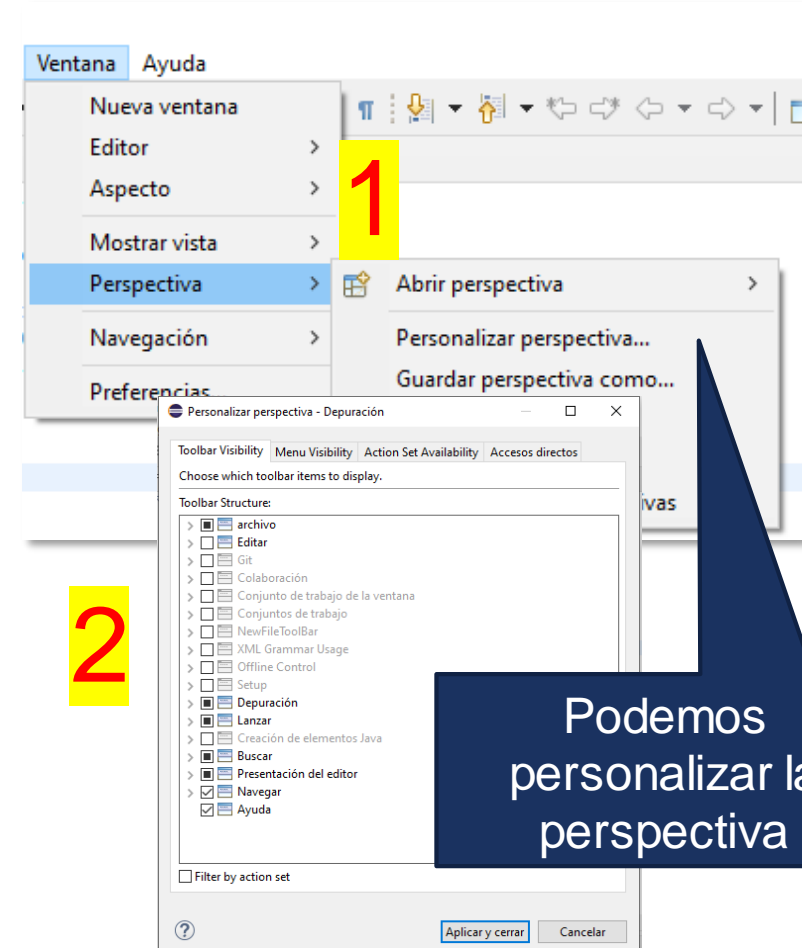
VISTAS Y PERSPECTIVAS

- Eclipse divide el espacio de trabajo en **vistas** y **perspectivas** que podemos configurar para trabajar más cómodos, ambas están en el menú “**ventana**”.
- Las vistas son los paneles en las que se divide el espacio de trabajo proporcionando funciones de apoyo.
- Ej. Vista Esquema: es de las vistas más útiles, permite ver el paquete con el que estás trabajando, la clase, el tipo de datos que usa, los constructores (C) y los tipos de datos empleados en ellos, métodos estáticos (S), finales (F), abstractos (A) y sobrescritos (con un triángulo).



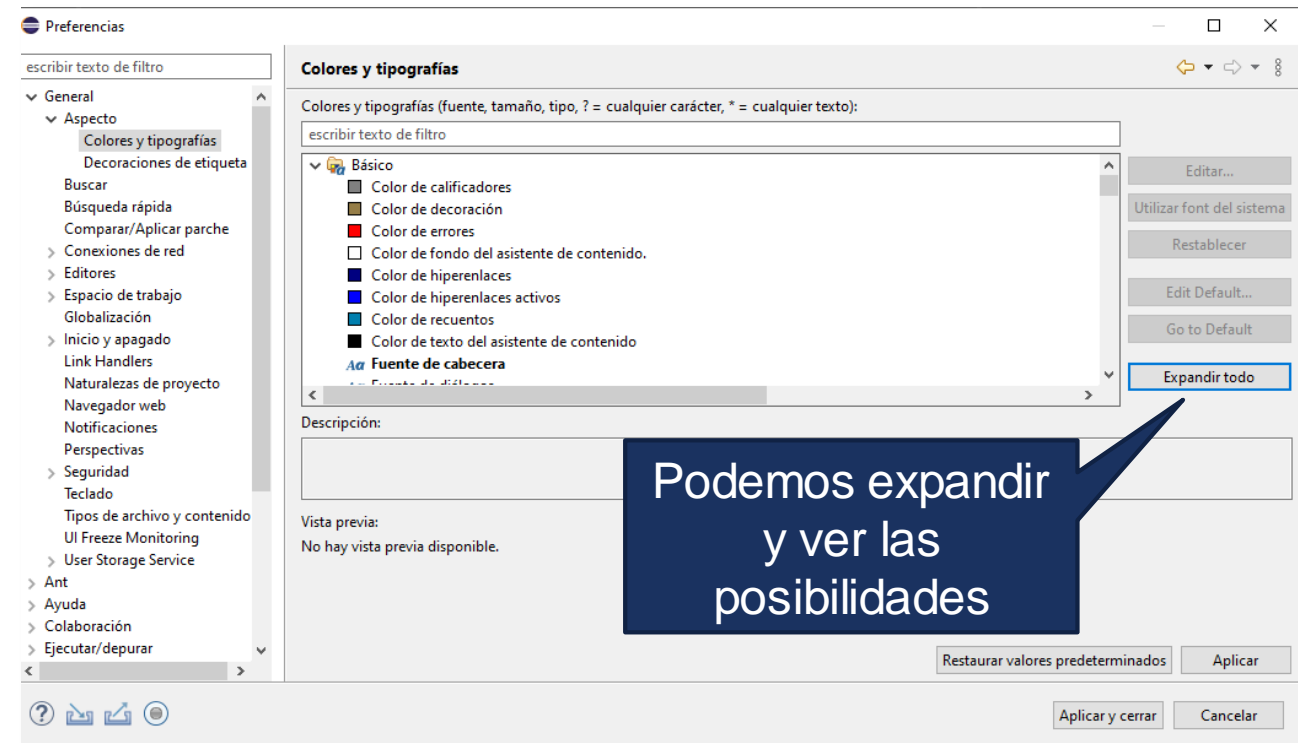
VISTAS Y PERSPECTIVAS II

- Las perspectivas son una preconfiguración de vistas relacionadas entre sí para trabajar de forma óptima. Destacan aquí la perspectiva de “**Java**” y “**Depurar**” (**Debug**). Los desarrolladores son las que más utilizan.
- El **IDE** nos ayuda en el momento de gestionar estos proyectos mediante asistentes y ayudas. Por ejemplo, al crear un nuevo proyecto en Java, se abre la perspectiva optimizada a ese lenguaje, con la colección de vistas que por defecto, considera más apropiadas a ese lenguaje.
- **Perspectiva personalizada:** (1) Podemos establecer la perspectiva que queremos mediante la personalización de la ventana de perspectiva. Escogemos la opción del menú: **Ventana>Perspectiva>Personalizar Perspectiva**
- En la ventana emergente (2), puede seleccionar el contenido que desea establecer, guarde su perspectiva personalizada. **Ventanas>Guardar perspectiva.**



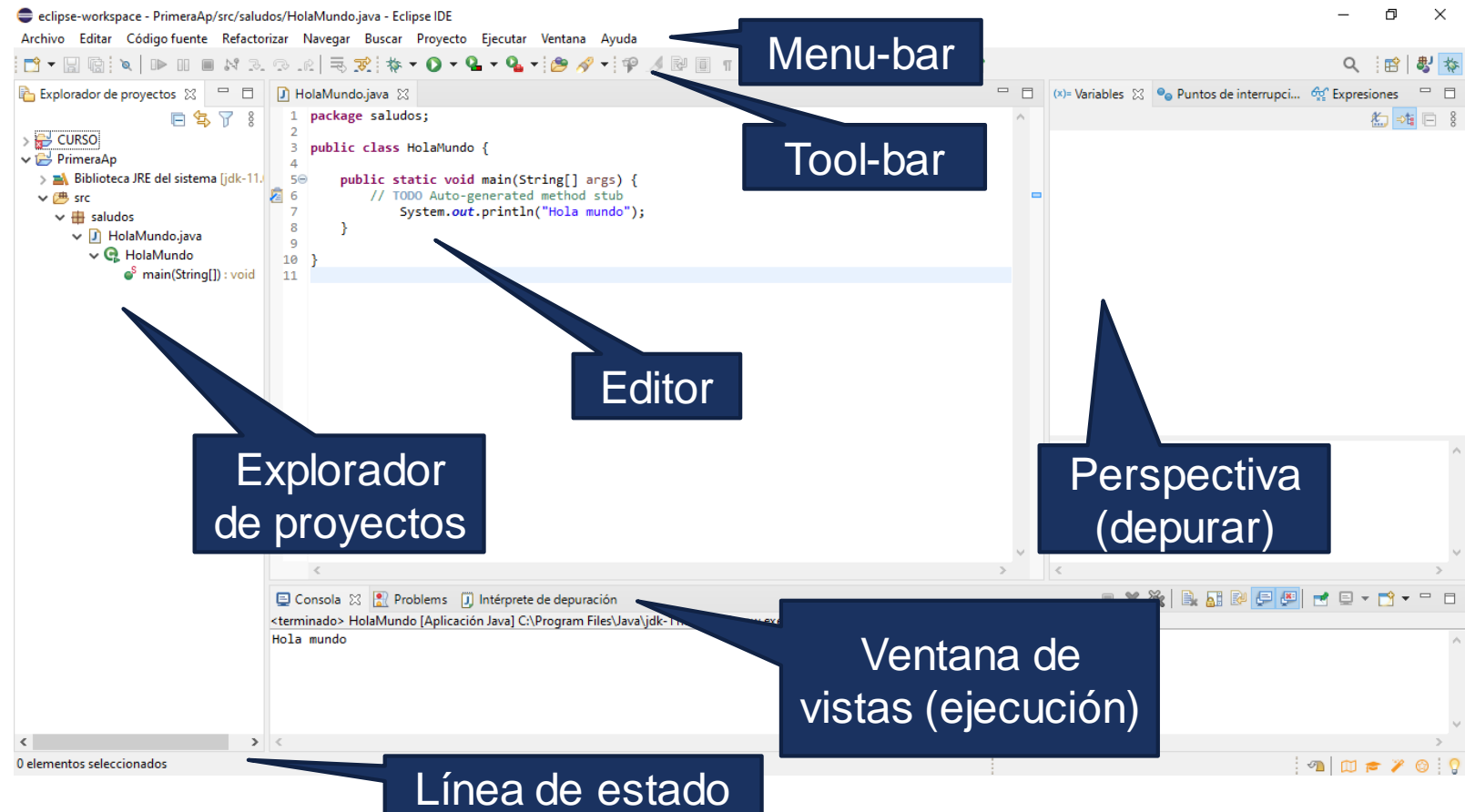
PREFERENCIAS

- Podemos ir a **>Ventana>Preferencias** y revisar si hay alguna de las opciones que queremos ajustar a nuestras necesidades. Si vamos expandiendo, veremos las distintas opciones.
- En un futuro cuando trabajemos con eclipse podemos parametrizar nuestras preferencias.



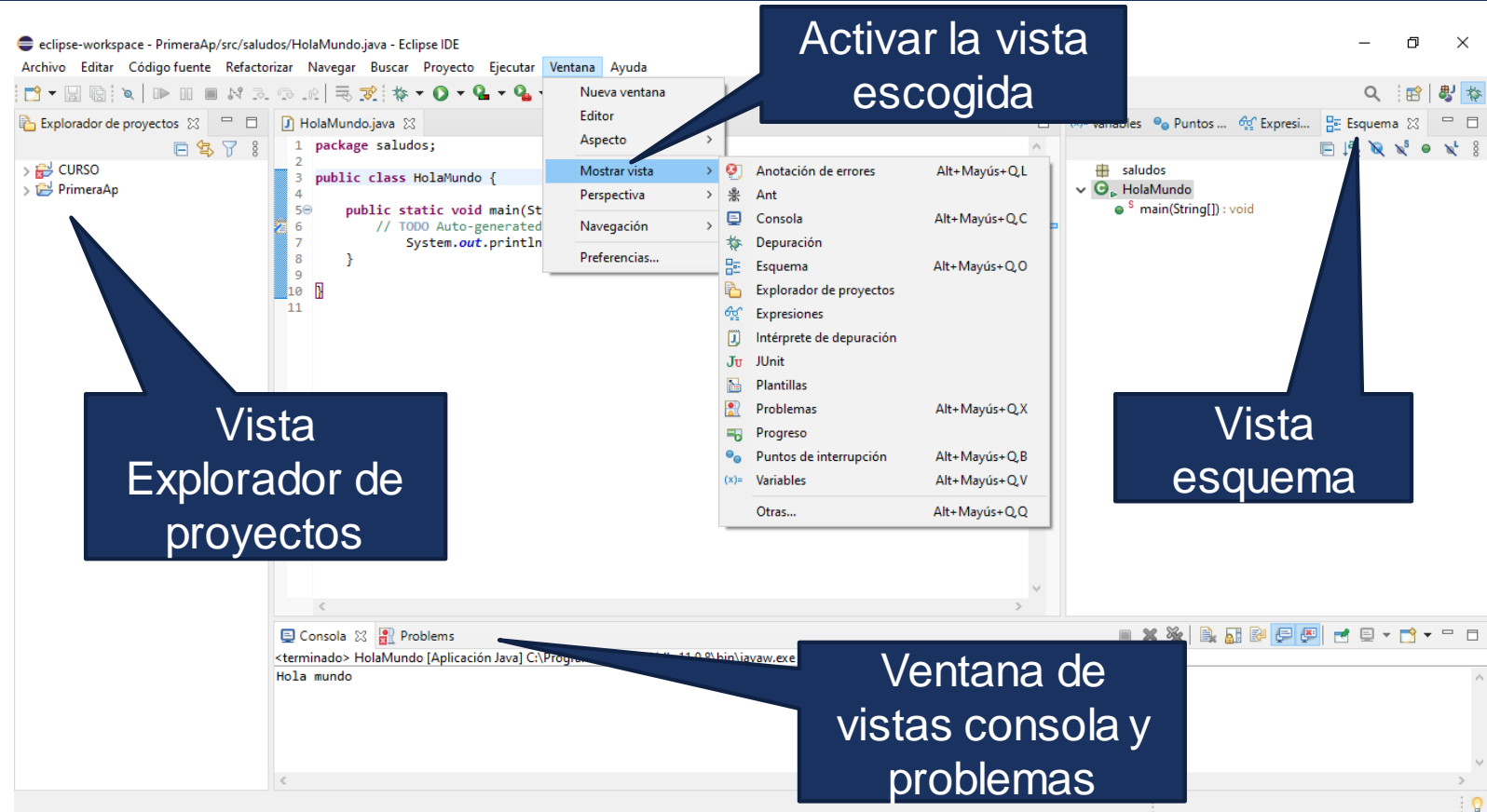
BANCO DE TRABAJO DE ECLIPSE (WORKBENCH)

- Banco de trabajo es un conjunto de múltiples ventanas. Cada ventana contiene una barra de menú, barra de herramientas, barra de accesos directos, así como uno o varios puntos de vista.
- Puede ajustar tamaños
- Eclipse puede también abrir varias ventanas en la barra de menús, seleccione: **>Ventana> Nueva ventana** para abrir varias ventanas.



BANCO DE TRABAJO DE ECLIPSE (WORKBENCH) II

- Eclipse permite a los usuarios visualizar una vista más intuitiva de los elementos de metadatos. Por ejemplo, la vista de navegación archivo de proyecto muestra una representación gráfica de las carpetas y los archivos.
- La siguiente figura muestra como podemos escoger varias vistas.
- Las vistas podemos arrastrarlas y moverlas entre las diferentes ventanas que las contienen.



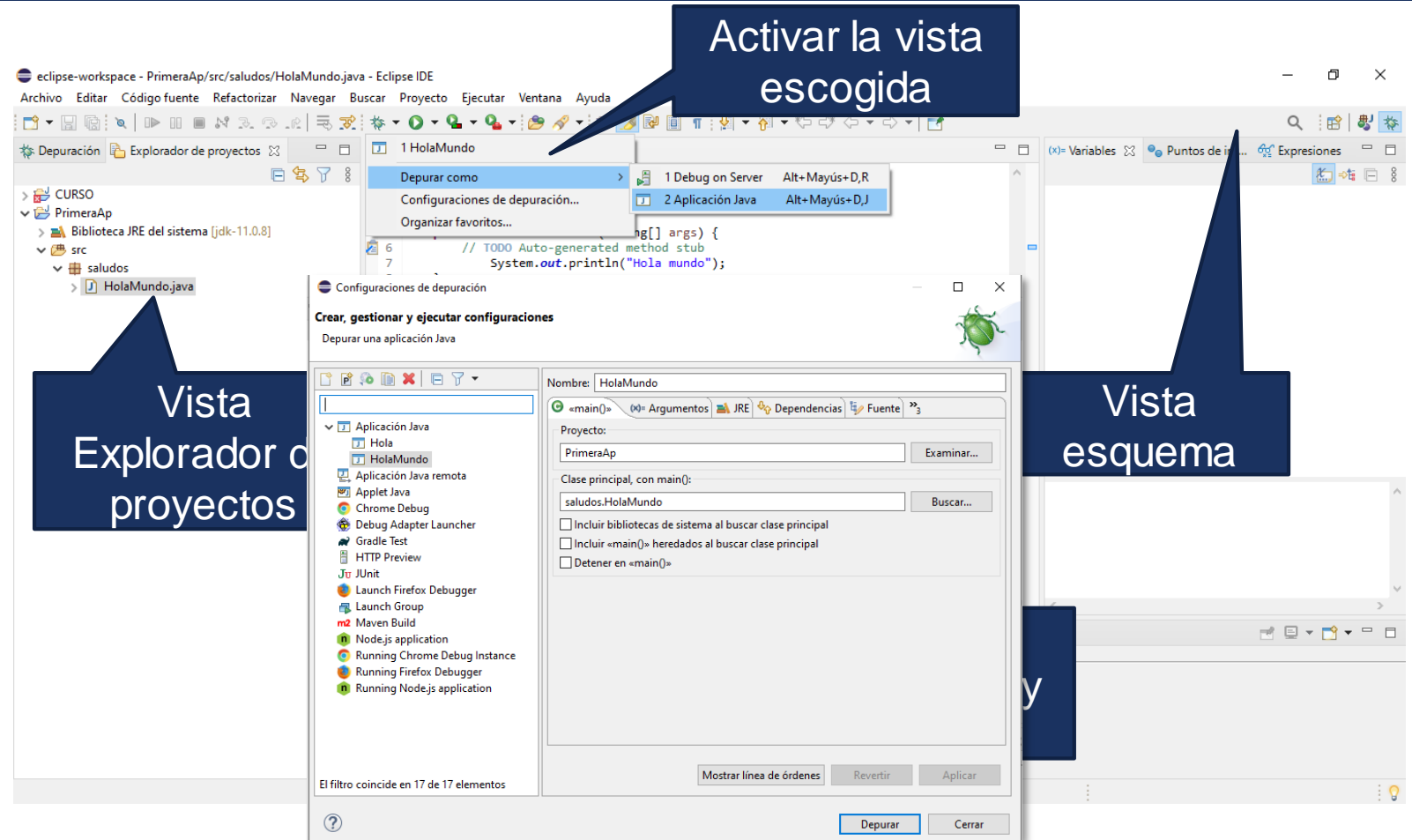
BANCO DE TRABAJO DE ECLIPSE (WORKBENCH) III

- Eclipse permite en la opción Ayuda, ver los contenidos de la ayuda. Marketplace donde podemos encontrar plugins o Acerca de Eclipse IDE.



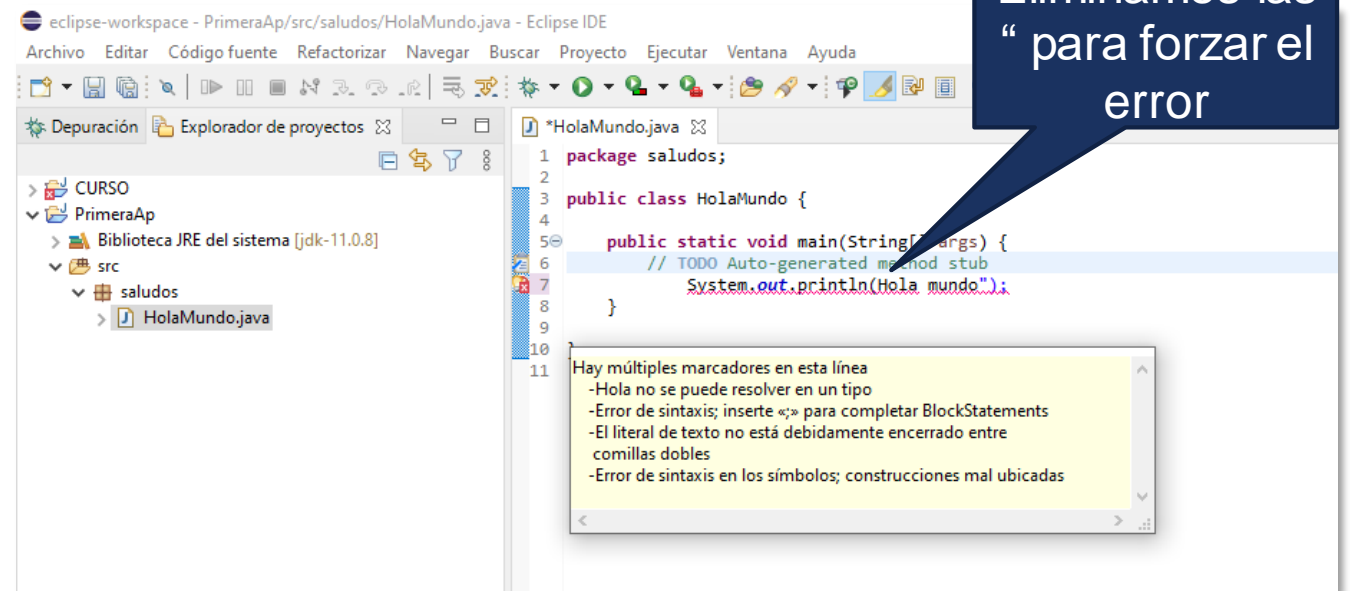
DEPURACIÓN

- Podemos ver el Explorador de paquetes para depurar programas Java, siga estos pasos:
- Haga clic en la función principal contiene la clase de Java
- Seleccione: **Depurar Como> Java Application**
- Esta operación se puede realizar también a través de las teclas de acceso directo, tecla de acceso combinación Alt + Shift + D, J.
- La operación anterior crea una nueva configuración de depuración, que se puede elegir por Configuraciones de depuración. Y
- Por el menú Ejecutar depuración y haga clic en el botón para iniciar la aplicación Java.



DEPURACIÓN II

- En el editor de Eclipse se realiza el análisis de los errores de contenido. Cuando se encuentra un error o una advertencia:
 - Línea roja ondulada destacó errores
 - Utilice líneas onduladas amarillas de advertencia prominente
 - Los errores y advertencias en la vista de problemas
 - Pequeños bulbos y visualización de avisos y errores amarillos identificados en la regla vertical
- Cuadro de diálogo ofrece una solución al problema. cuadro de diálogo Corrección rápida llamando a los siguientes maneras:
 - El puntero del ratón sobre las olas en línea
 - Haga clic en la pequeña bombilla
 - Mover el puntero del ratón sobre el texto resaltado y elegir los elementos de corrección rápida en el menú Edición o el atajo de teclado Ctrl + 1



DEPURACIÓN III

- Si introducimos un error al lanzar la depuración vemos como actúa.
- En la vista de Depuración se observa como en la línea 7 detecta un error. (1)
- También en la vista problemas.
- Una vez solucionado en la vista de Depuración ya no indica el error. (2)

Introducimos error

Vista Depuración

Ventana de vistas problemas

Vista depuración

1

2

Problems View:

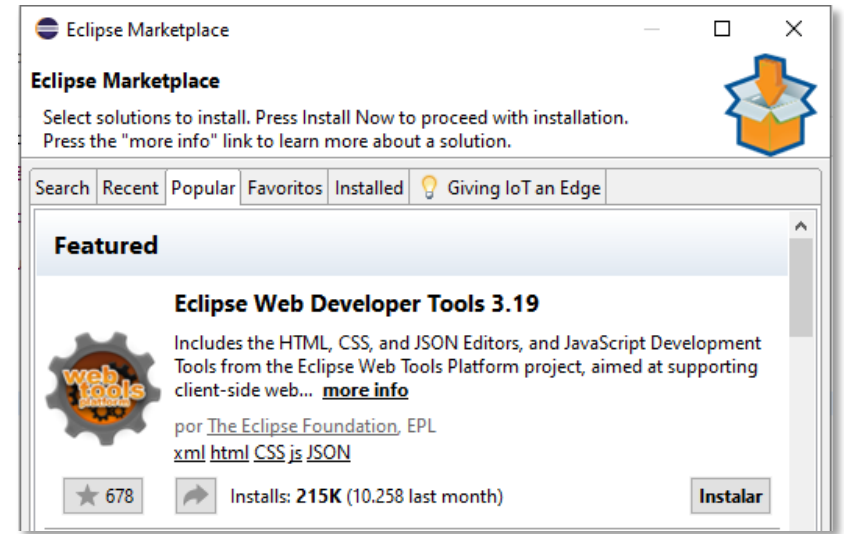
Descripción	Recurso	Ruta	Ubicación	Tipo
El literal de texto no está debidamente encerrado entre comillas dobles	HolaMundo.j...	/PrimeraAp/src/sal...	línea 7	Problema de ...
Error de sintaxis; inserte «» para completar BlockStatements	HolaMundo.j...	/PrimeraAp/src/sal...	línea 7	Problema de ...
Error de sintaxis en los símbolos; construcciones mal ubicadas	HolaMundo.j...	/PrimeraAp/src/sal...	línea 7	Problema de ...
Hola no se puede resolver en un tipo	HolaMundo.j...	/PrimeraAp/src/sal...	línea 7	Problema de ...
Must declare a named package because this compilation unit is associated to the name HolaMundo.j...	/CURSO/src		línea 1	Problema de ...

Debug Console:

Nombre	Valor
main() está lanzando	Error (id=19)

EJERCICIO 5

- Configurar Eclipse. Instala dos plugins y luego desinstala uno de ellos.
- Eclipse. Practicar la depuración. Crea un pequeño proyecto, puedes utilizar alguno ya creado y genera errores, analiza los indicadores del workbench, que te indican como detectarlo y solucionarlo.



Escribe una guía con portada, índice, url consultadas, etc.

- Paso a paso con pantallazos y la explicación, súbelo al aula virtual (AV).