INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO

ENTORNOS DE PROGRAMACIÓN

JAKELIN. SAYEL

“Reír nos hizo invencibles. No como los que siempre ganan, sino como los que nunca se rinden.”

Frida Kahlo

ÍNDICE

Definición de IDE

- Un Entorno de Desarrollo (IDE, en inglés, Integrated Development Environment) **es un software compuesto por una serie de herramientas que utilizan los programadores para desarrollar código.**
  - Puede dar soporte a un único lenguaje de programación o a varios
  - Las herramientas que normalmente componen un IDE son
    - Editor del código de programación
    - Compilador
    - Intérprete
    - Depurador
    - Constructor del interfaz gráfico

Evolución histórica del IDE

- Los primeros entornos de desarrollo integrados nacen a principios de los años 70, y se popularizan en la década de los 90. Tienen el objetivo de ganar fiabilidad y tiempo en los proyectos de software. Proporcionan al programador una serie de componentes con la misma interfaz gráfica, con la consiguiente comodidad, aumento de eficiencia y reducción de tiempo de codificación.
- Normalmente, un IDE está dedicado a un determinado lenguaje de programación. No obstante, las últimas versiones de los IDEs tienden a ser compatibles con varios lenguajes (por ejemplo, Eclipse, NetBeans, Microsoft Visual Studio...) mediante la instalación de plugins adicionales.

Evolución histórica del IDE

- En las décadas de utilización de la tarjeta perforada como sistema de almacenamiento el concepto de Entorno de Desarrollo Integrado sencillamente no tenía sentido.
- Los programas estaban escritos con diagramas de flujo y entraban al sistema a través de las tarjetas perforadas. Posteriormente, eran compilados.
- El primer lenguaje de programación que utiliza un IDE fue el BASIC.
- Éste primer IDE estaba basado en consola de comandos exclusivamente. Sin embargo, el uso que hace de la gestión de archivos, compilación, depuración… es perfectamente compatible con los IDE actuales.
- A nivel popular, el primer IDE puede considerarse que fue el IDE llamado Maestro I. Nació a principios de los 70 y fue instalado por unos 22000 programadores en todo el mundo. Lideró el campo durante los años 70 y 80.
- 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.

EVOLUCIÓN HISTÓRICA DEL IDE

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

Funciones de un entorno de desarrollo

- Como sabemos, los entornos de desarrollo están compuestos por una serie de herramientas software de programación, necesarias para la consecución de sus objetivos. Estas herramientas son:
  - Un editor de código fuente.
  - Un compilador y / o un intérprete.
  - Automatización de generación de herramientas.
  - Un depurador.

Principales Funciones de un entorno de desarrollo

- 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.

Otras Funciones de un entorno de desarrollo

- 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.

Entornos propietarios y libres

→ 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.

Entornos propietarios y libres

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++, JavaScript, Python	Linux
Geany	C/C++, Java	Windows, Linux, Mac OS X
GNAT Studio	Fortran	Windows, Linux, Mac OS X

Entornos propietarios y libres

➔ Entornos Integrados Propietarios

- ➔ 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.

Entornos propietarios y libres

IDE	Lenguajes que soporta	Sistema Operativo
Microsoft Visual Studio	C/C++, Basic, 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

Estructura de Entornos de Desarrollo

- **Los entornos de desarrollo, ya sean libres o propietarios, están formados por una serie de componentes software que determinan sus funciones.**
- **Estos componentes son:**

  - **Editor de textos:** Se resalta y colorea la sintaxis, tiene la función de 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 todo un arsenal de asistentes y utilidades de gestión y generación código.
  - **Interfaz gráfica:** Nos brinda la oportunidad de programar en varios lenguajes con un mismo IDE. Es una interfaz agradable que puede acceder a innumerables bibliotecas y plugins, aumentando las opciones de nuestros programas.

Configuración y personalización de entornos de desarrollo

- Una vez tenemos instalado nuestro entorno de desarrollo podemos acceder a personalizar su configuración.
- Al abrir un proyecto existente, o bien crear un nuevo proyecto, seleccionaremos un desplegable con el nombre de “configuración” desde el que podremos personalizar distintas opciones del proyecto.
- Podemos personalizar la configuración del entorno sólo para el proyecto actual, o bien para todos los proyectos, presentes y futuros.

Configuración y personalización de entornos de desarrollo

Parámetros configurables del entorno:

- ✔ Carpeta/s donde se alojarán todos los archivos de los proyectos (es importante la determinación de este parámetro, para tener una estructura de archivos ordenada).
- ✔ Carpetas de almacenamiento de paquetes fuente y paquetes prueba.
- ✔ Administración de la plataforma del entorno de desarrollo.
- ✔ Opciones de la compilación de los programas: compilar al grabar, generar información de depuración...
- ✔ Opciones de empaquetado de la aplicación: nombre del archivo empaquetado (con extensión .jar, que es la extensión característica de este tipo de archivos empaquetados) y momento del empaquetado.
- ✔ Opciones de generación de documentación asociada al proyecto.
- ✔ Descripción de los proyectos, para una mejor localización de los mismos.
- ✔ Opciones globales de formato del editor: número de espaciados en las sangrías, color de errores de sintaxis, color de etiquetas, opción de autocompletado de código, propuestas de insertar automáticamente código...
- ✔ Opciones de combinación de teclas en teclado.

ECLIPSE

Conocimientos previos.

- IDE (Integrated Development Environment): Entorno donde el programador tiene todas las herramientas de trabajo a su disposición.
- JDK: Java Development Kit (video instalación)
- JRE (Java Runtime Environment): Librerías básicas para ejecutar programas Java.
- JVM: Java virtual machines.

Instalación.

1. DESCARGAR ECLIPSE DESCARGAR LA ÚLTIMA VERSIÓN (TAMBIÉN UNA ANTIGUA POR EJEMPLO PHOTÓN)
   Ej: Eclipse 2021-9-> R packages (actualizaciones que le han hecho) y Descargais Eclipse IDE for Java Developers

2. DESCOMPRIMIR ECLIPSE (BOTÓN DERECHO -> DESCOMPRIMIR)
3. SIGUIENTE, SIGUIENTE…

video instalación

Instalación.

- ¿Cómo sé si ya está instalado el JVM?
  - Nos vamos a la consola o terminal y ejecutamos:

```
$ java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b15)
Java HotSpot(TM) Client VM (build 25.91-b15, mixed mode)
```

Instalación.

- ¿Cómo sé si ya está instalado el JDK?
  - Nos vamos a la consola o terminal y ejecutamos:

```
$ javac -version
javac 1.8.0_05
```

Instalación.

- ¿Qué hay que hacer para instalar el JRE y el JDK en Linux?
  - Nos vamos a la consola o terminal y ejecutamos :

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install default-jre
$ sudo apt-get install default-jdk
```

➔ Configurar variables de entorno

Configurar variables de entorno

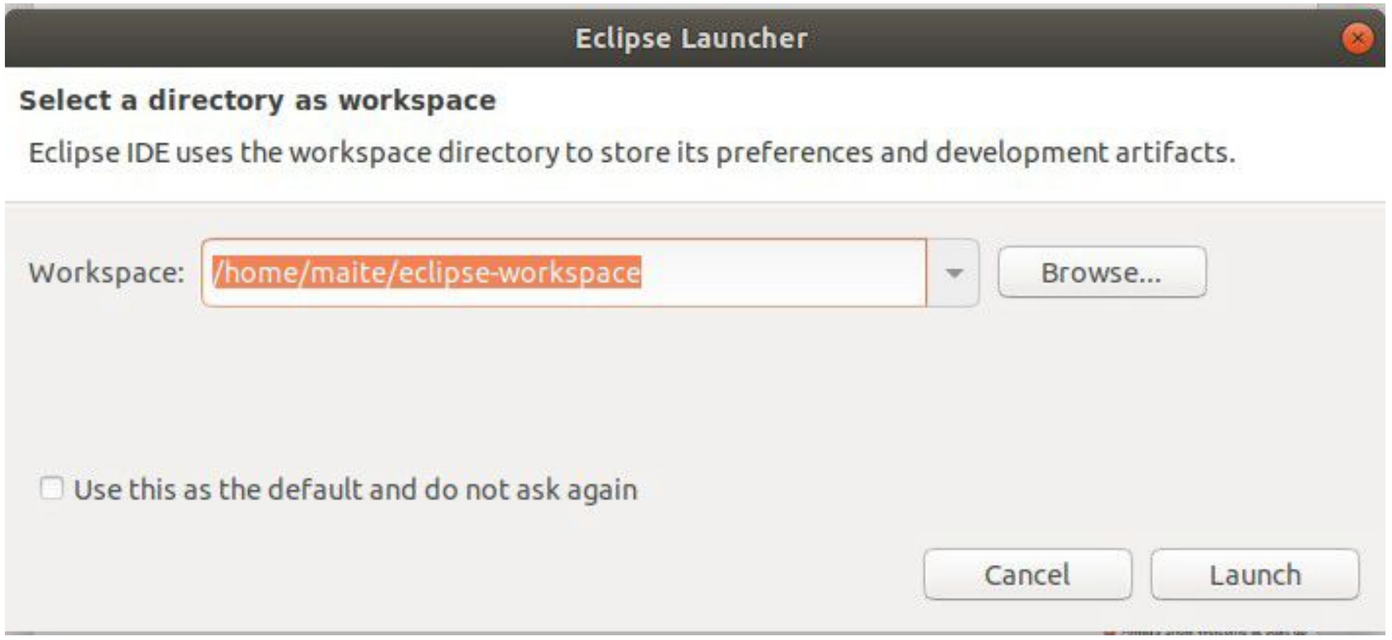
JAVA_HOME → Ruta de instalación de JDK

Path → %JAVA_HOME%\bin

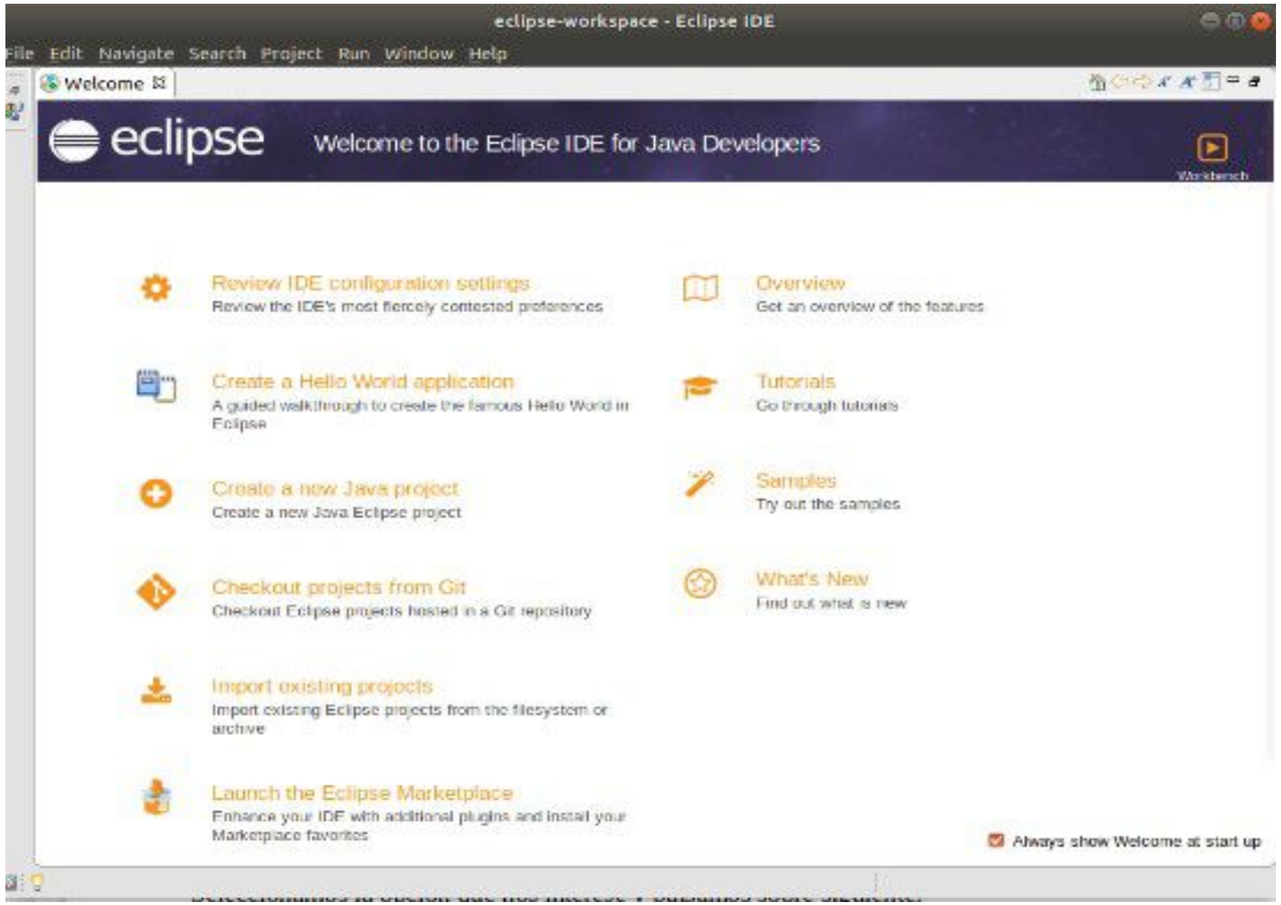
Uso básico.

1. Crear proyecto. -> Nombre del programa
2. Programa “Nombre del programa”.
3. Incorporar ficheros a un proyecto
4. Edición de programas
5. Compilación de programas
6. Ejecución de programas

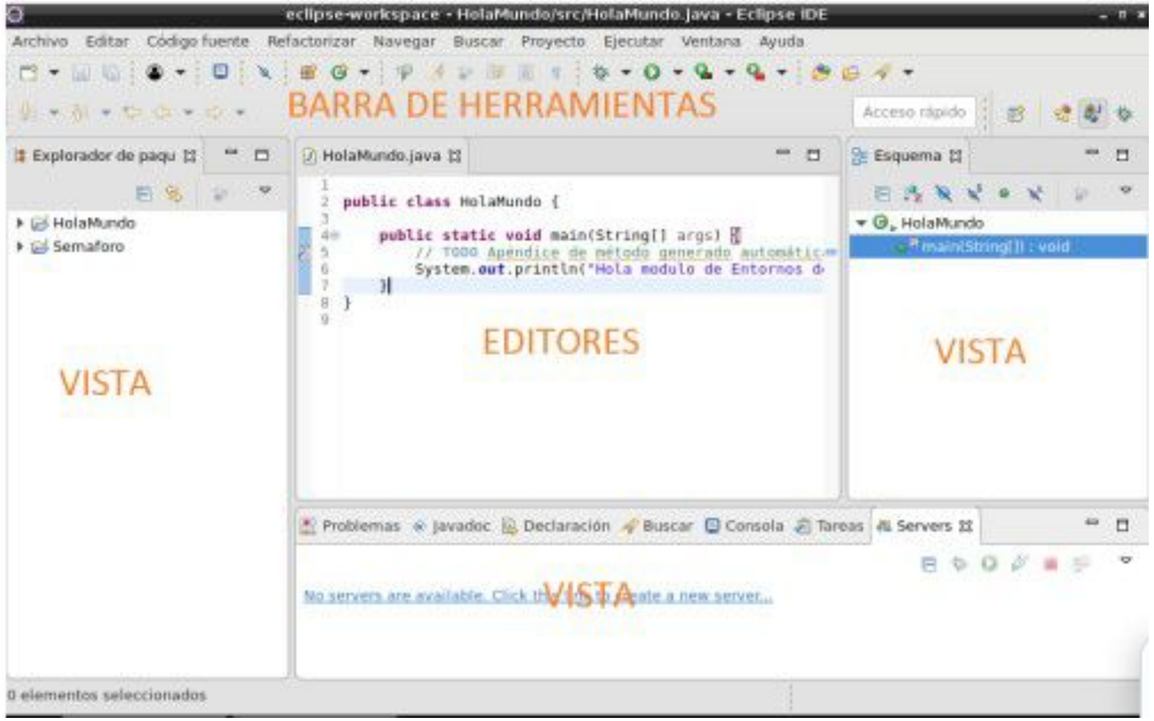
- Una vez descargamos y extraemos eclipse podemos pasar a utilizarlo:
- Al abrirlo nos pregunta el workspace que queremos utilizar, si no queremos que vuelva a preguntar, marcamos el check.



- A continuación, nos aparecerá la ventana de Bienvenida y ya podemos empezar a crear nuestros proyectos.
- Al igual que con el workspace podemos indicar que no vuelva a aparecer.

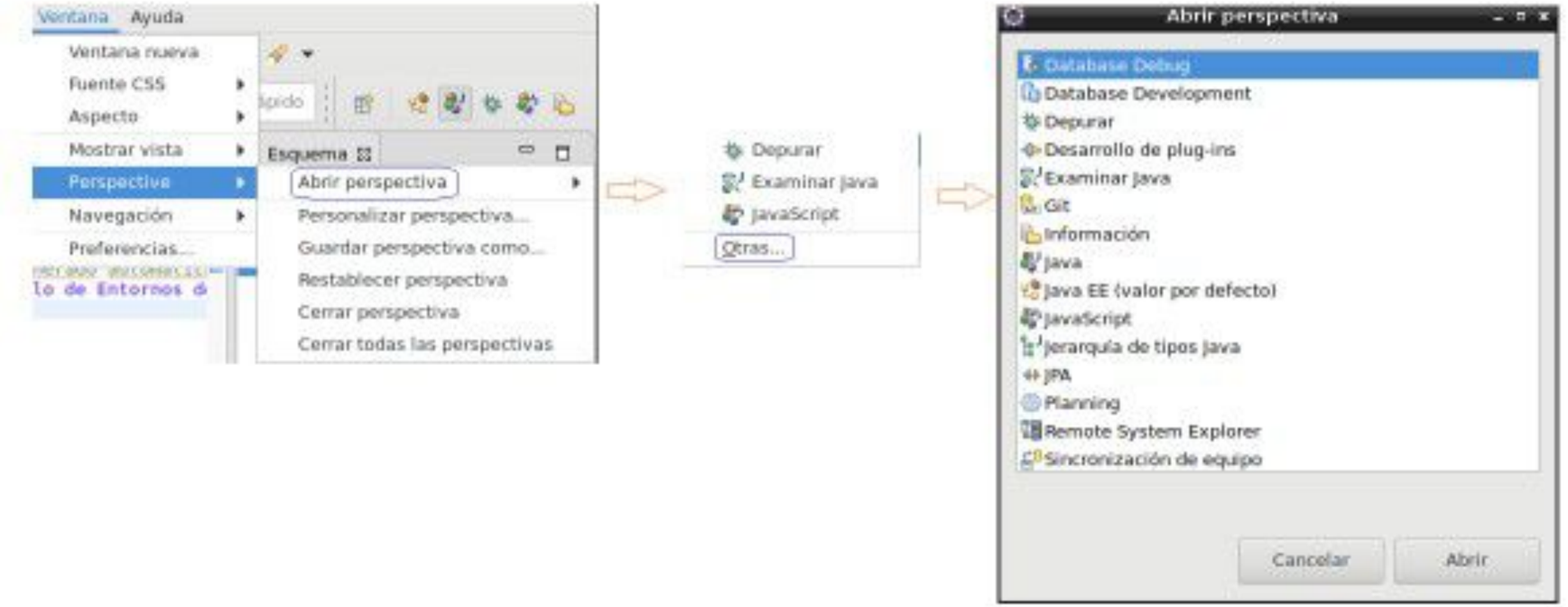


- **Zona de edición**. Espacio donde escribir el código de los programas. Es posible tener abiertos varios archivos simultáneamente. Éstos aparecerán organizados por pestañas
- **Vista**. Se trata de un conjunto de ventanas que proporcionan funciones de apoyo. Cada perspectiva podrá tener sus propias vistas. Es posible mostrar u ocultar las vistas desde la opción de menú Ventana/Mostrar Vista.

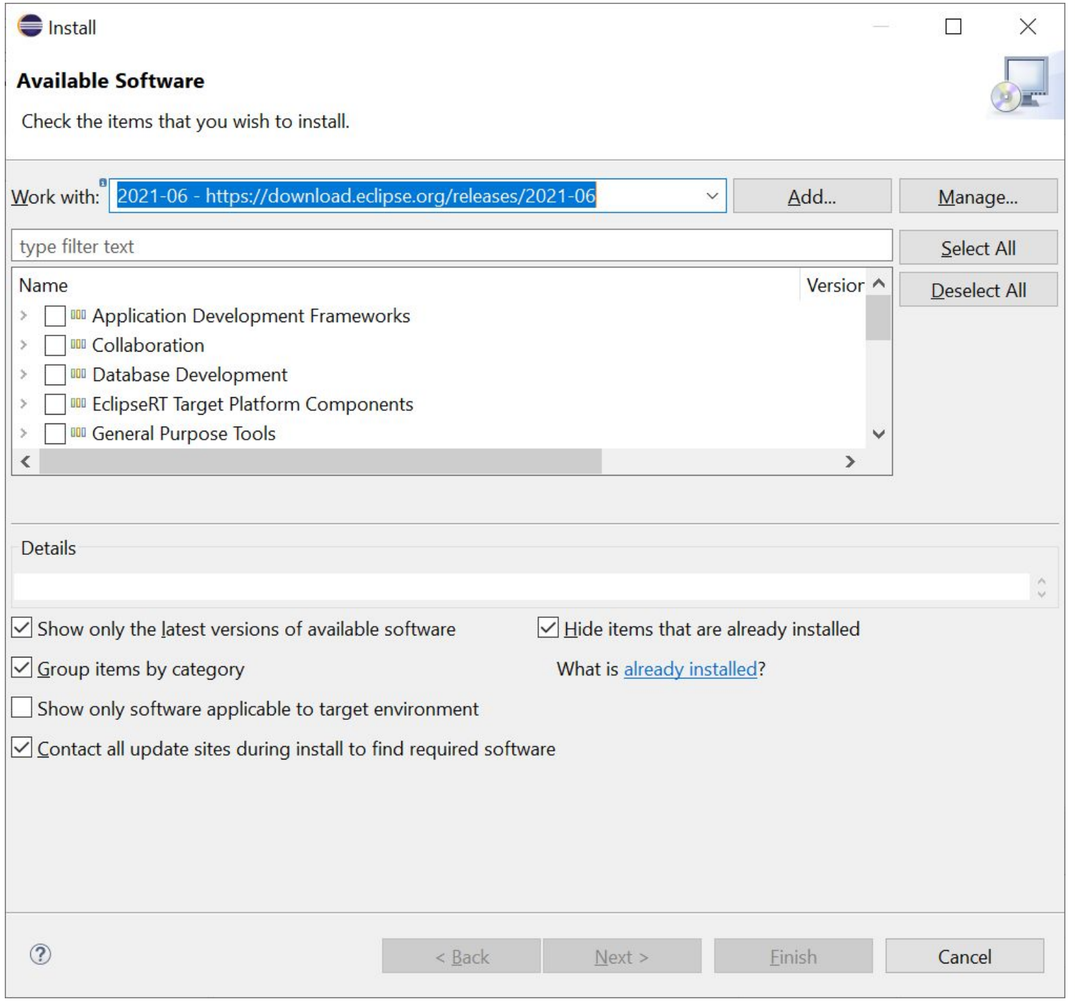


- **Barra de herramientas**. Se trata de un acceso directo a las operaciones proporcionadas por Eclipse, éstas están agrupadas por funcionalidades: abrir/crear nuevos proyectos, compilar, depurar, refactorizar, gestión de vistas y perspectivas ...

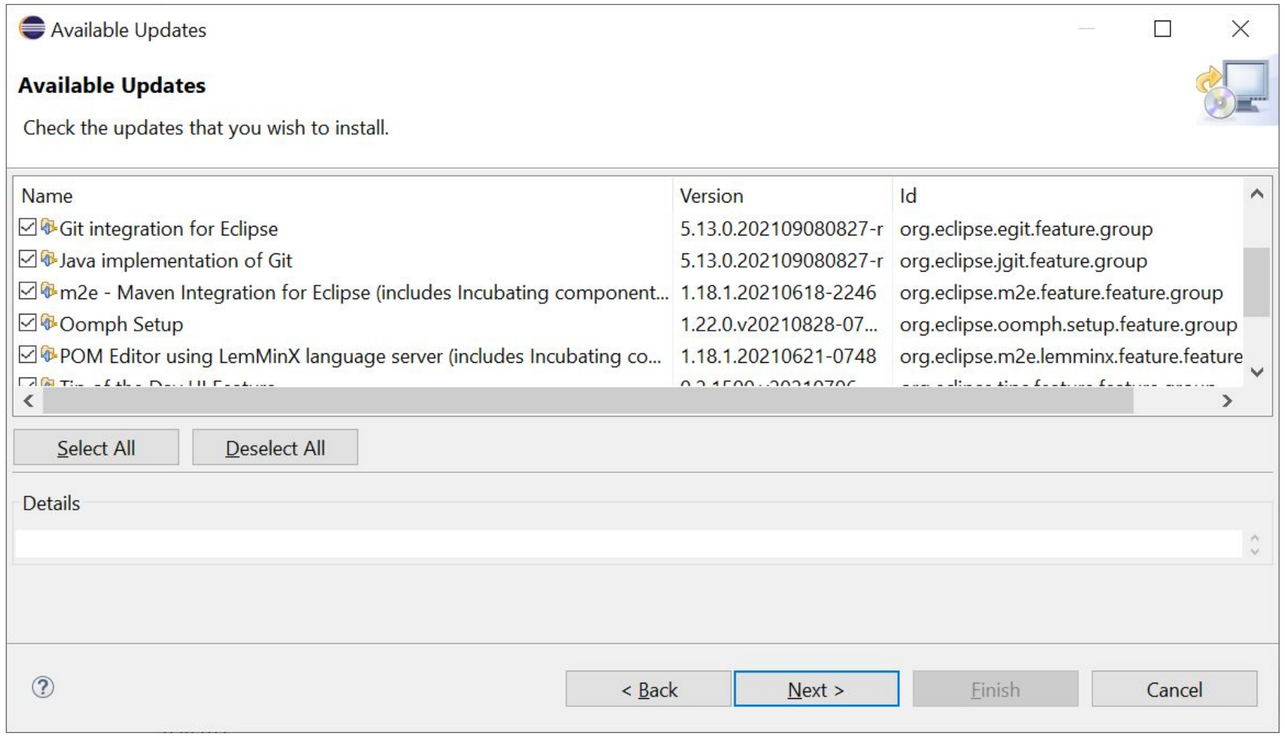
- Dependiendo de la tarea en la que estemos trabajando en cada momento, demandaremos de Eclipse unos servicios u otros: desarrollo de código en diferentes lenguajes (java, javascript, ..), depurar el programa, implementar plugins ... Para cada tarea, Eclipse ofrece un interfaz adaptado a las necesidades. Esto es lo que se llama **perspectiva**.
- Estas son la que eclipse posee en la instalación base, pero se pueden crear personalizadas.



Eclipse emplea módulos (**plug-in**) para añadir funcionalidades a las ya existentes, se puede ver como "un esqueleto" que ofrece unas herramientas de partida, pero es capaz de integrar diferentes paquetes adicionales que proporcionaran otros servicios requeridos por cada proyecto. La gran ventaja de las aplicaciones basadas en módulos es que son muchos más "ligeras", aquellos módulos no instalados, no consumen recursos en el equipo de forma innecesaria. Existen en la actualidad multitud de plugins disponibles para gran parte de los entornos de desarrollo más utilizados. Se pueden añadir desde el menú Help→ Install new software...



- Para mantener nuestro IDE al día y así garantizar las mejores prestaciones o solventar posibles errores detectados en nuestra distribución, Eclipse dispone de la opción de menú "Help/Check for updates", que nos mostrará un listado de todos los paquetes que han quedado desactualizados y la posibilidad de instalar las últimas versiones



SEGUIMOS!!