

ENTORNOS DE DESARROLLO

TEMA 2:

INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO

Desarrollo de Aplicaciones Multiplataforma



CONTENIDO BOCYL (nº 115 15-junio-2011)

- Funciones de un entorno de desarrollo. Estructura. Clasificación de entornos de desarrollo. Características comunes y específicas.
- Instalación de un entorno de desarrollo.
- Personalización y actualización del entorno de desarrollo.
- Uso básico de un entorno de desarrollo.
- Edición de programas.
- Generación de ejecutables.

Índice

CONTENIDO BOCYL (nº 115 15-junio-2011)	2
Índice	3
Índice de ilustraciones	5
1. CONCEPTO DE ENTORNO DE DESARROLLO	6
1.1. EVOLUCIÓN HISTÓRICA.....	7
1.1. ELECCIÓN DEL ENTORNO	10
1.2. COMPONENTES.....	11
1.2.1. Editor de código de programación	11
1.2.2. Compilador	11
1.2.3. Enlazador	11
1.2.4. Intérprete.....	11
1.2.5. Depurador.....	11
1.2.6. Constructor de interfaz gráfico	12
1.2.7. Otras herramientas.....	12
2. XAMPP	13
3. GESTOR DE BASES DE DATOS	21
3.1. Crear base de datos.....	21
3.2. Crear tabla	21
3.3. Añadir registros	22
3.4. Usar SQL	23
3.5. Eliminar base de datos	24
3.6. Importar base de datos	24
3.7. Pestaña de diseñador.....	25
4. JAVA	26
5. MANUAL DE ECLIPSE	28
5.1. Instalación.....	28
5.2. Elección de workspace.....	29

5.3.	Personalización del entorno	29
5.4.	Perspectivas y vistas	31
5.5.	Crear un proyecto	37
5.6.	Exportar/Importar proyectos.....	40
5.7.	Debug.....	42
5.7.1.	Evaluate expression.....	49
5.8.	Añadir bibliotecas	50
5.9.	Errores y warnings de compilador	60
5.10.	Generador de código.....	62
5.11.	Generador de documentación	65
5.12.	Refactorizar y buscar.....	69
5.13.	Atajos de teclado.....	73
5.14.	Control de versiones	73
5.15.	Plugins	73
5.16.	Bases de datos.....	80
	REFERENCIAS	87

Índice de ilustraciones

No se encuentran elementos de tabla de ilustraciones.

1. CONCEPTO DE ENTORNO DE DESARROLLO

Un entorno integrado de desarrollo (IDE), es un tipo de software compuesto por un conjunto de herramientas de programación que pretenden facilitar el desarrollo de software, así como agilizarlo. En concreto, el IDE se compone de:

Editor de código de programación.

Compilador.

Enlazador.

Intérprete.

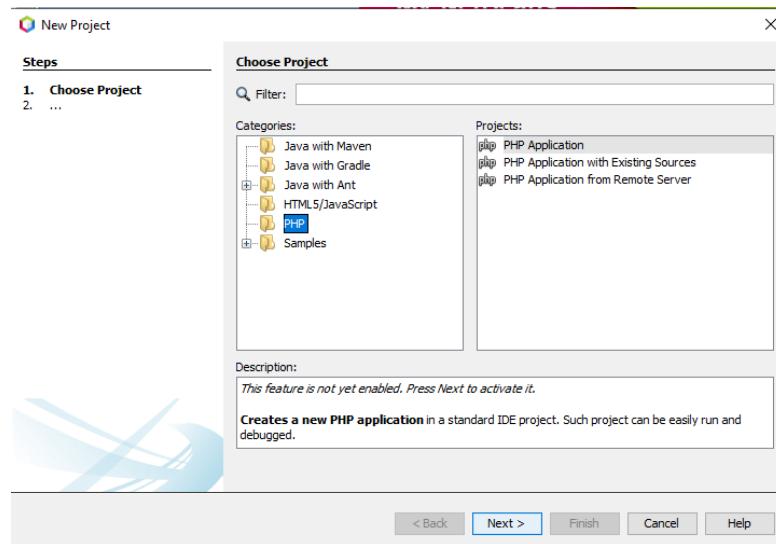
Depurador.

Constructor de interfaz gráfico.

Otras herramientas.

Antiguamente lo más normal era que los IDEs estuvieran dedicados a un determinado lenguaje de programación, no obstante, las últimas versiones tienden a ser compatibles con varios lenguajes (por ejemplo, Eclipse, NetBeans, Microsoft Visual Studio), bien sea porque el entorno los contempla por defecto una vez esté instalado o porque se pueden ir añadiendo mediante la instalación de plugins.

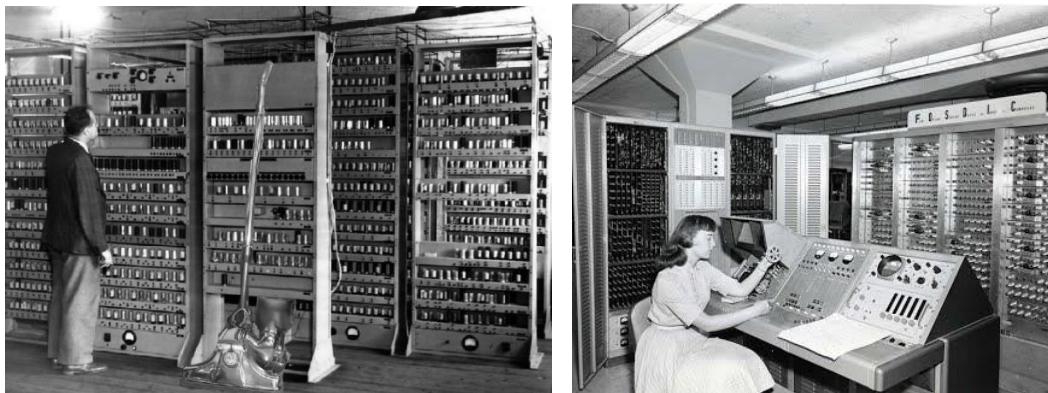
NetBeans, por ejemplo, nos proporciona de manera inicial la posibilidad de trabajar con proyectos de Java, PHP, HTM, etc.



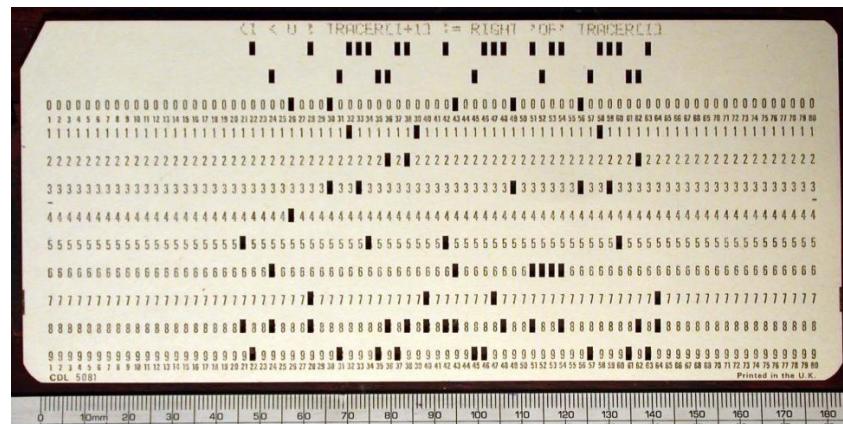
1.1. EVOLUCIÓN HISTÓRICA

Como todo aquello que pone en cuestión el éxito o la precocidad de naciones o egos, la historia de la informática no se ve exenta de polémicas relacionadas con fechas o efemérides, por ello dependiendo de la fuente por la que nos informemos, podremos observar que existen variaciones en torno a fechas o primeros ordenadores.

Nos vamos a mover a la primera generación de ordenadores, el Big Bang que dio origen al universo de la informática. Esta generación comienza en 1940 y, como os podéis imaginar, los ordenadores no se parecían en nada a los que manejamos hoy en día.

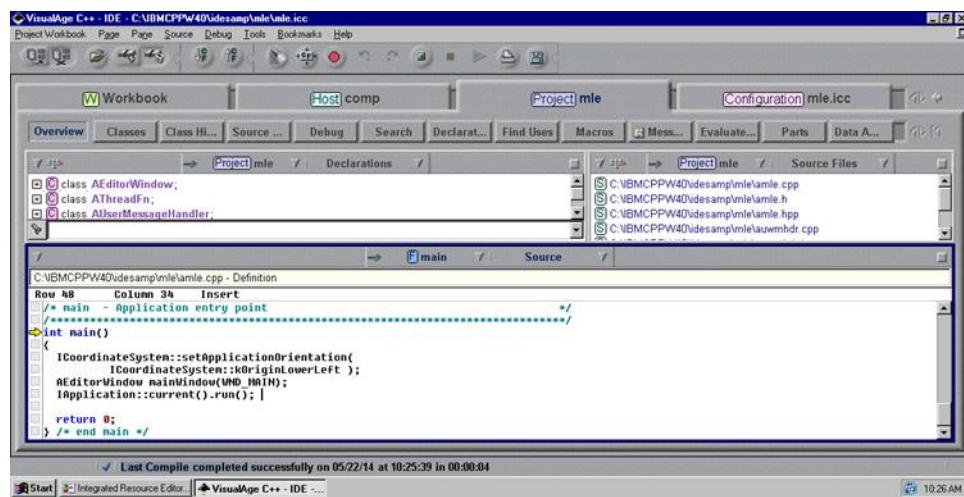


En esta generación no existía el teclado, la entrada de información e instrucciones se realizaba mediante tarjetas perforadas, que contenían los programas escritos con diagramas de flujo.



Como comprenderéis, en este punto, ni siquiera se contemplaba el concepto de entorno de desarrollo. Los IDEs tuvieron que esperar un tiempo para poder cobrar cierto sentido y nacer en el mercado de la informática. A nivel popular, el primer Entorno de Desarrollo puede considerarse que fue el Maestro I. Nació a principios de los 70, fue instalado por unos 22000 programadores en todo el mundo y lideró el campo durante los años 70 y 80.

En 1984 aparece el entorno de IBM llamado VisualAge que manejaba varios lenguajes de programación (BASIC, C, COBOL, C++, EGL, Fortran, PACBASE, Smalltalk, etc), este software daría paso posteriormente a la creación de Eclipse.

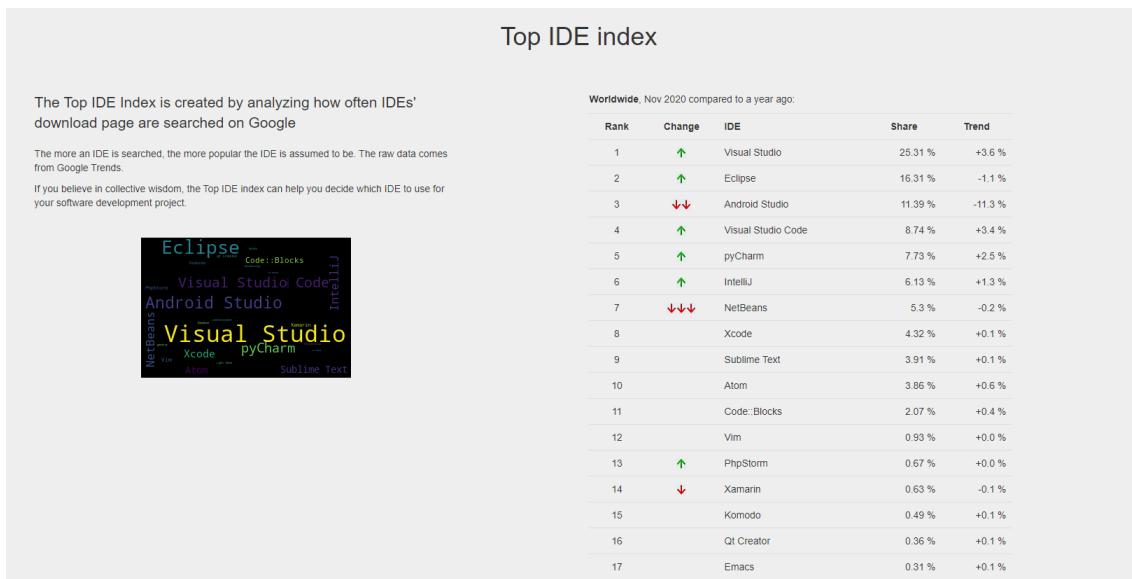


A partir de aquí el concepto de entorno de desarrollo adquirió una popularidad mucho mayor y comenzaron a nacer muchos de los IDEs que utilizamos actualmente.



En la actualidad no se contempla la programación a nivel de empresa sin Entorno de Desarrollo.

A continuación, se muestra los IDEs más populares en 2020, considerando como mecanismo de ordenación la cantidad de búsquedas de su página de descarga en Google.



Los IDEs más populares de 2004 a 2019:
https://www.youtube.com/watch?v=3CwJW5_C6R4

1.1. ELECCIÓN DEL ENTORNO

No hay unos entornos de desarrollo absolutamente mejores que otros. La elección del IDE más adecuado dependerá del lenguaje de programación que vayamos a utilizar para la codificación de las aplicaciones, el tipo de licencia con la que queramos trabajar, la experiencia que tengamos haciendo uso de este, las características concretas del proyecto que vamos a desarrollar, etc.

La elección del IDE adecuado para una empresa de desarrollo de software es muy importante y puede llegar a condicionar el éxito de los proyectos que desarrolle.

A la hora de elegir el IDE algunas de las características que más se tienen en cuenta son las siguientes:

- **Cantidad de lenguajes compatibles:** Algunos entornos de desarrollo solo son compatibles con un lenguaje de programación y/o son mejores para un modelo de programación específico, por eso, si nos embarcamos en el desarrollo de un proyecto concreto, conviene tener en cuenta toda esta casuística. Por ejemplo, IntelliJ es conocido principalmente como un IDE de Java. Otros admiten una gran variedad de lenguajes de manera conjunta, como el IDE Eclipse, que admite Java, XML, Python, entre otros.

Será muy importante, por tanto, que dentro de los lenguajes que ofrece un entorno, se encuentre aquel en el que se tiene pensado programar.

- **Sistemas operativos disponibles:** Es importante tener en cuenta el sistema operativo que van a usar los desarrolladores y comprobar si el entorno elegido tiene una buena versión para este.
Este apartado cobra especial importancia cuando la aplicación que se desarrolla está diseñada para un usuario final con un sistema operativo específico (como Android o iOS).
- **Características de automatización:** Si bien la mayoría de los IDE incluyen tres funciones fundamentales (el editor de texto, la automatización de compilación y el depurador), muchos admiten funciones adicionales, como la refactorización, la búsqueda de código y las herramientas de integración e implementación continuas (CI/CD).
- **Impacto en el rendimiento del sistema:** Es importante tener esta característica en cuenta si el desarrollador quiere hacer un uso intensivo de la memoria al mismo tiempo.

- **Complementos y extensiones:** Algunos entornos de desarrollo incluyen una función para personalizar los flujos de trabajo de forma que se adapten a las necesidades y preferencias del desarrollador.

1.2. COMPONENTES

1.2.1. Editor de código de programación

Es donde se escribe el código fuente de la aplicación.

Además de las características de un editor de textos normal y corriente puede presentar las siguientes peculiaridades:

- Numeración de las líneas de código.
- Reconoce la sintaxis del lenguaje de programación y la colorea.
- Permite detectar errores sintácticos.
- Autocompletado de código.

1.2.2. Compilador

Software que traduce un programa escrito en lenguaje de programación de alto nivel (C++, C, COBOL ...) a lenguaje máquina (lenguaje interpretable directamente por el ordenador).

1.2.3. Enlazador

Los enlazadores usan los resultados de las primeras etapas de compilación, otros recursos necesarios (como bibliotecas), quita los recursos que no necesita y enlaza el código generando un fichero ejecutable.

1.2.4. Intérprete

El intérprete, a diferencia del compilador, no convierte el lenguaje de programación a lenguaje máquina, la traducción se va realizando al mismo tiempo que se van ejecutando las instrucciones. Estos lenguajes son más lentos que los compilados porque tienen que traducirse mientras que se ejecutan, pero, son más flexibles.

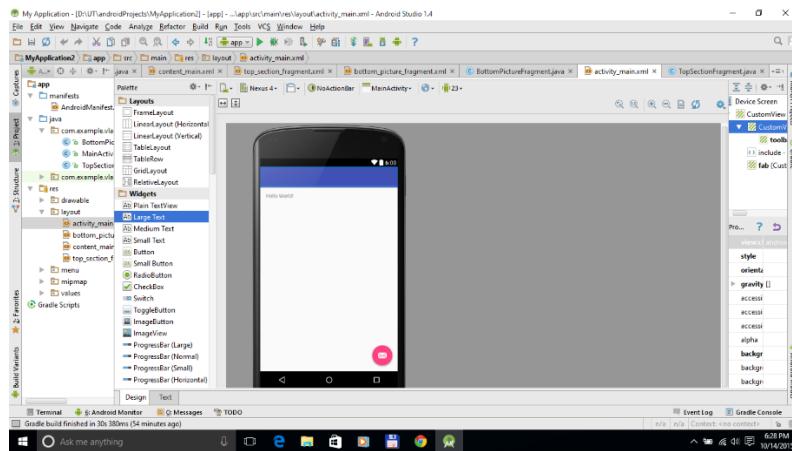
1.2.5. Depurador

Permite ejecutar el programa instrucción a instrucción para detectar los errores semánticos.

En cada instante permite mostrar el valor interno de las variables.

1.2.6. Constructor de interfaz gráfico

Permite crear y manipular componentes visuales.



1.2.7. Otras herramientas.

Como habéis podido deducir de su definición, el concepto de entorno de desarrollo es muy amplio, por tanto, en función del proyecto que estemos tratando, existen infinidad de herramientas que se pueden instalar e incorporar al entorno y que nos ayudarán en nuestro trabajo.

2. XAMPP

Xampp es una plataforma web, que se corresponde con un tipo de entorno de desarrollo empleado para diseñar y ejecutar un sitio web.

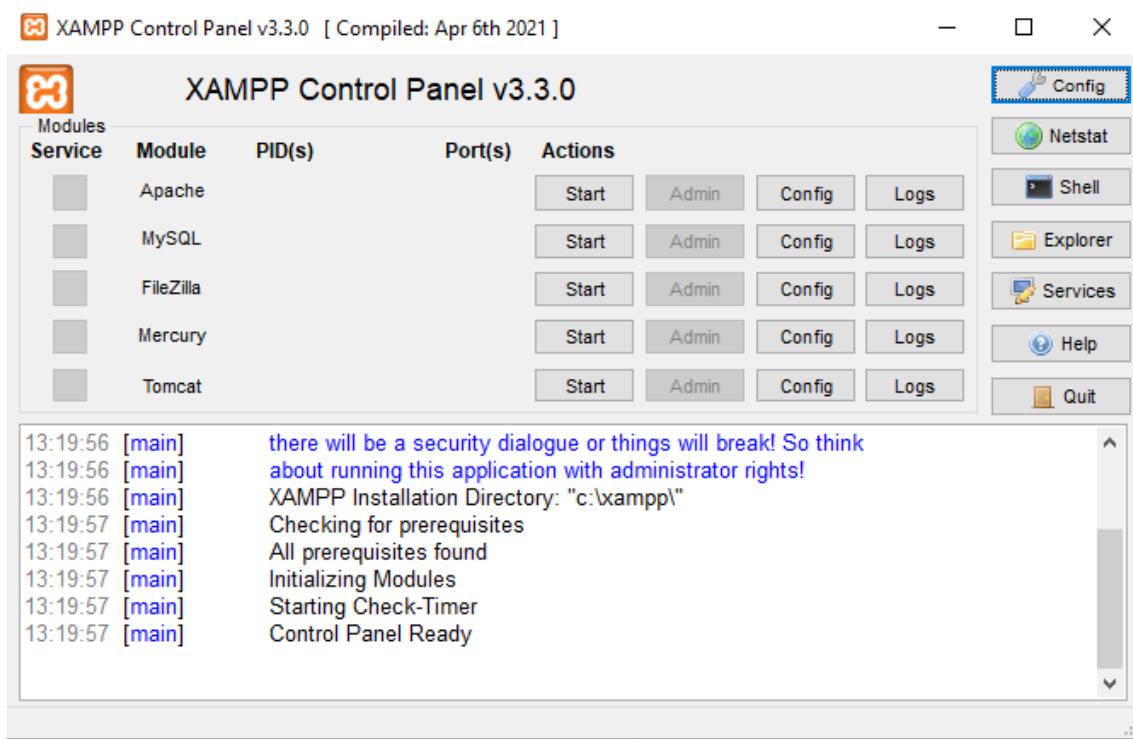
Hay muchos tipos de plataforma web, pero nosotros vamos a utilizar XAMP.

La palabra XAMPP viene de:

- **X** = Para cualquier sistema operativo
- **Apache**
- **MariaDB**
- **PHP**
- **PERL**

XAMPP es una de las mejores distribuciones de Apache que ayuda a los desarrolladores a crear un servidor web local con fines de prueba e implementación.

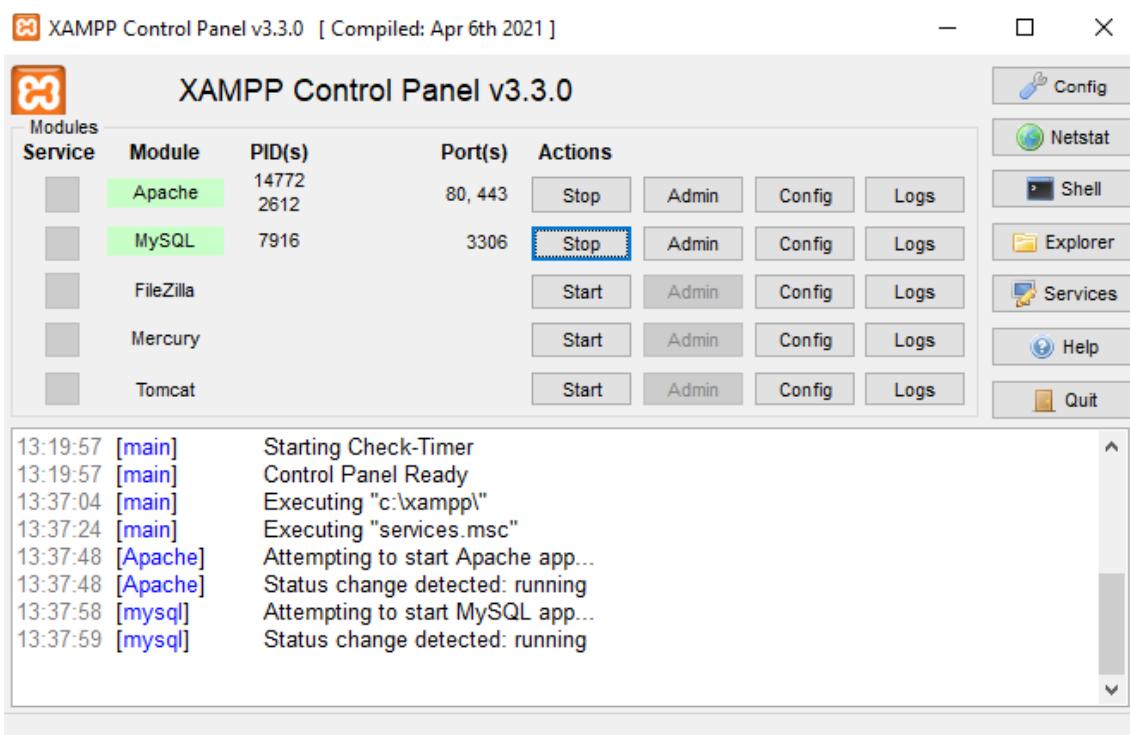
Fue diseñado para ser la forma más fácil de instalar y ejecutar un servidor de desarrollo. Dado que es multiplataforma, puede usarlo en Windows, macOS o Linux.



[Información sobre XAMPP](#)

Vemos que en este panel se nos muestran tres partes bien diferenciadas:

- El **panel de módulos** de XAMPP, donde se indica si el servicio está activo, el identificador del proceso que se está ejecutando (PID), el puerto que ocupa, unos botones para iniciar y detener los procesos, editar los archivos de configuración y abrir los archivos de registro de actividad.
- El **output** de Xampp que se muestra en la zona inferior.
- En la zona derecha se muestran las **herramientas** que permiten, entre otras cosas, cambiar la configuración de XAMPP, consultar los puertos ocupados, abrir un terminal, acceder directamente a la carpeta de xampp, consultar la ayuda del entorno o salir de la aplicación.



Un **proceso** es la ejecución de un programa o aplicación. Cada proceso se identifica por un número que es único en ese instante en la máquina, el **PID**.

Los **puertos** permiten dirigir la información que llega desde la web al proceso al que va destinada.

Instalar Xampp

Si existen otras aplicaciones lanzadas en los puertos por defecto de los servicios que utilizamos estos no se lanzarán.

Apache utiliza el puerto 80 para escuchar peticiones http y el puerto 443 para peticiones seguras (https).

Mysql tiene como puerto por defecto el 3306.

Para saber si un puerto está escuchado en Windows se usa el comando ***netstat -an | find ":puerto"***

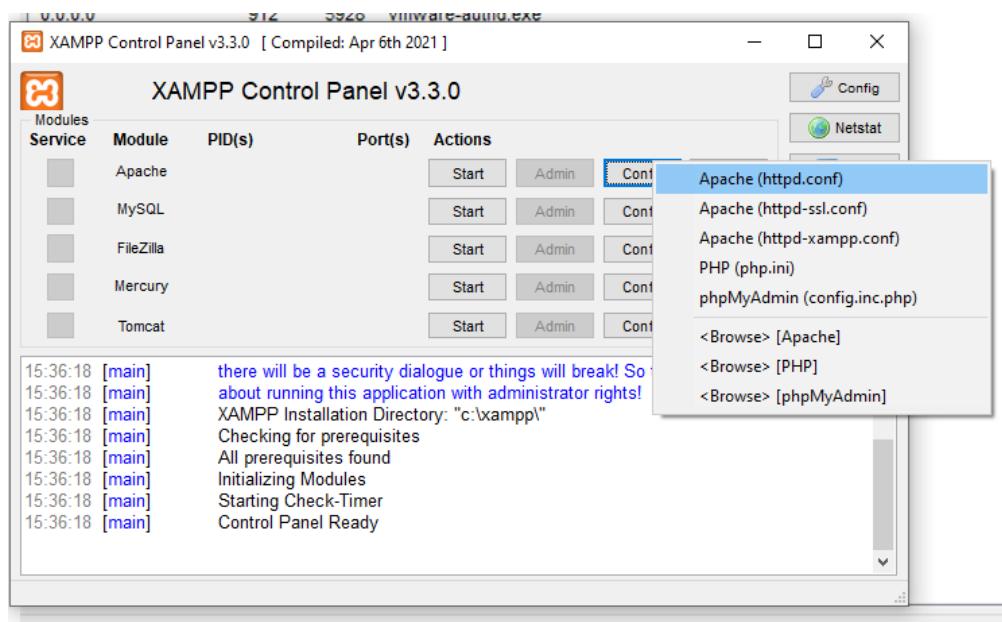
```
C:\Users\srunz>netstat -an | find ":9876"
C:\Users\srunz>netstat -an | find ":9876"
  TCP      0.0.0.0:9876              0.0.0.0:0                LISTENING
  TCP      [::]:9876                [::]:0                LISTENING
```

Se puede usar también el botón netstat de Xampp y ordenar por puerto.

Netstat - TCP Listening sockets			
Address	Port	PID	Name
0.0.0.0	80	14772	httpd.exe
0.0.0.0	135	1256	svchost.exe
192.168.0.28	139	4	System
192.168.56.1	139	4	System
192.168.136.1	139	4	System
192.168.157.1	139	4	System
0.0.0.0	443	14772	httpd.exe
0.0.0.0	445	4	System
0.0.0.0	902	5928	vmware-authd.exe
0.0.0.0	912	5928	vmware-authd.exe
0.0.0.0	3306	7916	mysqld.exe
0.0.0.0	5040	16356	svchost.exe
0.0.0.0	5357	4	System
127.0.0.1	5939	6088	TeamViewer_Service.exe
0.0.0.0	7680	4784	svchost.exe
0.0.0.0	49664	1000	lsass.exe
0.0.0.0	49665	908	wininit.exe
0.0.0.0	49666	1840	svchost.exe
0.0.0.0	49667	2096	svchost.exe
0.0.0.0	49668	5536	spoolsv.exe
0.0.0.0	49670	980	services.exe
192.168.0.28	55992	10908	Video.UI.exe
192.168.0.28	55993	10908	Video.UI.exe
192.168.0.28	55995	10908	Video.UI.exe
192.168.0.28	56487	18688	Spotify.exe
192.168.0.28	56517	7308	HxTsr.exe
192.168.0.28	56561	10860	SearchApp.exe
192.168.0.28	56562	10860	SearchApp.exe
192.168.0.28	56687	18688	Spotify.exe

Si nos ocurre el problema de que alguno de los puertos ya está ocupado habría que cambiarlo.

En cada módulo funciona de una manera, pero suele ser bastante intuitivo. Por ejemplo, si tuviéramos que cambiar el puerto 80 de apache por el 81 bastaría con modificar el fichero httpd.conf de la configuración de apache.



Habrá que cambiar la línea que dice que apache escucha en el puerto 80 por 81.

```

httpd.conf: Bloc de notas
Archivo Edición Formato Ver Ayuda
# other reason.
#
# Mutex default:logs
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80 highlighted

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule actions_module modules/mod_actions.so
<

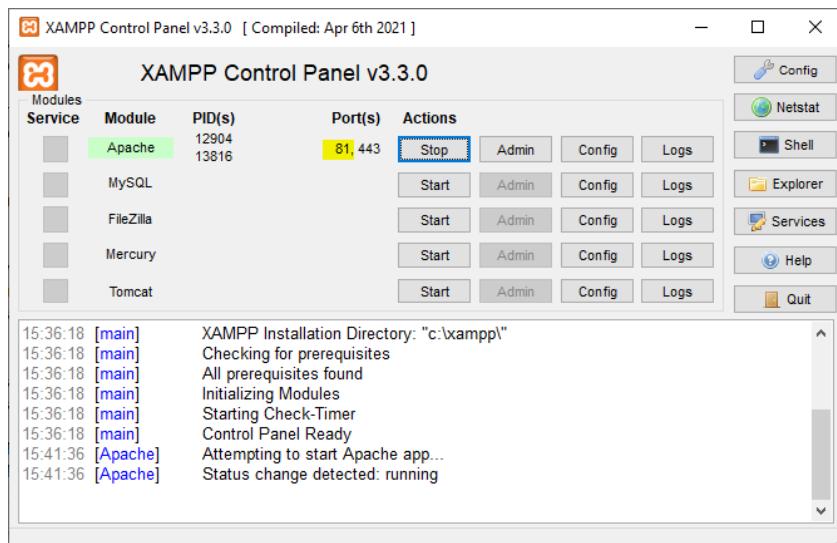
```

Y también el server name

```

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:80

```



Si modificáis tan solo este puerto sin tocar la configuración de xampp, para acceder al dashboard de Apache, tendréis que especificar la nueva dirección en la URL.

localhost:81/dashboard/

Normalmente, podemos acceder al administrador de Apache desde la url localhost.

The screenshot shows a web browser window with the URL `localhost/dashboard/` in the address bar. The page title is "XAMPP Apache + MariaDB + PHP + Perl". The main content area says "Welcome to XAMPP for Windows 8.0.12". It includes a message about successful installation and instructions for starting the server. At the bottom, there's a link to "Start the XAMPP Control Panel to check the server status."

Localhost es una palabra para definir de forma inequívoca el equipo que estamos utilizando, pero si usamos nuestra ip también podemos conectarnos a la dirección que nos despliega apache.



De hecho, si utilizamos la ip de un compañero que está conectado a la misma red también podremos acceder a su dashboard.



Desde este panel podéis consultar, entre otras cosas, la versión de apache que tenéis instalada y acceder a phpmyadmin.

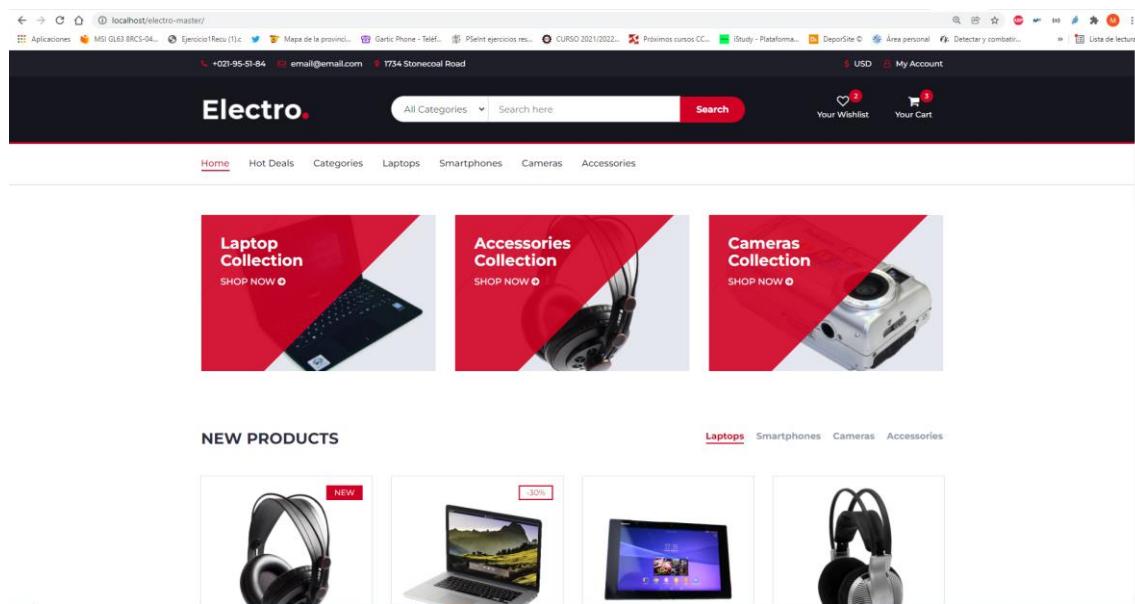
PHP Version 8.0.12	
System	Windows NT DESKTOP-G9ILJ6L 10.0 build 19044 (Windows 10) AMD64
Build Date	Oct 19 2021 11:17:33
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:script config.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\instantclient\ sdk\shared" "--with-oci8=19..\..\instantclient\ sdk\shared" "--enable-object-out-dir=.\obj" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930.TS.VS16
PHP Extension Build	API420200930.TS.VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress_zlib, compress_bz2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*., string.rot13, string.toupper, string.tolower, convert.*., consumed, dechunk, zlib.*., bz2.*

The screenshot shows the phpMyAdmin interface with several panels open:

- Configuraciones generales:** Shows the server connection collation set to utf8mb4_unicode_ci.
- Configuraciones de apariencia:** Shows the language set to Spanish and the theme set to pmahomme.
- Servidor de base de datos:** Displays information about the MySQL server, including its IP (127.0.0.1), type (MariaDB), and version (10.4.21-MariaDB).
- Servidor web:** Displays information about the Apache server, including its version (2.4.51), PHP version (8.0.12), and extensions (libmysql - mysqli - myqdnd).
- phpMyAdmin:** A sidebar with links to documentation, support, and license information.

PhpMyAdmin es una herramienta web que nos permite gestionar bases de datos de MySQL.

Si queremos desplegar un programa de PHP que hayamos desarrollado en el apache de XAMPP simplemente tendremos que añadirlo en la carpeta .htdocs. y acceder a él mediante la url pertinente.



Para probarlo descargad una plantilla gratuita.

Aquí hay muchas: <https://www.templatemonster.com/free-templates/responsive-business-website-template-52524.html?aff=ipixeestudio>

Yo he elegido esta: <https://themewagon.com/themes/free-bootstrap-ecommerce-template-electro/>

EJERCICIO 1: Despliega un proyecto en XAMPP.

EJERCICIO 2: Accede al proyecto desplegado de un compañero.

3. GESTOR DE BASES DE DATOS

PhpMyAdmin es una aplicación web que nos permite administrar bases de datos de MySQL de manera sencilla y con una interfaz amistosa.

3.1. Crear base de datos

Bases de datos

Crear base de datos

prueba_datos utf8mb4_general_ci Crear

Base de datos	Cotejamiento	Acción
ejercicio_java_ee	utf8_general_ci	Seleccionar privilegios
information_schema	utf8_general_ci	Seleccionar privilegios
mysql	utf8mb4_general_ci	Seleccionar privilegios
performance_schema	utf8_general_ci	Seleccionar privilegios
phpmyadmin	utf8_bin	Seleccionar privilegios
prueba1	utf8mb4_general_ci	Seleccionar privilegios
test	latin1_swedish_ci	Seleccionar privilegios

Total: 7

↑ Seleccionar todo Para los elementos que están marcados: Eliminar

⚠ Nota: Activar aquí las estadísticas de la base de datos podría causar tráfico pesado entre el servidor web y el servidor MySQL.

- Activar estadísticas

3.2. Crear tabla

Servidor: 127.0.0.1 > Base de datos: prueba_datos

Estructura SQL Buscar Generar una consulta Exportar Importar

No se han encontrado tablas en la base de datos.

Crear tabla

Nombre: jugadores Número de columnas: 4

Continuar

Nombre de la tabla: jugadores Agregar 1 columnas(s) Continuar

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice	A_J	Comentarios	Virtualidad
id	INT		Ninguno			PRIMARY	<input checked="" type="checkbox"/>			
nombre	VARCHAR		Ninguno							
apellidos	VARCHAR		Ninguno							
edad	INT		Ninguno							

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)		No	Ninguna			AUTO_INCREMENT	Cambiar Eliminar Más
2	nombre	varchar(50)	utf8mb4_general_ci	No	Ninguna			Cambiar Eliminar Más	
3	apellidos	varchar(50)	utf8mb4_general_ci	No	Ninguna			Cambiar Eliminar Más	
4	edad	int(2)		No	Ninguna			Cambiar Eliminar Más	

3.3. Añadir registros

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento Disparadores

Columna	Tipo	Función	Nulo	Valor
id	int(11)			
nombre	varchar(50)			Gerard
apellidos	varchar(50)			Moreno
edad	int(2)			29

Ignorar

Columna	Tipo	Función	Nulo	Valor
id	int(11)			
nombre	varchar(50)			Pau
apellidos	varchar(50)			Torres
edad	int(2)			25

Insertar como una nueva fila y luego Volver

Continuar inserción con 2 filas

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Examinar, Estructura, SQL, Buscar, Insertar, Exportar, Importar, Privilegios, Operaciones.
- Status Bar:** Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,00009 segundos.)
- Query Editor:** SELECT * FROM `jugadores`
- Table View:** Shows two rows of data: Gerard Moreno (id 1) and Pau Torres (id 2).
- Buttons:** Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar].
- Filter and Sort:** Mostrar todo, Número de filas: 25, Filtrar filas: Buscar en esta tabla, Sort by key: Ninguna.
- Operations:** + Opciones, Editar, Copiar, Borrar.
- Buttons at the bottom:** Seleccionar todo, Para los elementos que están marcados: Editar, Copiar, Borrar, Exportar.
- Bottom Filter and Sort:** Mostrar todo, Número de filas: 25, Filtrar filas: Buscar en esta tabla, Sort by key: Ninguna.
- Buttons at the bottom:** Imprimir, Copiar al portapapeles, Exportar, Mostrar gráfico, Crear vista.
- Save Options:** Guardar esta consulta en favoritos, Etiqueta: [empty], Permitir que todo usuario pueda acceder a este favorito.
- Buttons at the bottom:** Guardar esta consulta en favoritos.

3.4. Usar SQL

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Examinar, Estructura, SQL, Buscar, Insertar, Exportar, Importar, Privilegios, Operaciones, Seguimiento, Disparadores.
- SQL Editor:** Ejecutar la(s) consulta(s) SQL en la tabla prueba_datos.jugadores:
- SQL Statement:** 1 INSERT INTO `jugadores` (`nombre`, `apellidos`, `edad`) VALUES ('Lucas', 'Ocampos', '27'), ('Radamel', 'Falcao', '35')
- Buttons at the bottom:** SELECT*, SELECT, INSERT, UPDATE, DELETE, Limpiar, Formato, Obtener consulta almacenada automáticamente.
- Checkboxes at the bottom:** Enlazar parámetros
- Table View:** Shows four rows of data: Gerard Moreno (id 1), Pau Torres (id 2), Lucas Ocampos (id 3), and Radamel Falcao (id 4).
- Buttons at the bottom:** + Opciones, Editar, Copiar, Borrar.

3.5. Eliminar base de datos

Bases de datos

Nombre de la base de dato: utf8mb4_general_ci

Base de datos	Cotejamiento	Acción
ejercicio_java_ee	utf8_general_ci	[+] Seleccionar privilegios
information_schema	utf8_general_ci	[+] Seleccionar privilegios
mysql	utf8mb4_general_ci	[+] Seleccionar privilegios
performance_schema	utf8_general_ci	[+] Seleccionar privilegios
phpmyadmin	utf8_bin	[+] Seleccionar privilegios
prueba1	utf8mb4_general_ci	[+] Seleccionar privilegios
prueba_datos	utf8mb4_general_ci	[+] Seleccionar privilegios
test	latin1_swedish_ci	[+] Seleccionar privilegios
universidad	utf8mb4_general_ci	[+] Seleccionar privilegios

Total: 9

↑ [] Seleccionar todo Para los elementos que están marcados: [] Eliminar

⚠ Nota: Activar aquí las estadísticas de la base de datos podría causar tráfico pesado entre el servidor web y el servidor MySQL.

- Activar estadísticas

EJERCICIO 3: Crea la base de datos que se muestra en los apuntes.

3.6. Importar base de datos

Script

Importando al servidor actual

Archivo a importar:

Importación parcial:

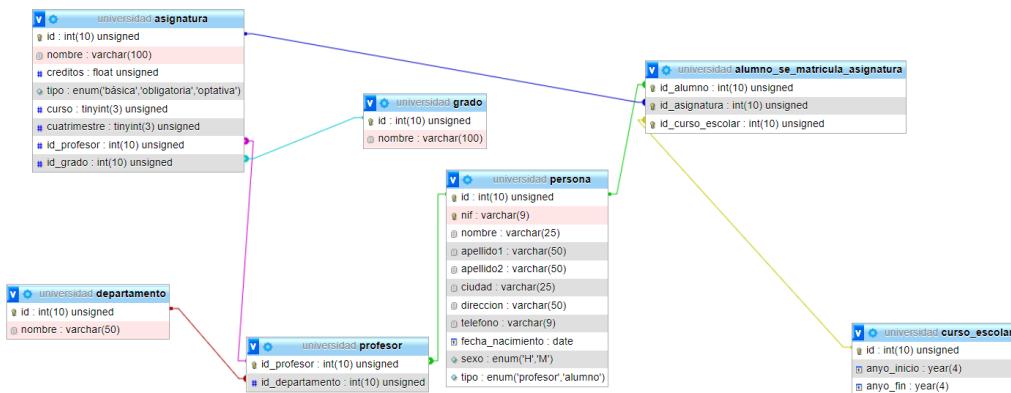
Otras opciones:

Formato:

Opciones específicas al formato:

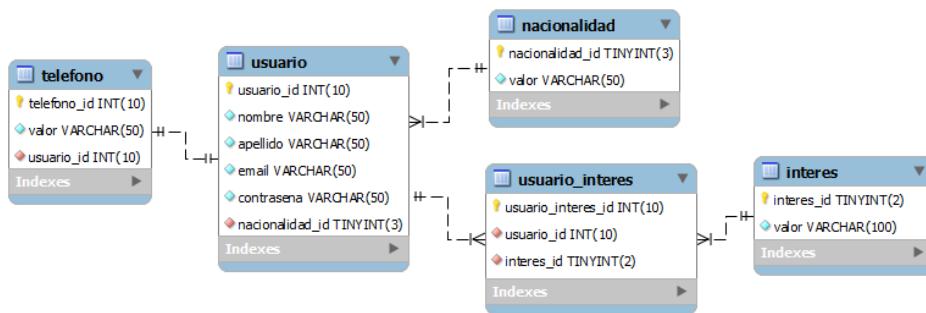
Consola

3.7. Pestaña de diseñador



EJERCICIO 4: Importa la base de datos del script adjuntado en los apuntes y visualízala en la pestaña del diseñador.

EJERCICIO 5: Crea la siguiente base de datos.

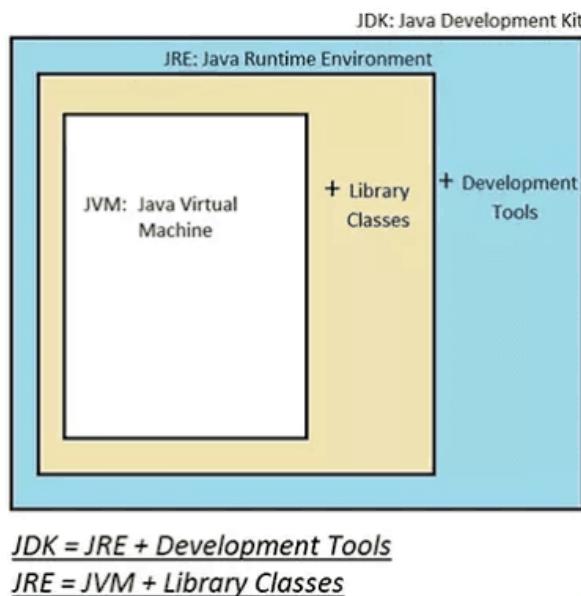


4.JAVA

Para poder instalar un entorno de desarrollo que utilice java antes tendremos que instalar Java en el equipo.

Un usuario sólo necesita el **JRE** (Java Runtime Environment) para ejecutar las aplicaciones desarrolladas en lenguaje Java, mientras que para desarrollar nuevas aplicaciones en dicho lenguaje es necesario un entorno de desarrollo, denominado **JDK** (Java Development Kit).

Java se ejecuta dentro de una máquina virtual **JVM** (Java Virtual Machine), de esta manera, la ejecución del programa es independiente del entorno en el que se ejecute.



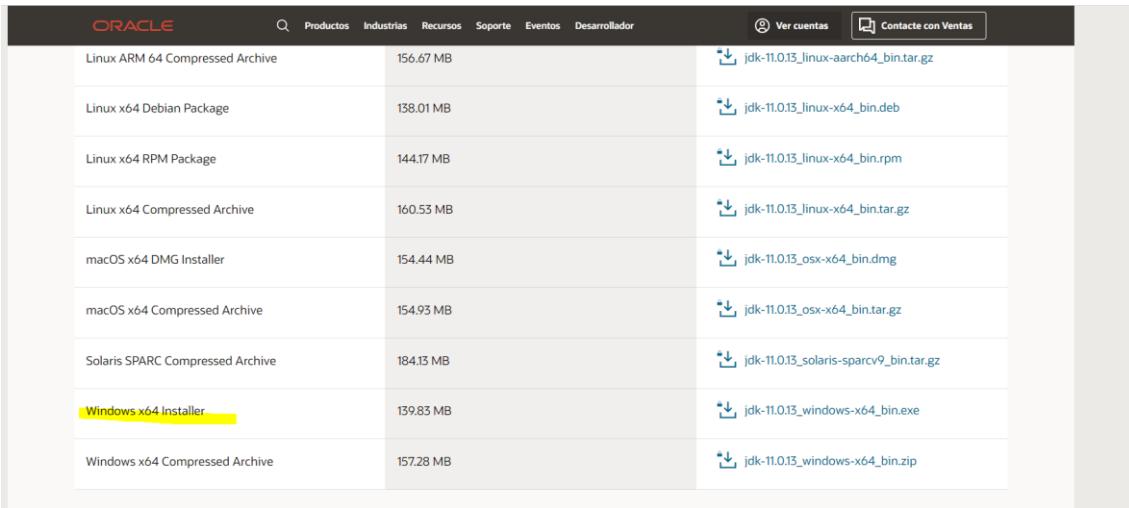
Cabe destacar que dentro de Java las versiones que os descarguéis pertenecerán a Oracle JDK o a Open JDK.

Oracle JDK está desarrollado por la empresa Oracle. Desde 2019, si una empresa quiere usar uno de estos jdk's tiene que pagar una licencia. Si estamos usando el jdk con fines educacionales no es necesario.

Open JDK es una implementación de código abierto y gratuita.

Realmente no hay una diferencia de rendimiento significativa entre ambas plataformas, la principal diferencia, es que en el caso de los errores que se detectan en Oracle JDK son corregidos con mayor rapidez y en el OpenSource debemos esperar algo más.

Se puede descargar, por tanto, el jdk desde la página oficial de Open JDK [Web Open JDK](#) o desde la página oficial de Oracle [Oracle JDK Java 11](#)



ORACLE	Productos	Industrias	Recursos	Soporte	Eventos	Desarrollador	Ver cuentas	Contacte con Ventas
Linux ARM 64 Compressed Archive	156.67 MB						 jdk-11.0.13_linux-aarch64_bin.tar.gz	
Linux x64 Debian Package	138.01 MB						 jdk-11.0.13_linux-x64_bin.deb	
Linux x64 RPM Package	144.17 MB						 jdk-11.0.13_linux-x64_bin.rpm	
Linux x64 Compressed Archive	160.53 MB						 jdk-11.0.13_linux-x64_bin.tar.gz	
macOS x64 DMG Installer	154.44 MB						 jdk-11.0.13_osx-x64_bin.dmg	
macOS x64 Compressed Archive	154.93 MB						 jdk-11.0.13_osx-x64_bin.tar.gz	
Solaris SPARC Compressed Archive	184.13 MB						 jdk-11.0.13_solaris-sparcv9_bin.tar.gz	
Windows x64 Installer	139.83 MB						 jdk-11.0.13_windows-x64_bin.exe	
Windows x64 Compressed Archive	157.28 MB						 jdk-11.0.13_windows-x64_bin.zip	

Una vez esté instalado Java podemos instalar el entorno de desarrollo.

5. MANUAL DE ECLIPSE

A continuación, os he redactado los puntos importantes del uso del entorno de desarrollo Eclipse como un manual.

Como ejercicio del tema os encargaré realizar un manual de un entorno que no sea eclipse y de vuestra libre elección que tendrá que seguir un esquema análogo al que os mostraré aquí.

EJERCICIO 6: Desarrolla un manual de un Entorno de Desarrollo diferente a Eclipse.

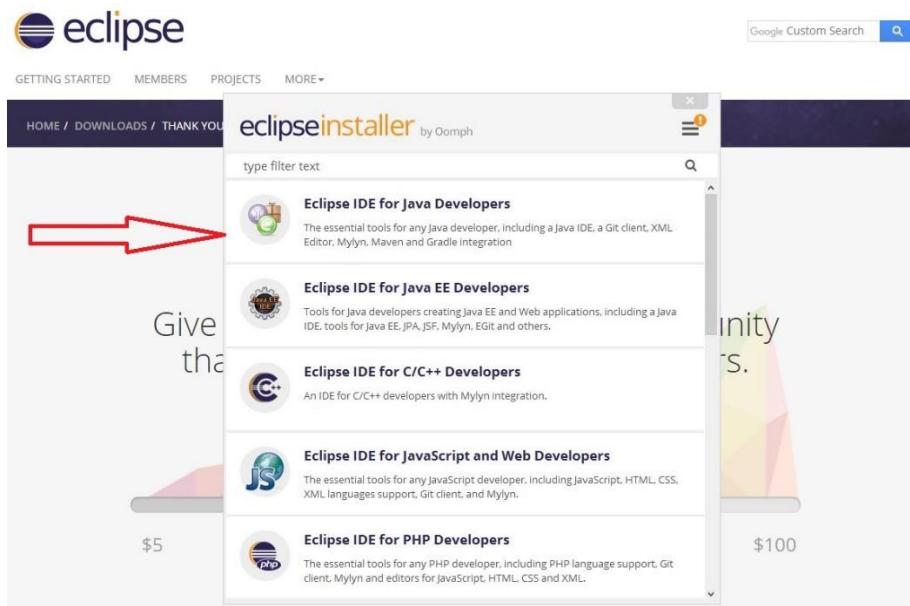
5.1. Instalación

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

Eclipse es un entorno de desarrollo de código abierto y que fue diseñado para ser extendido de forma indefinida a través de plug-ins.

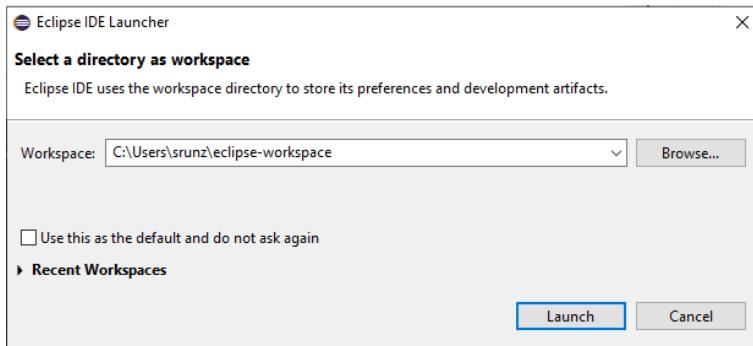
Fue ideado con la concepción de que fuera un IDE genérico (no se desarrolló para ningún lenguaje específico) pero es muy utilizado para la elaboración de programas en Java.

Vamos a instalar la versión de java más ligera ya que, todas las funcionalidades que le queramos agregar posteriormente, podemos añadirlas con plug-ins.

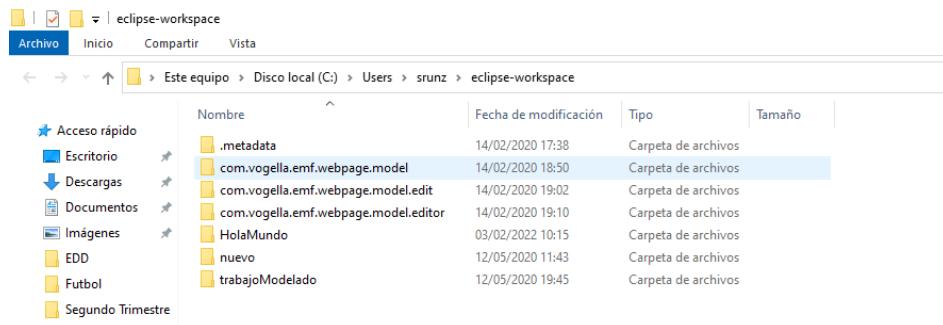


5.2. Elección de workspace

Al ejecutarlo, Eclipse nos pregunta por el espacio de trabajo en el que tenemos almacenados nuestros proyectos y nos permite elegir el directorio por defecto o seleccionar otro que deseemos.



La carpeta que indiquemos es en la que se almacenarán dentro de nuestro ordenador los proyectos que vayamos haciendo.



EJERCICIO 7: Localiza tu workspace de Eclipse y modifícalo.

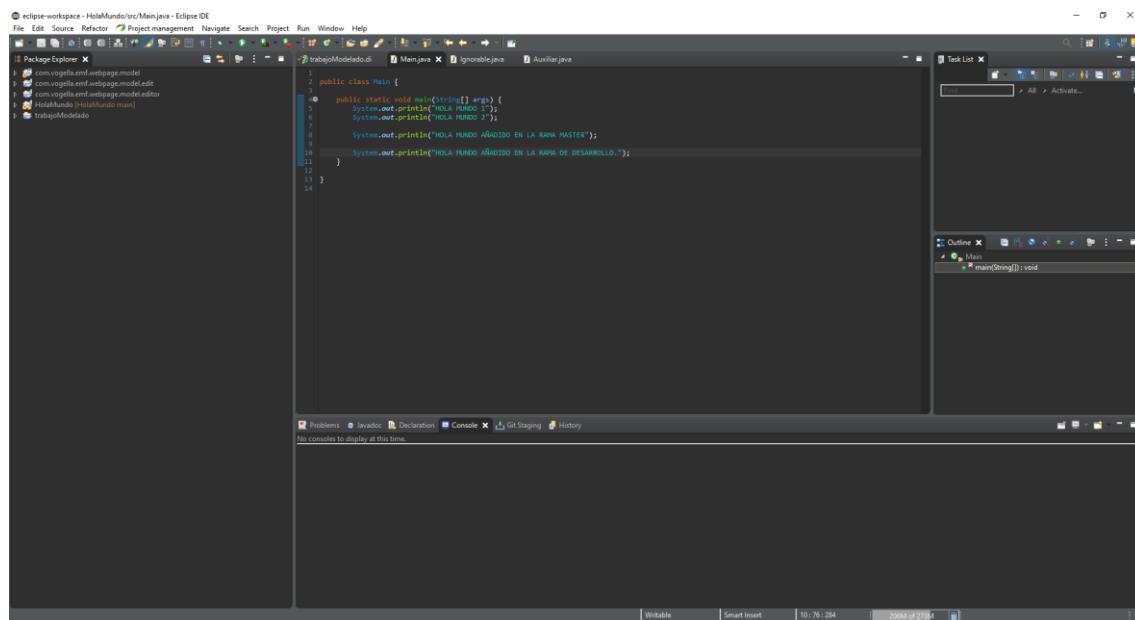
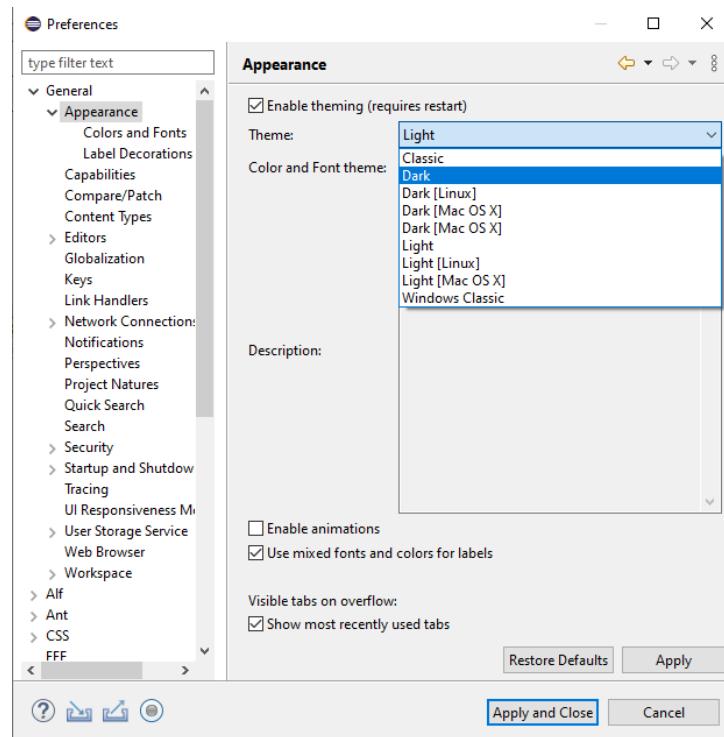
5.3. Personalización del entorno

Los programadores nos pasamos gran parte de nuestro día delante de un ordenador.

Existen estudios médicos que explican por qué programar sobre un fondo oscuro es menos dañino para la vista que hacerlo sobre un fondo blanco [¿Por qué los programadores usan fondos oscuros?](#).

Por eso, lo primero que se debe hacer cuando se descargue un entorno nuevo, es personalizarlo para que esté a nuestro gusto.

Window>preferences

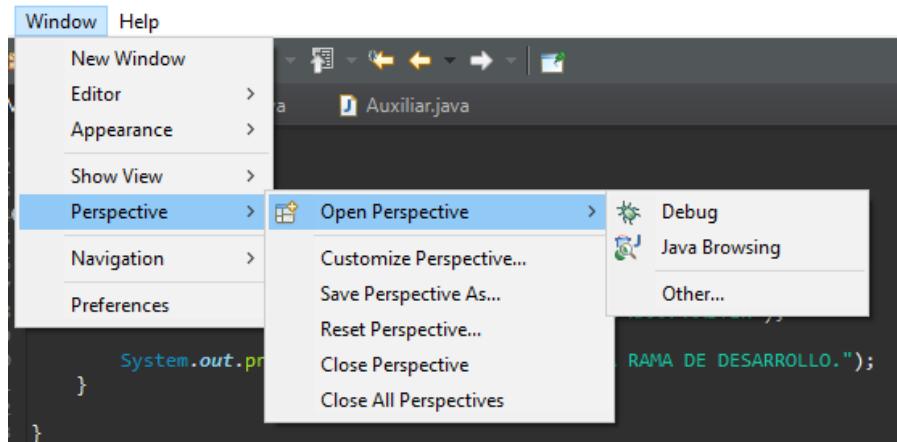


EJERCICIO 8: Pon un fondo negro a Eclipse.

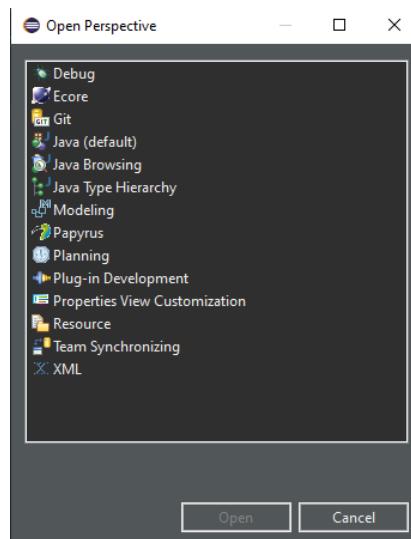
5.4. Perspectivas y vistas

Una **perspectiva** de Eclipse es una agrupación de vistas y editores de manera que den apoyo a una actividad completa del proceso de desarrollo software.

Sin embargo, es posible crear perspectivas propias añadiendo nuevas vistas y cambiando su distribución en la pantalla.



En other mostrará todas las perspectivas que hay en el entorno.



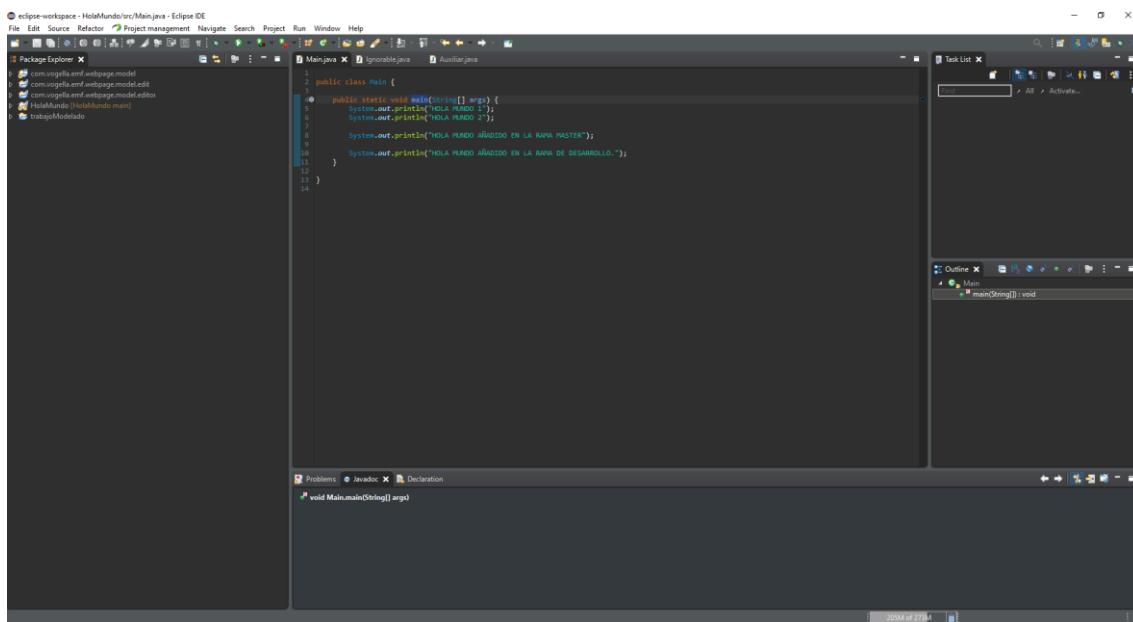
La que se usa normalmente es la de Java (default).

También se puede seleccionar la perspectiva desde el menú que aparece arriba a la derecha.



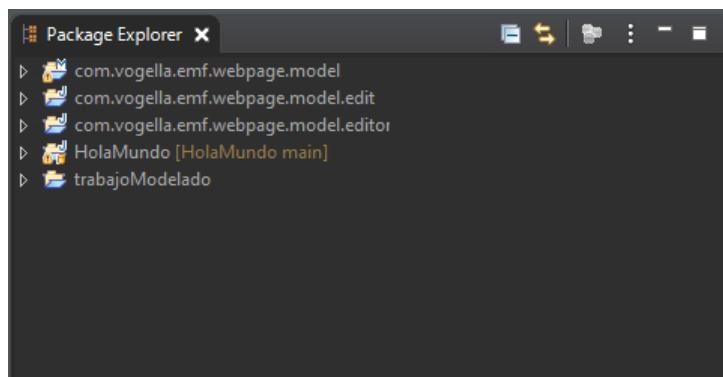
Si modificáis mucho una perspectiva y luego no sabéis como estaba al principio siempre podéis dar al botón de reset perspective y volverá a su estado inicial.

El Java default tiene este aspecto.



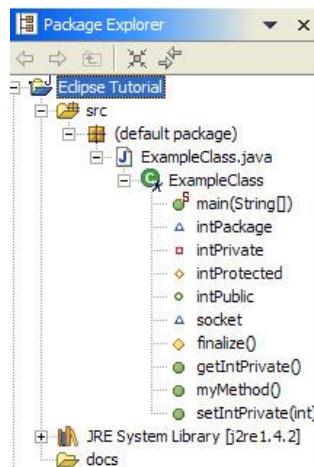
Las **vistas** proporcionan funcionalidades de apoyo al programador que usa el entorno.

Por ejemplo, en la perspectiva de Java (Default) tenemos una vista que se llama **explorador de paquetes** y contiene el esquema por el cual se organizan los proyectos, paquetes y clases de Java.



La vista del explorador de paquetes muestra la estructura lógica de paquetes, clases, ficheros incluso se pueden llegar a ver los métodos de Java.

Esta distribución variará en función de la tecnología y el lenguaje de programación que estemos usando.



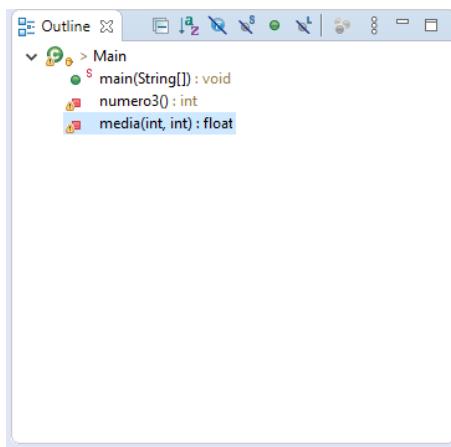
Voy a modificar un poco el código para mostrarlos otras vistas.

```

1① /**
2  * <h2>Esta clase se utilizará para enseñar a los alumnos a usar Eclipse y Git.
3  * </h2>
4  *
5  * @version 1
6  * @author sruzu15
7  *
8  */
9 public class Main {
10
11    public static void main(String[] args) {
12        System.out.println("HOLA MUNDO 1");
13        System.out.println("HOLA MUNDO 2");
14
15        System.out.println("HOLA MUNDO AÑADIDO EN LA RAMA MASTER");
16
17        System.out.println("HOLA MUNDO AÑADIDO EN LA RAMA DE DESARROLLO.");
18    }
19
20    /**
21     * Devuelve el número 3.
22     *
23     * @return Número 3.
24     */
25    private int numero3() {
26        return 3;
27    }
28
29    /**
30     * Devuelve la media de los números pasados como parámetro.
31     *
32     * @param n1 primer número
33     * @param n2 segundo número
34     * @return media de los números
35     */
36    private float media(int n1, int n2) {
37        return (float) ((n1 + n2) / 2);
38    }
39 }
40

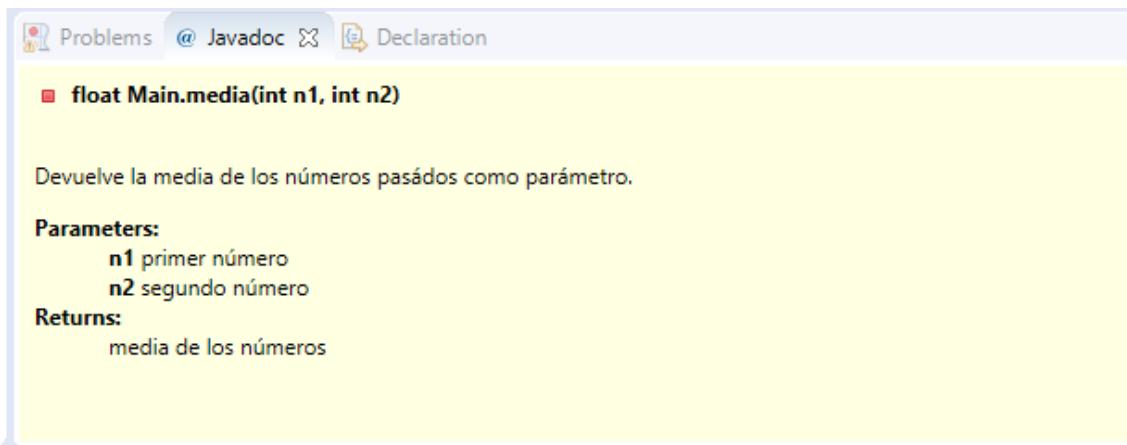
```

La vista **resumen** (outline) nos permite visualizar de manera rápida los métodos y atributos que se encuentran definidos dentro de la clase de java.

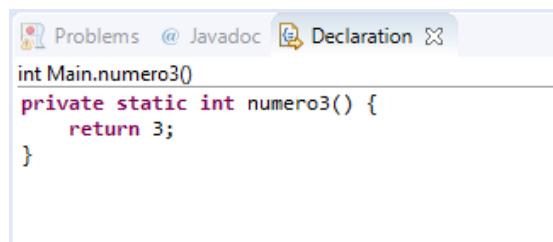


Si pulsamos en cualquiera de estos elementos, nos llevará a la línea de código en la que se encuentra.

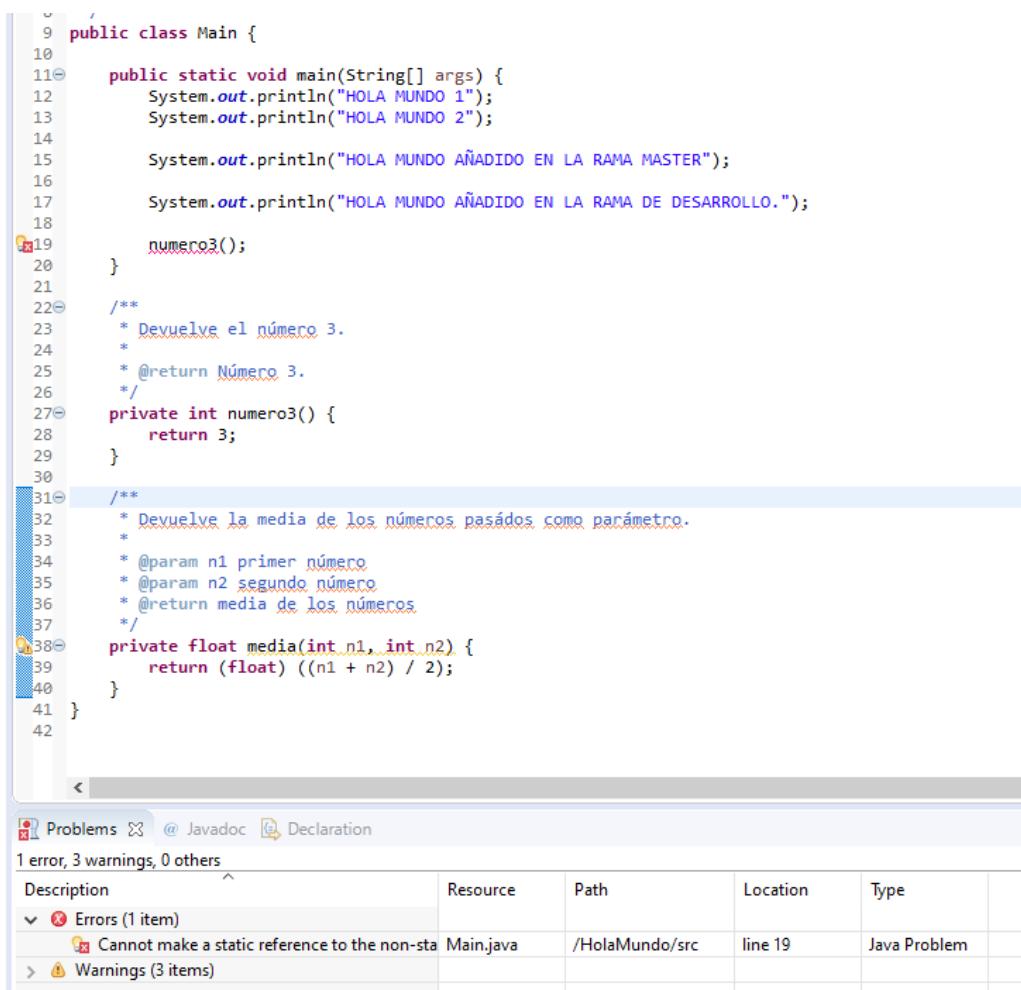
La vista **Javadoc** nos permite ver el aspecto del Javadoc de la clase o método en el que nos encontremos.



La vista de **declaración** nos permite ver rápidamente la declaración de una variable o un método.



La vista **problemas** nos permite ver los problemas sintácticos del programa.



```

9  public class Main {
10
11     public static void main(String[] args) {
12         System.out.println("HOLA MUNDO 1");
13         System.out.println("HOLA MUNDO 2");
14
15         System.out.println("HOLA MUNDO AÑADIDO EN LA RAMA MASTER");
16
17         System.out.println("HOLA MUNDO AÑADIDO EN LA RAMA DE DESARROLLO.");
18
19     numero3();
20 }
21
22 /**
23 * Devuelve el número 3.
24 *
25 * @return Número 3.
26 */
27 private int numero3() {
28     return 3;
29 }
30
31 /**
32 * Devuelve la media de los números pasados como parámetro.
33 *
34 * @param n1 primer número
35 * @param n2 segundo número
36 * @return media de los números.
37 */
38 private float media(int n1, int n2) {
39     return (float) ((n1 + n2) / 2);
40 }
41 }
42

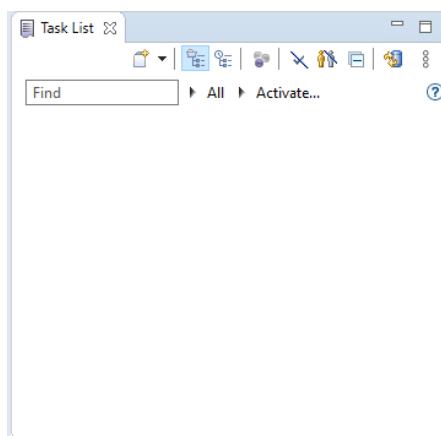
```

Problems @ Javadoc Declaration

1 error, 3 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (1 item)				
Cannot make a static reference to the non-static method numero3() from type Main	Main.java	/HolaMundo/src	line 19	Java Problem
Warnings (3 items)				

La **lista de tareas** nos permite gestionar notas a modo de recordatorio.

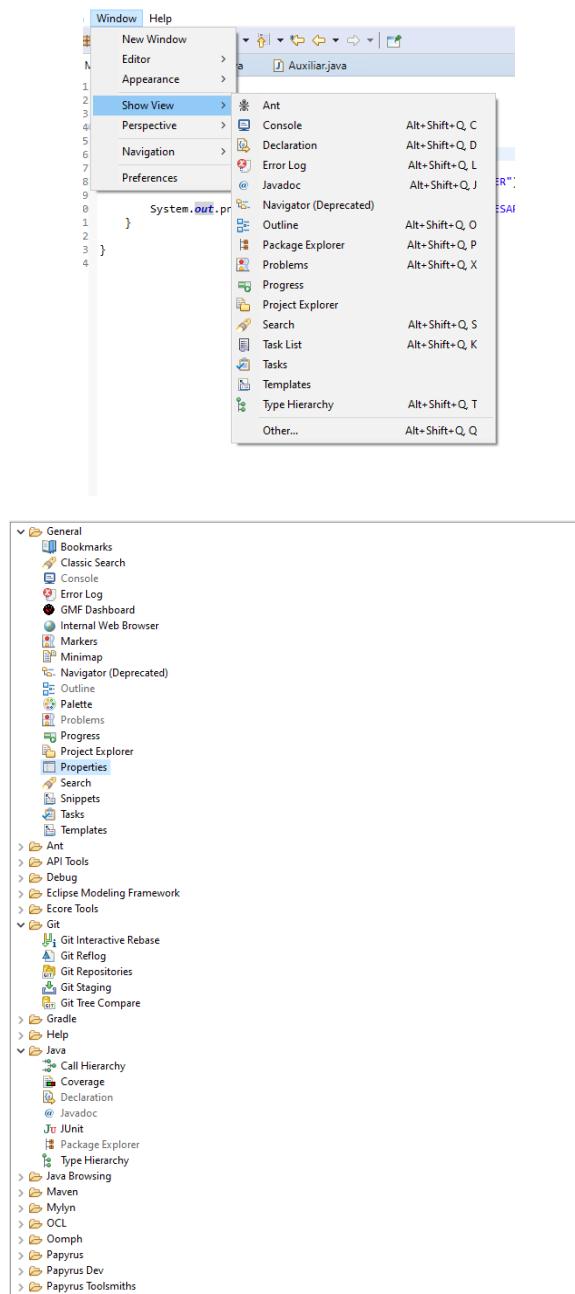


Podemos abrir vistas deliberadamente o también se irán abriendo en función de las acciones que realicemos.

Por ejemplo, si ahora ejecutamos la aplicación se mostrará la vista **consola** que mostrará la salida por pantalla de Java y nos permitirá parar los procesos en ejecución.

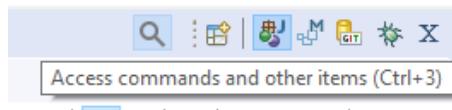


Podéis abrir a mano las vistas.



Como veis hay muchísimas vistas y eso que aún no hemos instalado plugins.

Podéis buscar directamente las vistas desde la lupa de búsqueda que ofrece Eclipse.

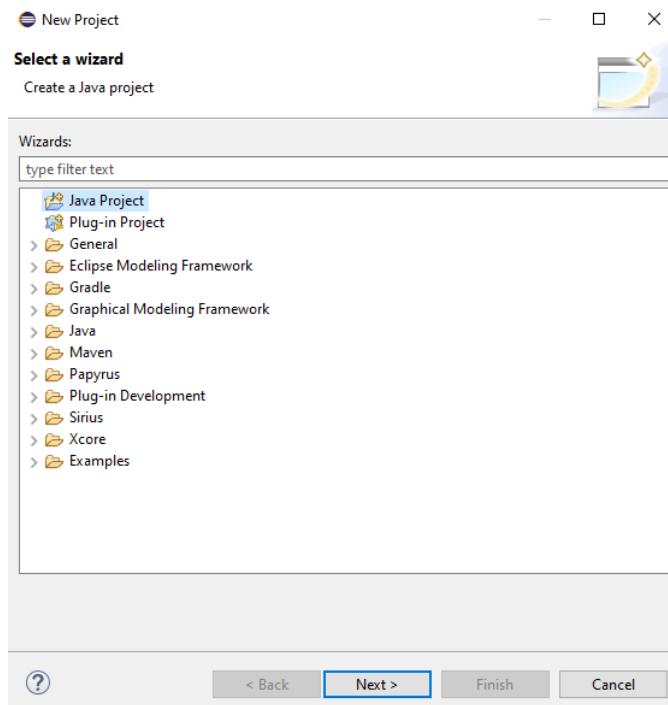


EJERCICIO 9: Haz un resumen sobre cómo se estructuran las opciones en Eclipse e indica las vistas que te parecen más útiles.

5.5. Crear un proyecto

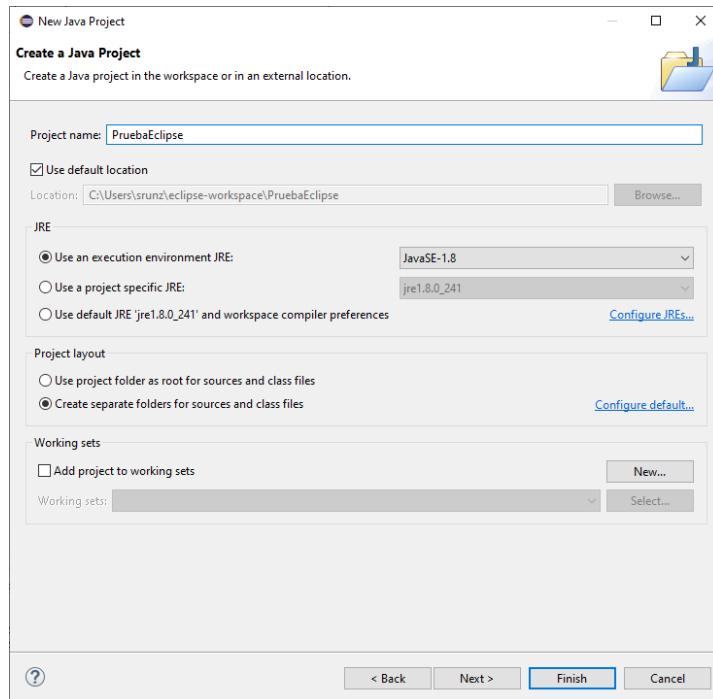
Para crear un proyecto seleccionamos la opción de proyecto y elegimos el tipo de proyecto que queremos crear.

En este caso será un proyecto de Java.

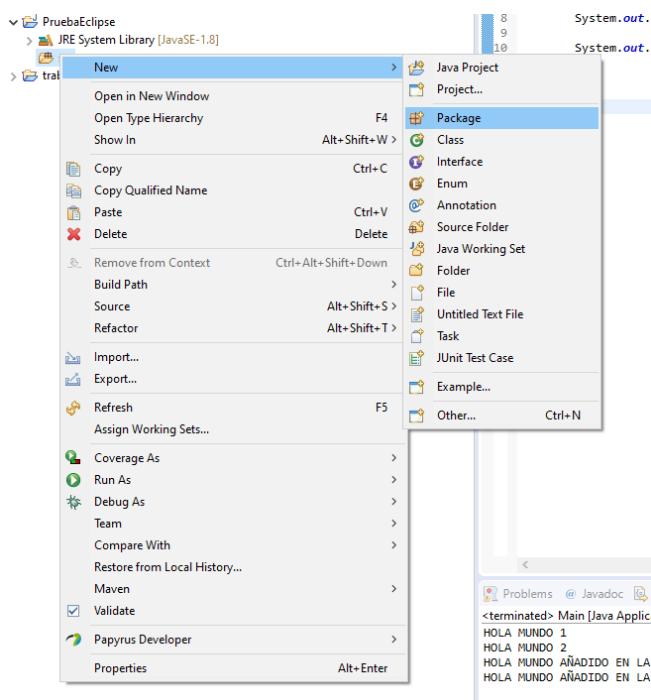


Si queremos usar algún framework concreto o un tipo de proyecto que no aparece en esta lista seguramente sea necesario instalar un plug-in.

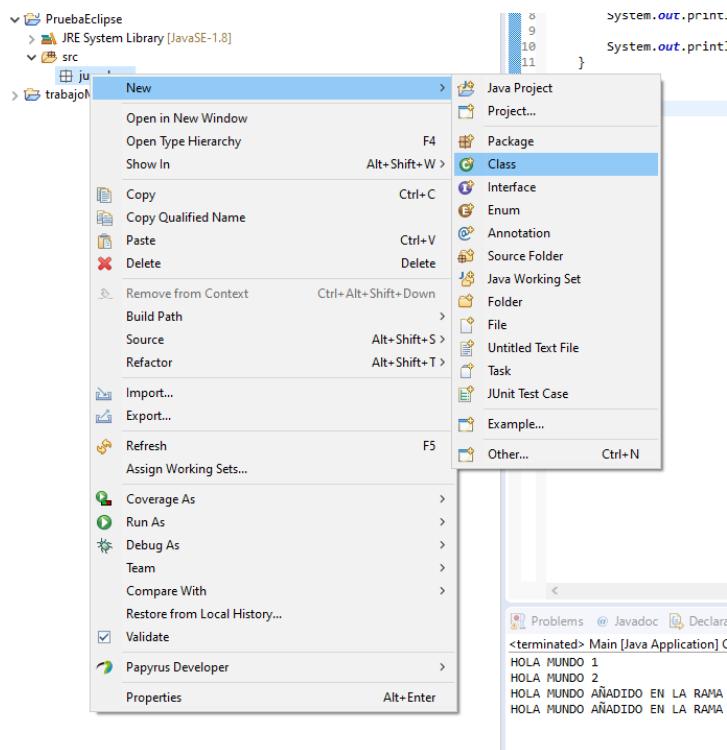
Creamos un proyecto de prueba.



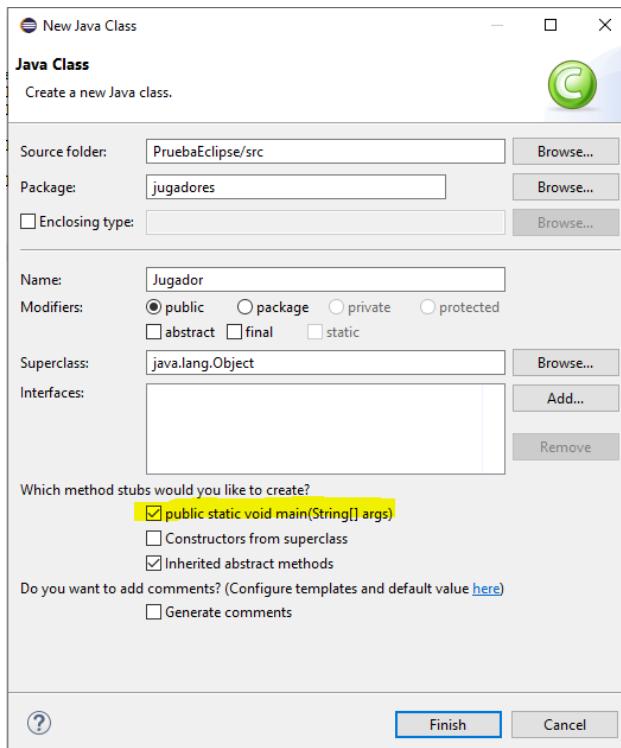
Cuando creamos el proyecto tendremos que añadirle un paquete y una clase de Java.



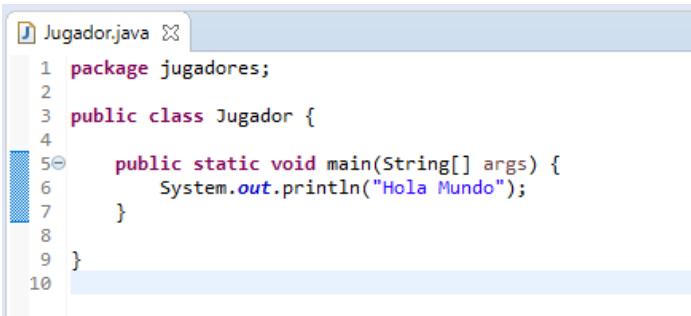
Añadimos una clase.



Si queremos que la clase sea el main (como es el caso) hay que marcar el siguiente checkbox.



Cuando creamos la clase la convertimos en un hola mundo.



```
Jugador.java
1 package jugadores;
2
3 public class Jugador {
4
5     public static void main(String[] args) {
6         System.out.println("Hola Mundo");
7     }
8
9 }
10
```

El editor de texto nos mostrará los ficheros que estén abiertos con pestañas como en un navegador.

Para ejecutar el proyecto solo hay que pulsar el botón de run del menú.



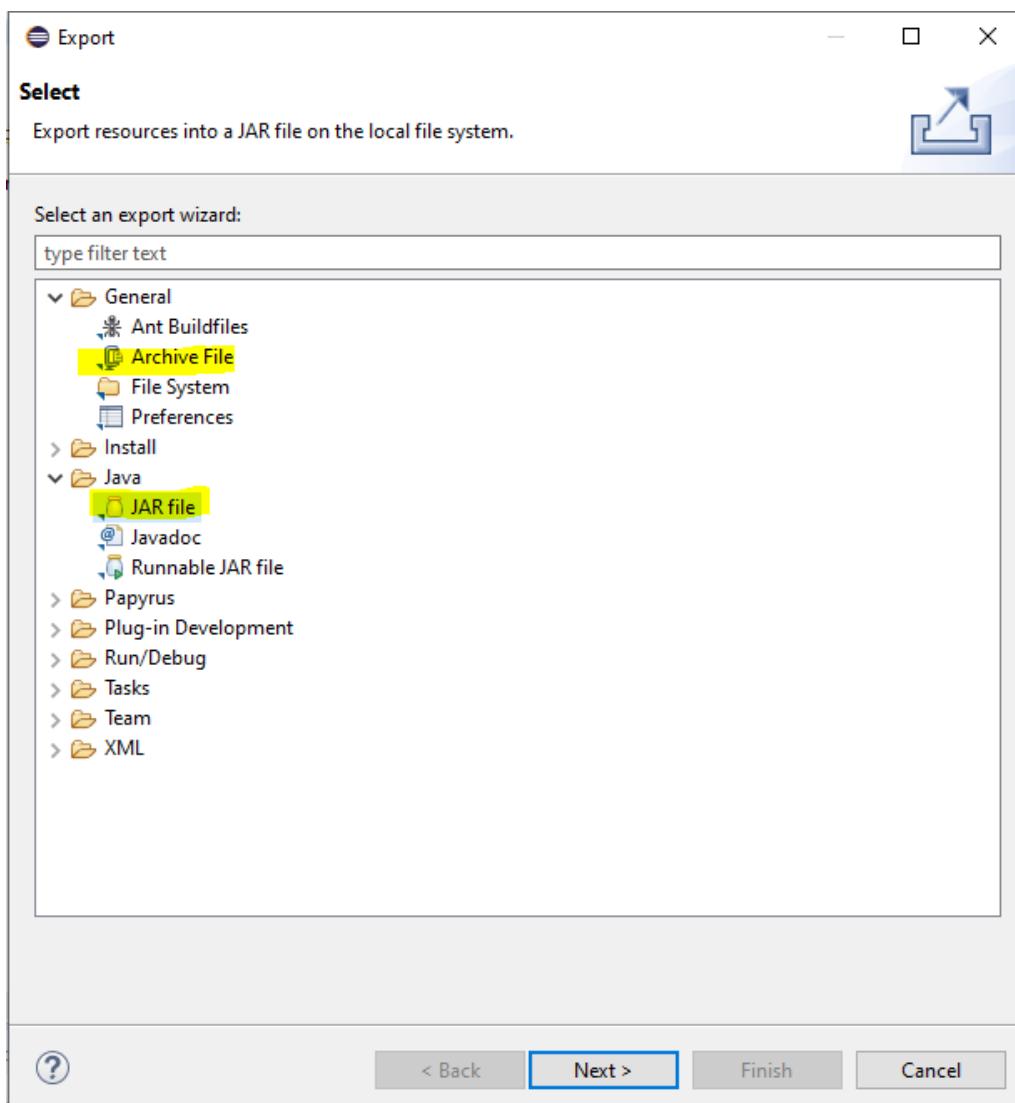
El output se mostrará en la ventana de consola.

5.6. Exportar/Importar proyectos

Exportar nos permite crear un archivo que contenga el proyecto, ya sea para utilizarlo directamente en el entorno de explotación o para importarlo en otro entorno y continuar con la edición.

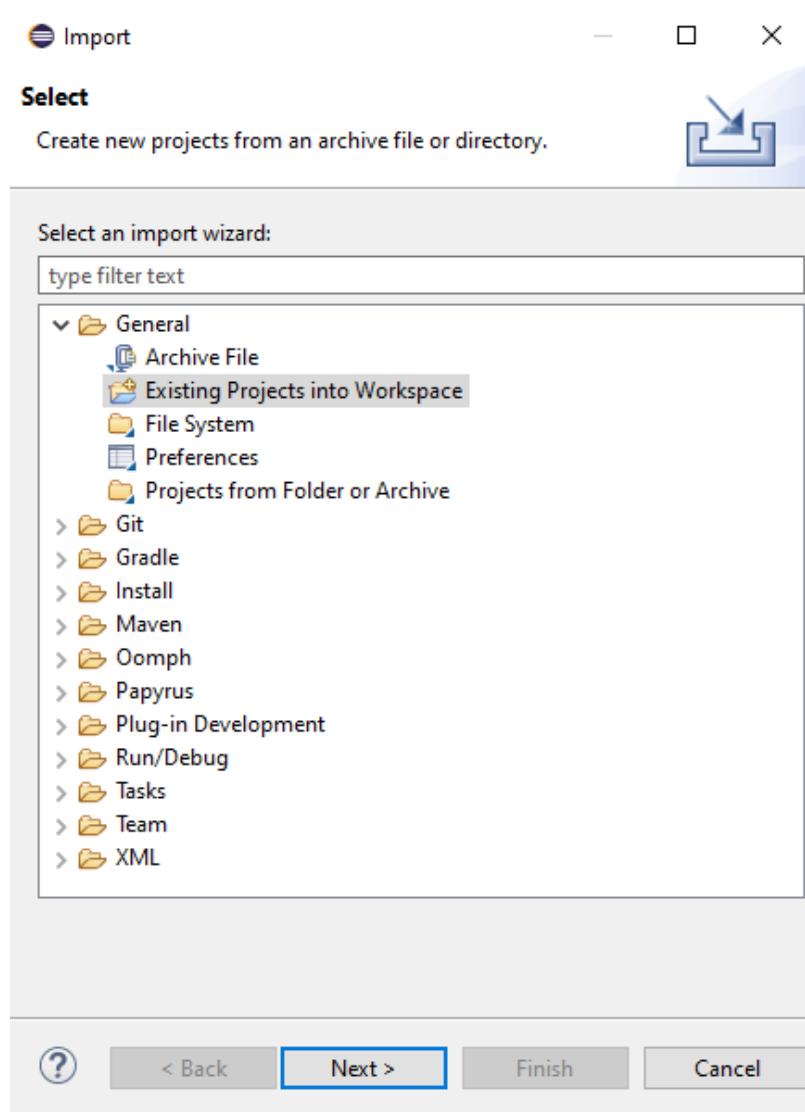
En un principio vamos a comenzar utilizando las dos opciones seleccionadas.

- **Archivo de almacenamiento:** Nos permite exportar el proyecto para poderlo importar en otro entorno de Eclipse.
- **Archivo JAR:** Archivo de JAR que contendrá la aplicación de Java lista para ejecutarse.



Cuando se exporta el proyecto como archivo de almacenamiento se genera un zip.

Para luego importar este zip hay que extraerlo e importarlo como un proyecto existente.



EJERCICIO 10: Crea un proyecto de Java en Eclipse (puedes utilizar alguno de los desarrollados para la asignatura de programación). Expórtalo como zip, elimina el proyecto del workspace de Eclipse y luego impórtalo.

5.7. Debug

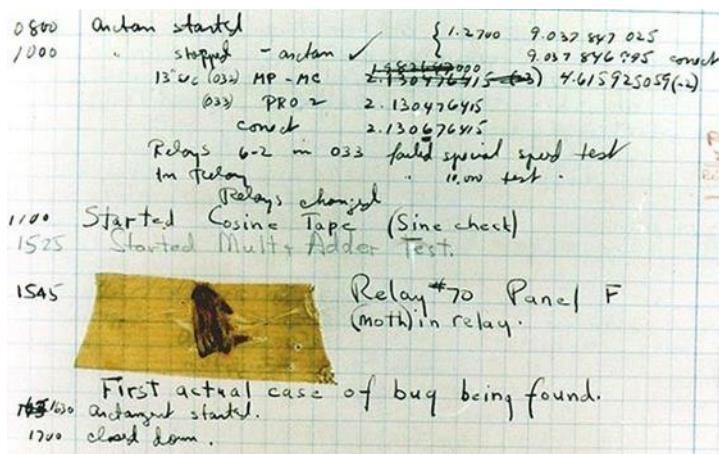
En informática el término bug se refiere a un error o defecto en el software que hace que un programa no funcione correctamente.

El origen de esta expresión en informática se remonta a 1947, cuando los operadores del ordenador Mark III comenzaron a experimentar un comportamiento extraño en sus

máquinas que hacía que no funcionaran correctamente, estos fallos provenían de un relé que fallaba.

Cuando procedieron a cambiar el relé descubrieron que había una polilla atascada en su interior. De ahí viene el término que utilizamos hoy en día para referirnos a los errores de software.

La aparición de un bug no implica que un programa se cuelgue o se cierre repentinamente, simplemente significa que el resultado no es el esperado. Así, surge cuando se dan unas condiciones específicas.



También llamado como debugging – o en español depuración -, es el nombre que se le da al proceso de encontrar y eliminar los errores que pueden cometer softwares y hardwares.

Su objetivo es encontrar errores que pueden impedir que el código funcione de forma adecuada.

Cada entorno de desarrollo tiene su propia herramienta para debuggar (o debuggear) un código. A través de ellas, se puede determinar los puntos de parada, conocidos como **breakpoints**, para verificar el estado actual de la aplicación, entre otras utilidades.

Hay que tener en cuenta que el debug solo sirve para errores que suceden en tiempo de ejecución, los errores que impiden la compilación del programa no se solucionan aplicando esta técnica.

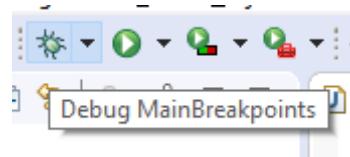
Un **breakpoint**, o punto de interrupción, es una marca que se sitúa en una línea de nuestro código fuente, de tal manera que cuando la ejecución llegue a ese punto el proceso de nuestra aplicación se detendrá y podremos analizar su estado, como una instantánea, en ese momento, justo antes de ejecutar esa línea de código.

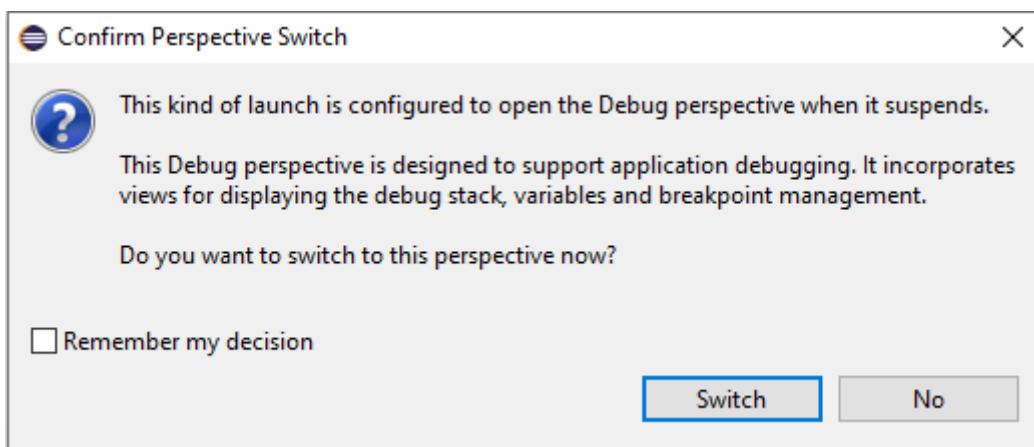
```
1 package main;
2
3 public class MainBreakpoints {
4
5     public static void main(String[] args) {
6         int x = 1;
7         int y = 10;
8
9         for(int i = 0; i < 10; i++) {
10            x = x + i;
11            y = y - i;
12        }
13    }
14 }
15 }
```

Si queremos poner un breakpoint en la línea 10 tendremos que hacer doble click sobre esa línea en el entorno.

```
1 package main;
2
3 public class MainBreakpoints {
4
5     public static void main(String[] args) {
6         int x = 1;
7         int y = 10;
8
9         for(int i = 0; i < 10; i++) {
10            x = x + i;
11            y = y - i;
12        }
13    }
14 }
15 }
```

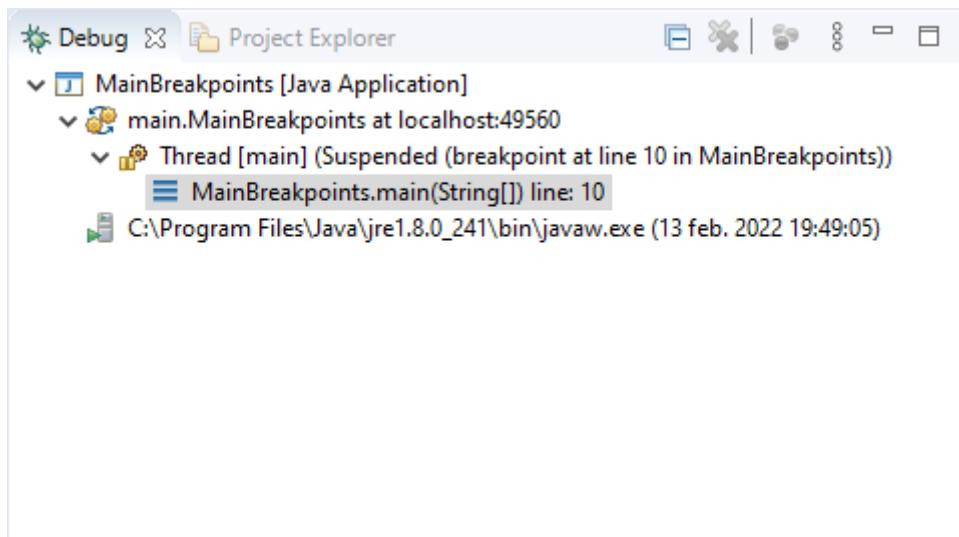
Una vez tenemos situados los breakpoints que deseamos podemos comenzar el debug.



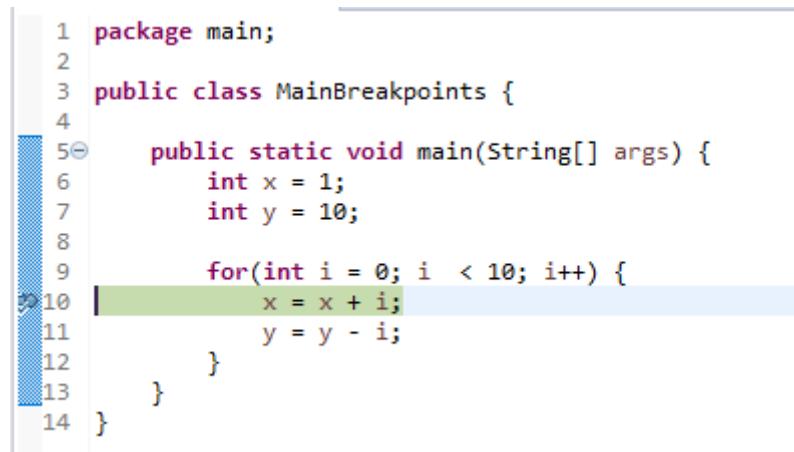


En cuanto lancéis el debug se iniciará la perspectiva correspondiente.

La vista debug permite seleccionar el proceso sobre el que se está debugeando.

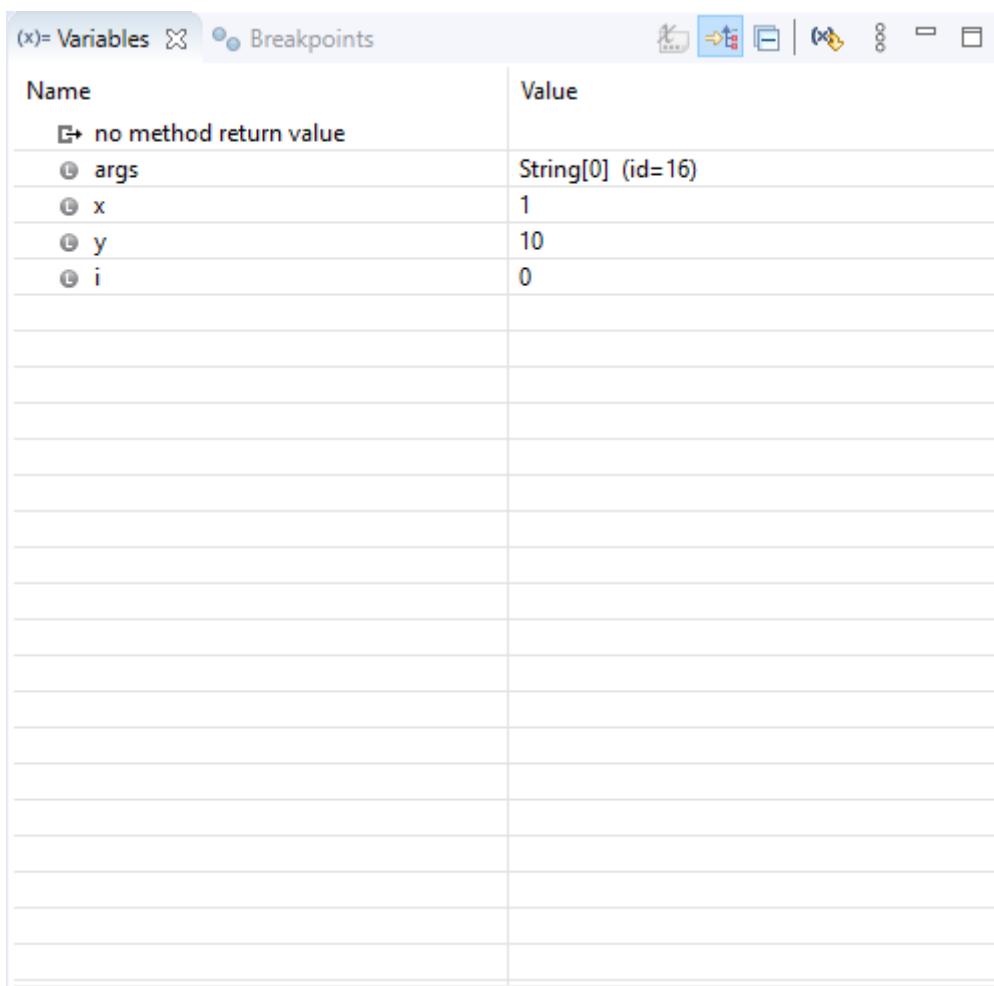


Sobre el editor de texto se muestra el breakpoint en el que se encuentra detenido el programa.



```
1 package main;
2
3 public class MainBreakpoints {
4
5     public static void main(String[] args) {
6         int x = 1;
7         int y = 10;
8
9         for(int i = 0; i < 10; i++) {
10            x = x + i;
11            y = y - i;
12        }
13    }
14 }
```

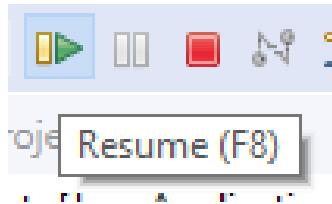
En variables se muestra el valor de las variables en el momento en el que se detiene el breakpoint (justo antes de ejecutarse la línea a la que se detiene).



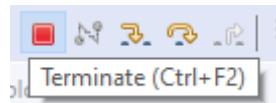
Name	Value
↳ no method return value	
↳ args	String[0] (id=16)
↳ x	1
↳ y	10
↳ i	0

Los botones más utilizados son:

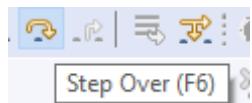
- **Reanudar:** Reanuda la ejecución del programa hasta el siguiente breakpoint.



- **Terminar:** Termina el debug.



- **Dar un paso:** Continua a la siguiente línea de código en modo debug.



- **Entrar en:** Funciona igual que dar un paso con la diferencia de que si la instrucción es una llamada a un método, entra en este y continúa el debug en la primera línea del método.



EJERCICIO 11: Importa el proyecto 1 de debug que se adjunta en clase. Pon breakpoints en las líneas 10 y 11 y, antes de ejecutarlo, responde a las siguientes preguntas.

- ¿Cuántas veces se detiene el programa en cada breakpoint?
- ¿Cuál será el valor de las variables cuando el programa se detenga la primera vez en cada uno de los breakpoints?
- ¿Cuál será el valor de las variables la cuarta vez que se detenga el programa en cada uno de los breakpoints?
- ¿Cuál será el valor de las variables x e y cuando i sea 5 en cada uno de los breakpoints?

Ejecútalo y comprueba el resultado.

EJERCICIO 12: Crea tres procesos simultáneos de debug del proyecto 1 y déjalos en la iteración 2, 3 y 4 respectivamente.

EJERCICIO 13: Importa el proyecto 2 de debug que se adjunta en clase. Pon breakpoints en las líneas 12 ,13, 15 y 16 y, antes de ejecutarlo, responde a las siguientes preguntas.

- ¿Cuántas veces se detiene el programa en cada breakpoint?
- ¿Cuál será el valor de las variables cuando el programa se detenga la primera vez en cada uno de los breakpoints?
- ¿Cuál será el valor de las variables la cuarta vez que se detenga el programa en cada uno de los breakpoints?

- ¿Cuál será el valor de las variables x, y, z y w cuando i sea 5 en cada uno de los breakpoints?

Ejecútalo y comprueba el resultado.

```
5@   public static void main(String[] args) {  
6       int x = 1;  
7       int y = 10;  
8       int z = 1;  
9       int w = 1;  
10        
11      for(int i = 0; i < 10; i++) {  
12          x = x + i;  
13          y = y - i;  
14          for(int j = 0; j < 10; j++) {  
15              z = z + j;  
16              w = w + i;  
17          }  
18      }  
19  }  
20 }
```

EJERCICIO 14: Importa el proyecto 3 de debug que se adjunta en clase. Pon breakpoints en las líneas 13, 14, 20, 21, 23, 26 y 28 y, antes de ejecutarlo, responde a las siguientes preguntas.

- ¿Cuánto valdrán las variables cuando se detenga el programa en el breakpoint de la línea 13?
- ¿Cuánto valdrán las variables cuando se detenga el programa en el breakpoint de la línea 14?
- Con el input 5, 7, 2 ¿En qué breakpoints se detendrá el programa?
- Con el input 8, 7, 2 ¿En qué breakpoints se detendrá el programa?
- Con el input 3, 7, 15 ¿En qué breakpoints se detendrá el programa?
- Con el input 1, 1, 1 ¿En qué breakpoints se detendrá el programa?

Ejecútalo y comprueba el resultado.

```

3 import java.util.Scanner;
4
5 public class MainBreakpoints {
6
7    public static void main(String[] args) {
8        int n1, n2, n3;
9
10       Scanner sc = new Scanner(System.in);
11
12       System.out.println("Introduzca el primer número: ");
13       n1 = sc.nextInt();
14       System.out.println("Introduzca el segundo número: ");
15       n2 = sc.nextInt();
16       System.out.println("Introduzca el tercer número: ");
17       n3 = sc.nextInt();
18
19       if (n1 > n2) {
20           if (n1 > n3) {
21               System.out.println("El mayor es : " + n1);
22           } else {
23               System.out.println("El mayor es : " + n3);
24           }
25       } else if (n2 > n3) {
26           System.out.println("El mayor es: " + n2);
27       } else {
28           System.out.println("El mayor es: " + n3);
29       }
30   }
31 }
```

5.7.1. Evaluate expression

El debugger también nos permitirá ver cuál será el valor de una expresión en el momento concreto en el que se encuentra detenido el programa.

Expressions	
Name	Value
X+Y =? "y*x"	18
X+Y =? "1+2"	3
Add new expression	

La vista expressions es la que se encarga de esto y se puede añadir cualquier expresión de Java que se desee (puede incluir métodos, clases, etc).

EJERCICIO 15: En el ejercicio 1 de debug en el segundo breakpoint (línea 10) ¿qué valor tendrían las expresiones “y * x” e “y / x” ?

- Cuando i es 0
- Cuando i es 3
- Cuando i es 7

EJERCICIO 16: En el ejercicio 2 de debug ¿qué valor tendrían las expresiones “y * x + z” e “y

/ x” ?

- En el breakpoint de la línea 16 Cuando i es 0 y j es 0.
- En el breakpoint de la línea 16 Cuando i es 4 y j es 2.
- En el breakpoint de la línea 16 Cuando i es 6 y j es 6.

5.8. Añadir bibliotecas

Hay una gran cantidad de bibliotecas que se encuentran accesibles desde el proyecto creado en el entorno de desarrollo correspondiente y que solo tenemos que importarlas poder llamar a sus clases y métodos.

No obstante, existen muchas más bibliotecas que podemos utilizar para dotar de mayor funcionalidad a nuestro programa y que es necesario añadir al espacio de trabajo o al proyecto para que se puedan importar.

Por ejemplo, en el siguiente proyecto queremos crear una función para encriptar textos.

```
1 package importe.manual;
2
3④ import java.nio.charset.StandardCharsets;
4 import java.security.MessageDigest;
5 import java.util.Arrays;
6
7 import javax.crypto.Cipher;
8 import javax.crypto.SecretKey;
9 import javax.crypto.spec.SecretKeySpec;
10
11 import org.apache.commons.codec.binary.Base64;
12
13
14 public class ImporteManual {
15
16④     public static void main(String[] args) {
17         System.out.println(encryptar("prueba"));
18     }
19
20④     public static String encryptar(String texto) {
21
22         String secretKey = "llave-encryptar";
23         String base64EncryptedString = "";
24
25         try {
26             MessageDigest md = MessageDigest.getInstance("MD5");
27             byte[] digestOfPassword = md.digest(secretKey.getBytes(StandardCharsets.UTF_8));
28             byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
29
30             SecretKey key = new SecretKeySpec(keyBytes, "DESEde");
31             Cipher cipher = Cipher.getInstance("DESEde");
32             cipher.init(Cipher.ENCRYPT_MODE, key);
33
34             byte[] plainTextBytes = texto.getBytes(StandardCharsets.UTF_8);
35             byte[] buf = cipher.doFinal(plainTextBytes);
36             byte[] base64Bytes = Base64.encodeBase64(buf);
37             base64EncryptedString = new String(base64Bytes);
38
39         } catch (Exception ex) {
40
41             return base64EncryptedString;
42         }
43     }
44 }
```

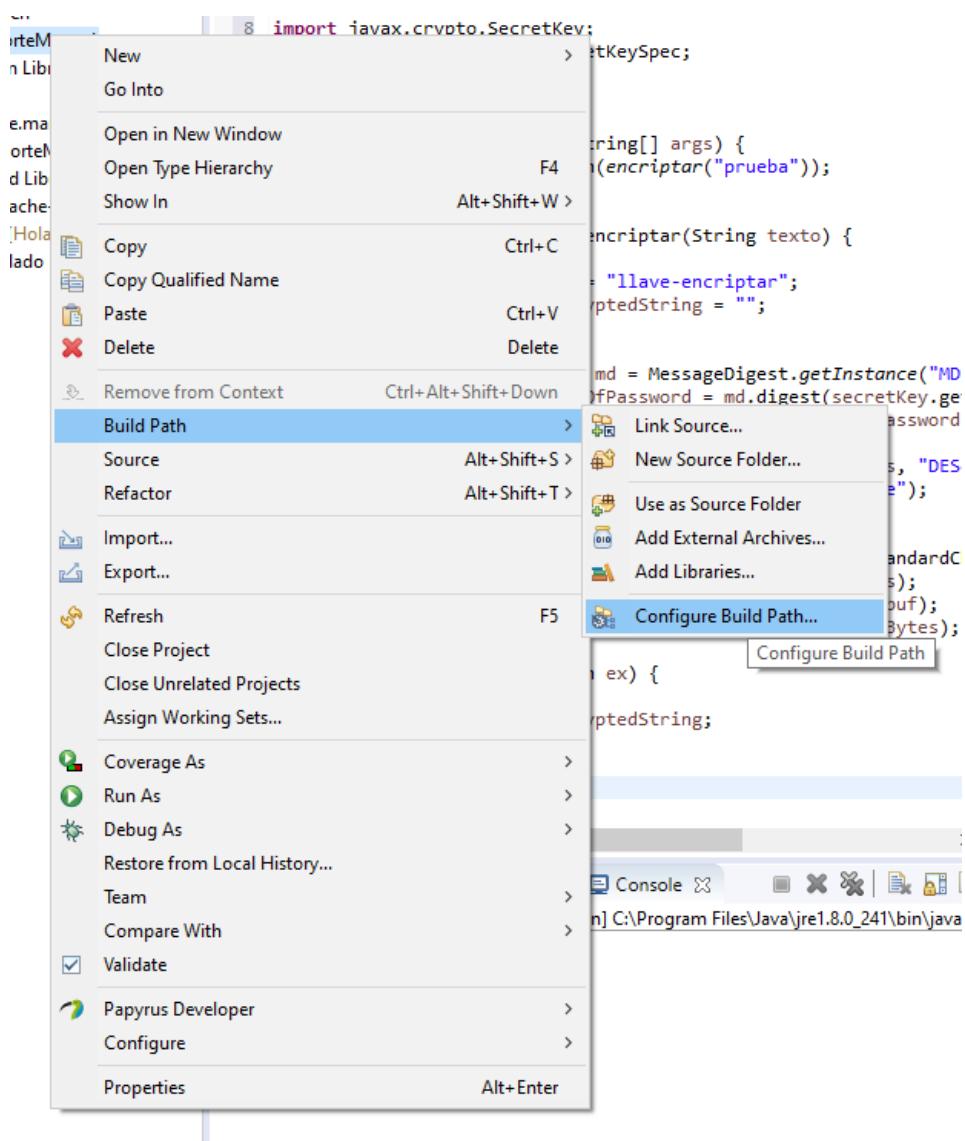
Para usar la función creada necesitamos usar la clase Base64 de la biblioteca org.apache.commons.codec que no podemos importar directamente.

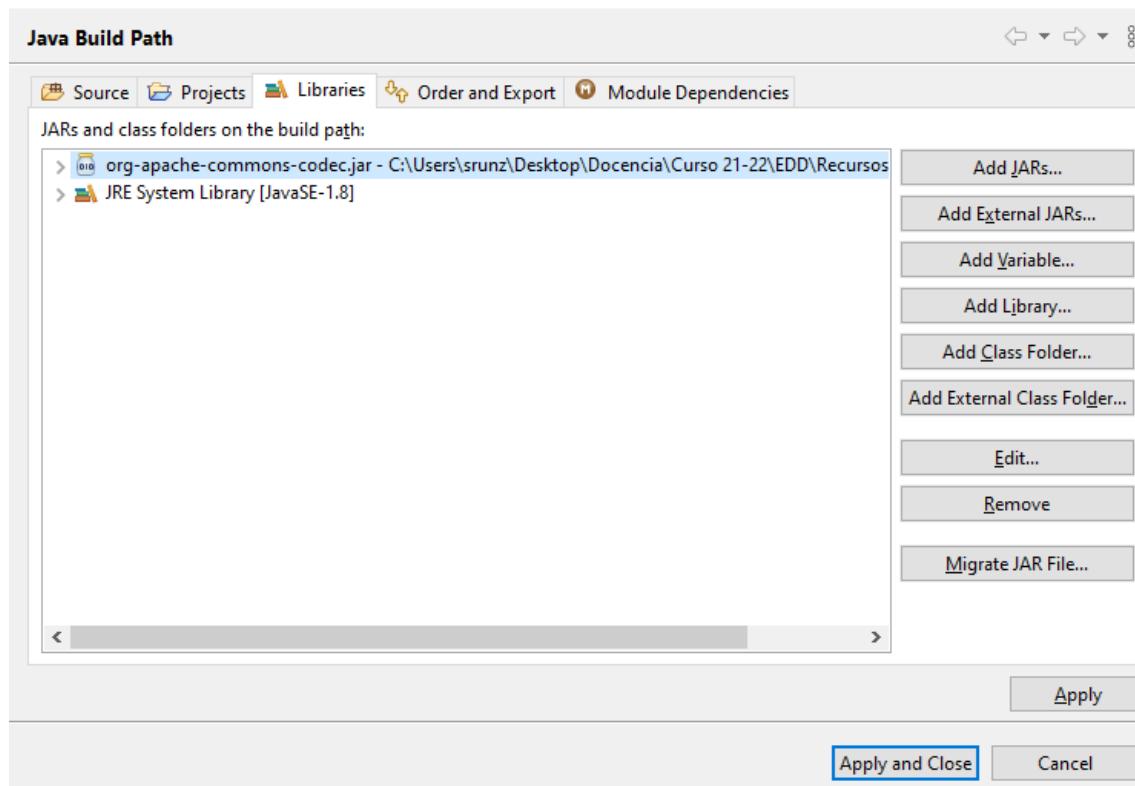
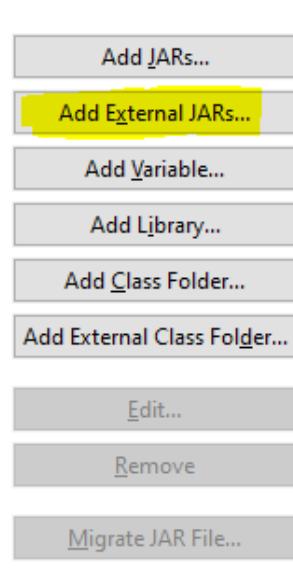


```
C:\ll import org.apache.commons.codec.binary.Base64;
```

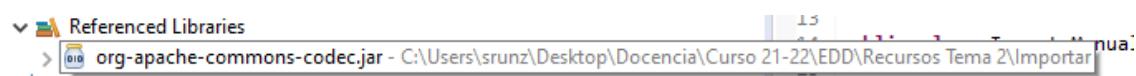
Como veis el entorno no la reconoce.

Se puede referenciar la biblioteca desde el proyecto.



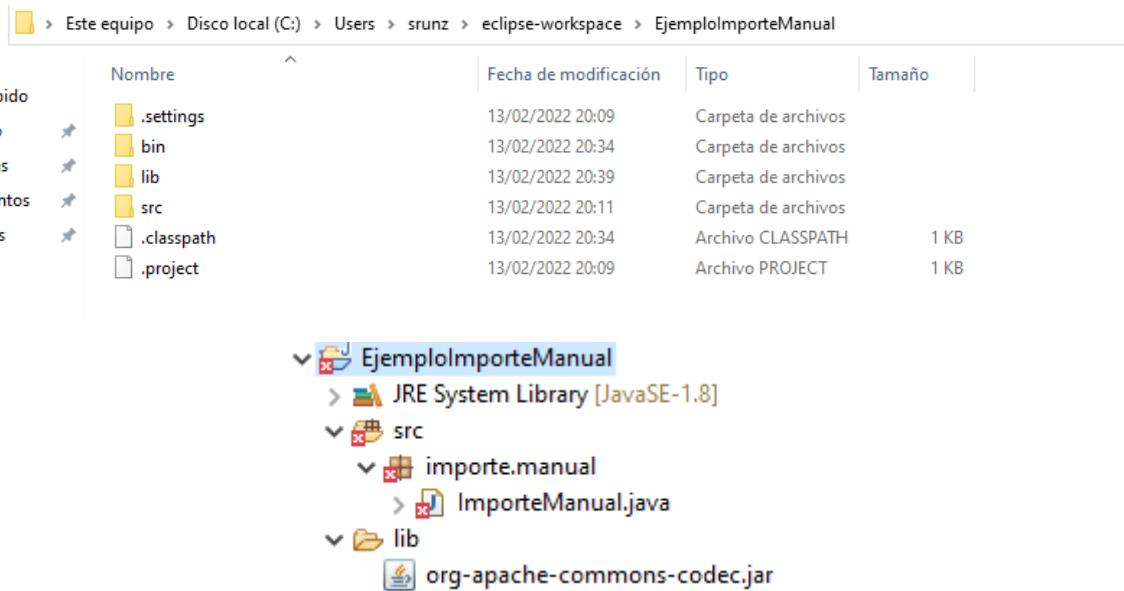


Pero esto simplemente crea una referencia a la librería, por tanto, cuando exportemos el proyecto vamos a tener problemas.

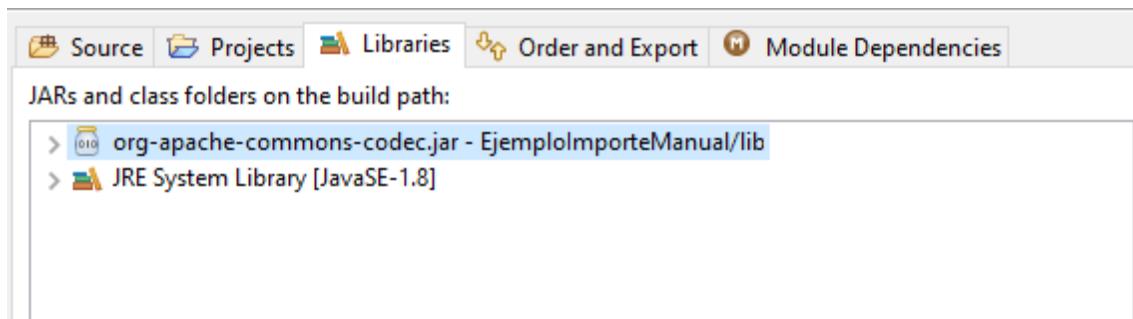


EJERCICIO 17: Descarga el proyecto y referencia el jar. ¿Qué pasa si cambias de dirección el jar después de referenciarlo? ¿Qué pasaría si exportaras el proyecto a otro ordenador?

Para evitar esto, se puede crear una carpeta lib en el proyecto de manera manual y añadir ahí todas las bibliotecas que queramos añadir.



En este caso seleccionamos la opción de añadir jars.



Y añadimos el jar correspondiente.

EJERCICIO 18: En vez de referenciarlo, importa el jar necesario. ¿Qué pasa si cambias de dirección el jar después de referenciarlo? ¿Qué pasaría si exportaras el proyecto a otro ordenador?

The screenshot shows a Java project structure on the left and a code editor on the right. The project structure includes packages like Clasebreakpoints, com.vogella.emf.webpage.model, com.vogella.emf.webpage.model.editor, ejemplo.maven, EjemplolImporteManual, JRE System Library [JavaSE-1.8], src, Referenced Libraries, lib, and org.apache.commons-codec.jar. The HolaMundo main class is selected. The code editor contains Java code for encrypting strings using DESede cipher:

```

1 package importe.manual;
2
3 import java.nio.charset.StandardCharsets;
4 import java.security.MessageDigest;
5 import java.util.Arrays;
6
7 import javax.crypto.Cipher;
8 import javax.crypto.SecretKey;
9 import javax.crypto.spec.SecretKeySpec;
10
11 import org.apache.commons.codec.binary.Base64;
12
13
14 public class ImporteManual {
15
16     public static void main(String[] args) {
17         System.out.println(encryptar("prueba"));
18     }
19
20     public static String encryptar(String texto) {
21
22         String secretKey = "llave-encryptar";
23         String base64EncryptedString = "";
24
25         try {
26             MessageDigest md = MessageDigest.getInstance("MD5");
27             byte[] digestOfPassword = md.digest(secretKey.getBytes(StandardCharsets.UTF_8));
28             byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
29
30             SecretKey key = new SecretKeySpec(keyBytes, "DESede");
31             Cipher cipher = Cipher.getInstance("DESEde");
32             cipher.init(Cipher.ENCRYPT_MODE, key);
33
34             byte[] plainTextBytes = texto.getBytes(StandardCharsets.UTF_8);
35             byte[] buf = cipher.doFinal(plainTextBytes);
36             byte[] base64Bytes = Base64.encodeBase64(buf);
37             base64EncryptedString = new String(base64Bytes);
38
39         } catch (Exception ex) {
40
41     }
42
43     return base64EncryptedString;
44 }

```

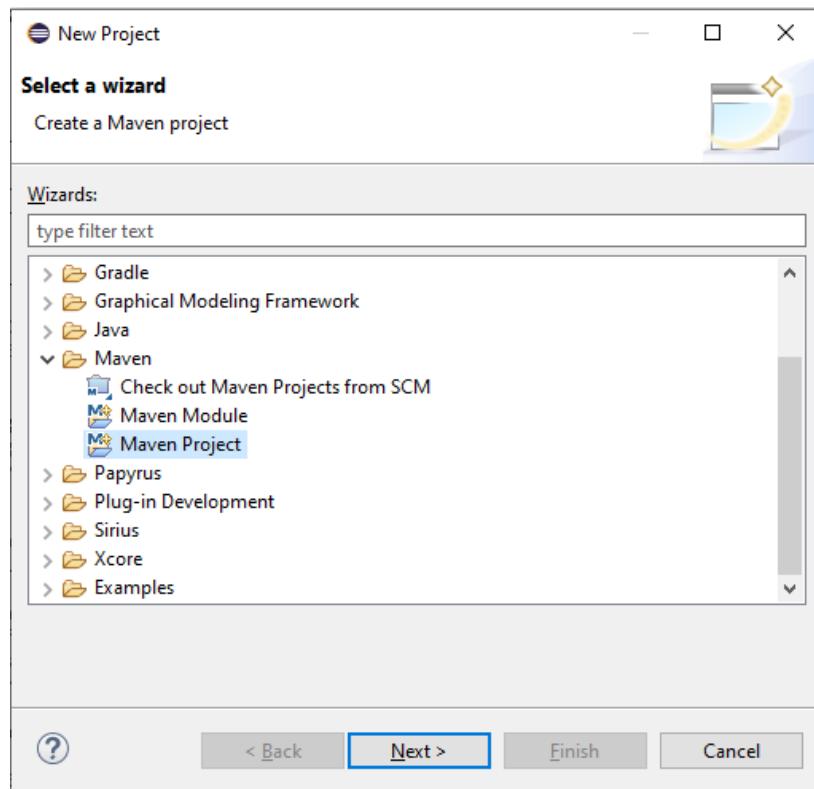
The IDE interface includes tabs for Problems, Javadoc, Declaration, Search, and Console. The console shows the command used to run the application.

Cuando usamos muchas bibliotecas de terceros que es necesario añadir al espacio de trabajo el mantenimiento y gestión de estas puede llegar a ser harto complicado.

Para hacer algo más ligero este trabajo se utilizan los **gestores de dependencias** que tienen como objetivo facilitar la instalación, el mantenimiento y la exportación de las bibliotecas añadidas en el proyecto.

En java son muy utilizados Maven y Gradle, que son mucho más que gestores de dependencias, pero incluyen esta funcionalidad.

Para probarlo vamos a crear un nuevo proyecto de Maven.



Para la creación de proyectos, Maven ofrece **arquetipos** que son patrones o modelos a partir de los cuales se pueden desarrollar tareas de un mismo tipo.

prototipo	descripción
maven-archetype-archetype	El prototipo utilizado para generar la plantilla prototipo.
maven-archetype-j2ee-simple	Se usa para generar plantillas de aplicaciones J2EE simplificadas
maven-archetype-plugin	Se utiliza para generar plantillas de complementos Maven
maven-archetype-plugin-site	Se utiliza para generar la plantilla del sitio del complemento Maven
maven-archetype-portlet	Se utiliza para generar la plantilla de portlet JSR-268
maven-archetype-quickstart	Se usa para generar plantillas de proyecto Maven
maven-archetype-simple	Se usa para generar plantillas de proyecto Maven
maven-archetype-site	Un prototipo utilizado para generar una plantilla de sitio Maven. El sitio muestra algunos tipos de documentos compatibles, como APT, Markdown, XDoc y FML, y muestra cómo operar en su sitio web.
maven-archetype-site-simple	Se usa para generar plantillas de proyecto Maven
maven-archetype-site-skin	Se utiliza para generar la plantilla de Maven Site Skin
maven-archetype-webapp	Se utiliza para generar plantillas de proyecto Maven Webapp

Vamos a usar el arquetipo más genérico para probar la tecnología.

Group Id	Artifact Id	Version
com.github.ywchang com.haoxuer.maven.archetype org.apache.maven.archetypes	maven-archetype-quickstart maven-archetype-quickstart maven-archetype-quickstart	1.1 1.01 1.4

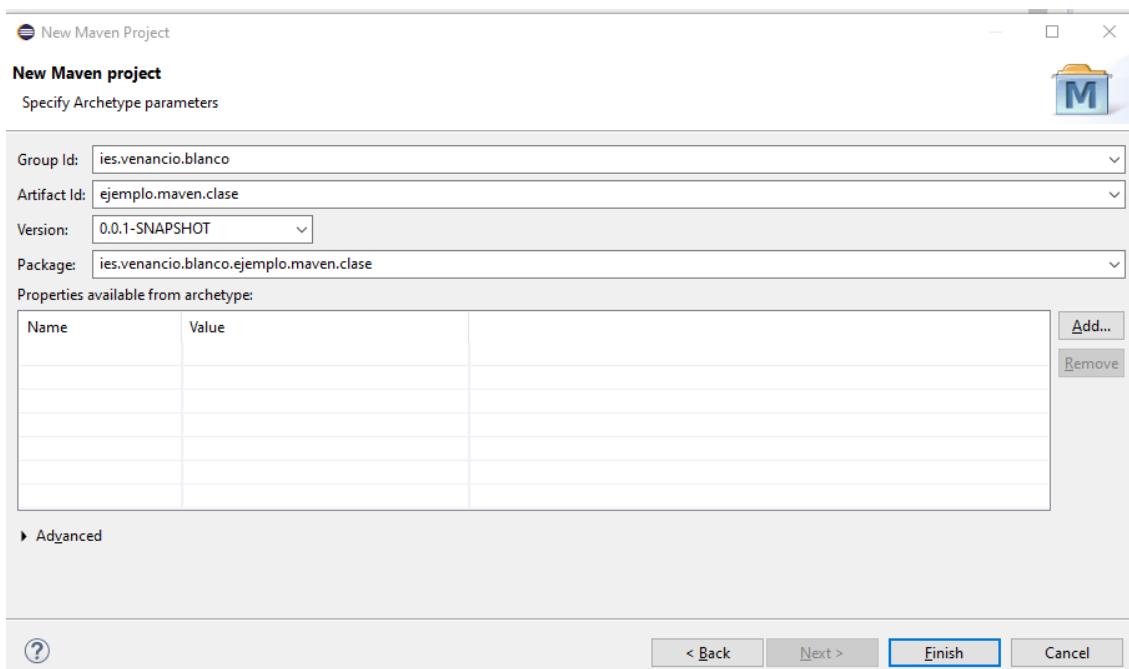
An archetype which contains a sample Maven project.
<https://repo1.maven.org/maven2>

Show the last version of Archetype only Include snapshot archetypes [Add Archetype...](#)

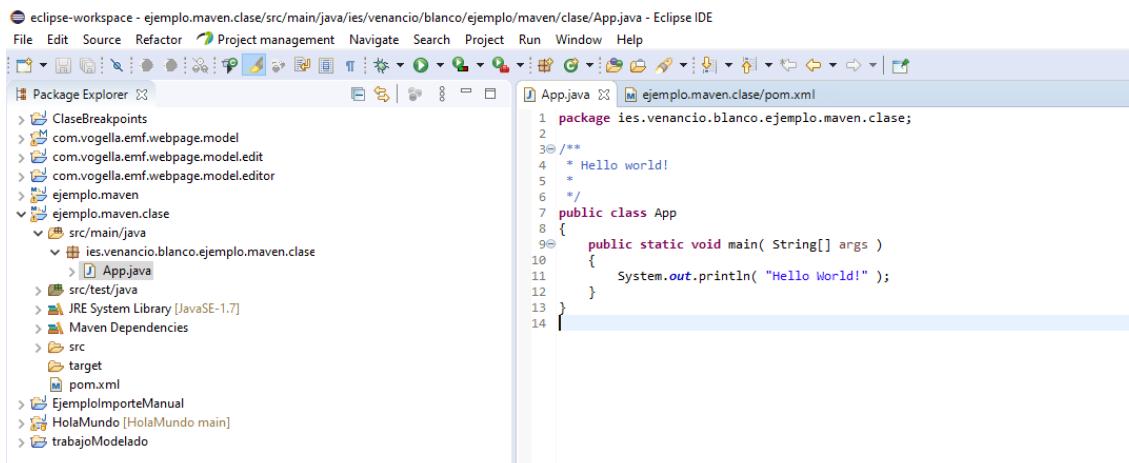
[Advanced](#)

En la siguiente pantalla se nos pedirán los siguientes datos:

- **Id del grupo:** Nombre único de la empresa o grupo que creó el proyecto. (Sigue la nomenclatura de paquetes Java)
- **Id del artefacto:** Nombre único del proyecto. (Sigue la nomenclatura de paquetes Java)
- **Versión:** Versión del proyecto.



Una vez se ha generado el proyecto es interesante observar su estructura.



El fichero pom.xml (Project Object Model) es la unidad principal del proyecto Maven.

Es una archivo con extensión XML que tiene información sobre las dependencias, configuraciones y otra información importante sobre el proyecto.

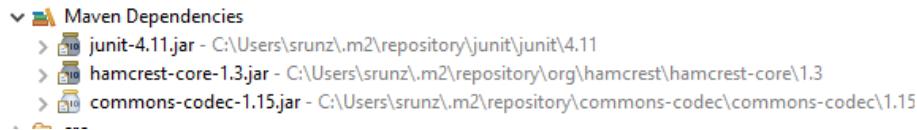
En este caso vamos a ahondar tan solo en las dependencias y como queremos añadir la biblioteca apache.commons.codec simplemente tendremos que buscar su dependencia en el repositorio de Maven.

<https://mvnrepository.com/artifact/commons-codec/commons-codec>

Se añade manualmente la dependencia y guardamos los cambios.

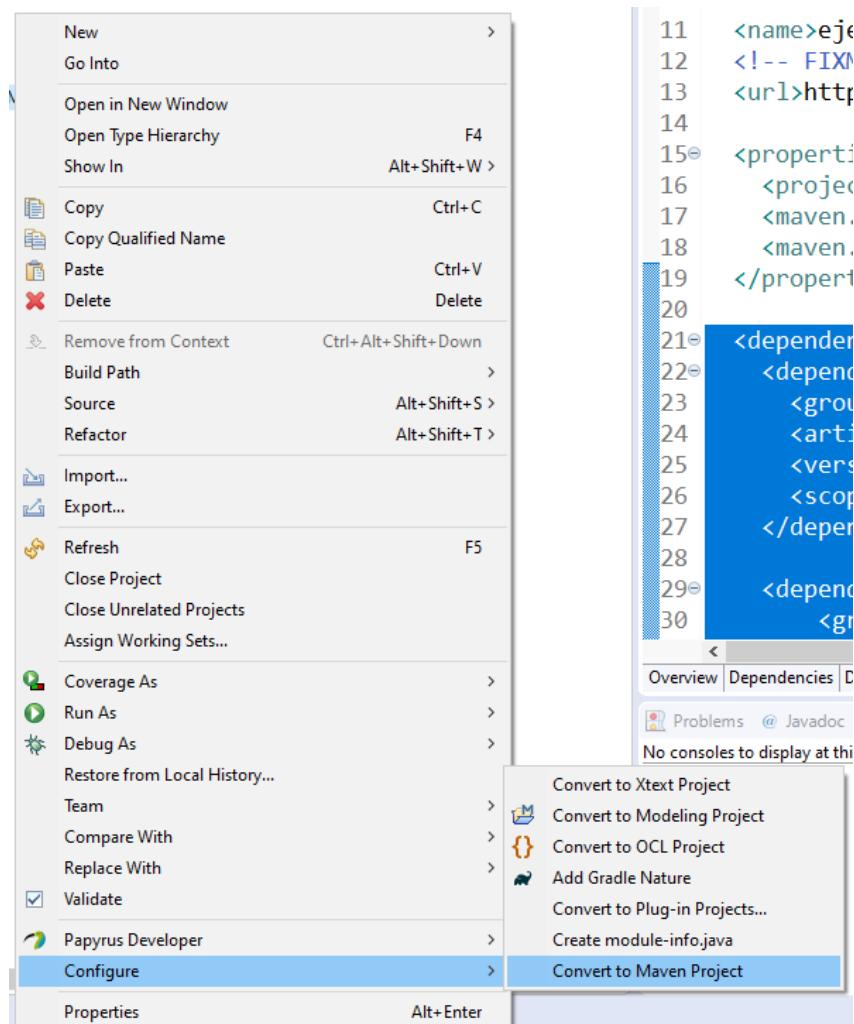
```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.15</version>
  </dependency>
</dependencies>
```

Si la dependencia se ha añadido correctamente aparecerá la biblioteca dentro de las dependencias de Maven.



Y ya se podrá importar la biblioteca.

Se puede convertir cualquier proyecto de Java existente a Maven.



EJERCICIO 19: Elimina el jar, convierte el proyecto a Maven y añade la biblioteca usando el gestor de dependencias. ¿Qué ventajas tiene?

5.9. Errores y warnings de compilador

Los entornos de desarrollo son capaces de detectar errores de sintaxis que provocarían fallos en compilación.

Cuando eclipse detecta un error lo marca con un símbolo rojo a la izquierda de la línea.

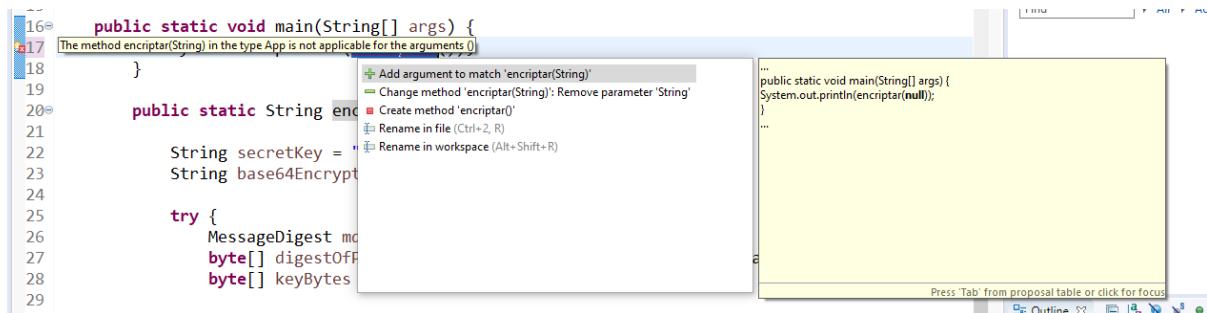
El entorno nos indica por qué se produce este error.

```

16o   public static void main(String[] args) {
17x     Syntax error, insert ";" to complete BlockStatements
18       }
  
```

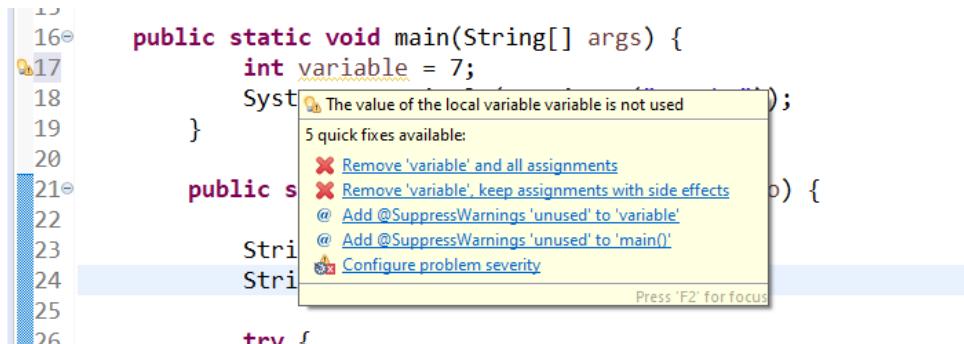
Si eclipse considera que puede solucionar el error lo marca con una bombilla.

```
public static void main(String[] args) {
    System.out.println(encryptar());
}
```

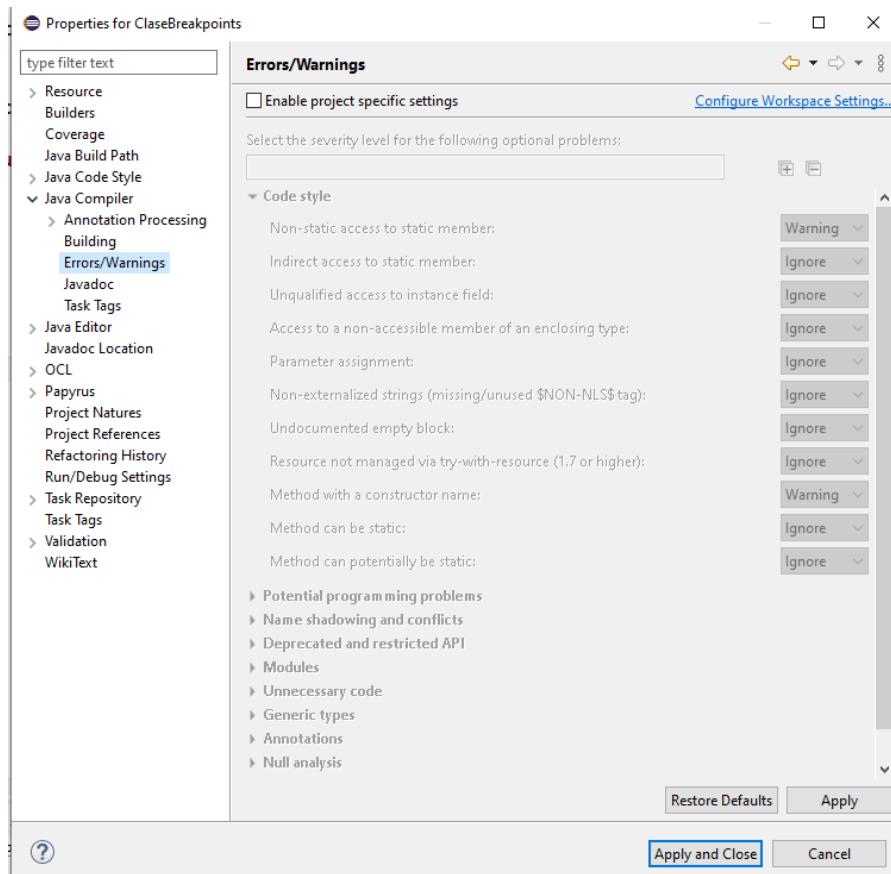


Los entornos de desarrollo también nos muestran **avisos**, que son mensajes de advertencia que se muestran cuando, aunque el código es válido (va a poder compilar y ejecutarse), se ha detectado alguna posible mala práctica de programación que puede provocar que el código no haga lo que el programador espera o que sea menos eficiente.

Es conveniente analizarlos y arreglarlos.



Se puede configurar el entorno para que detecte ciertas prácticas de programación como warnings o como errores (según deseemos).



EJERCICIO 20: Consigue 4 errores y 3 warnigs diferentes.

5.10. Generador de código

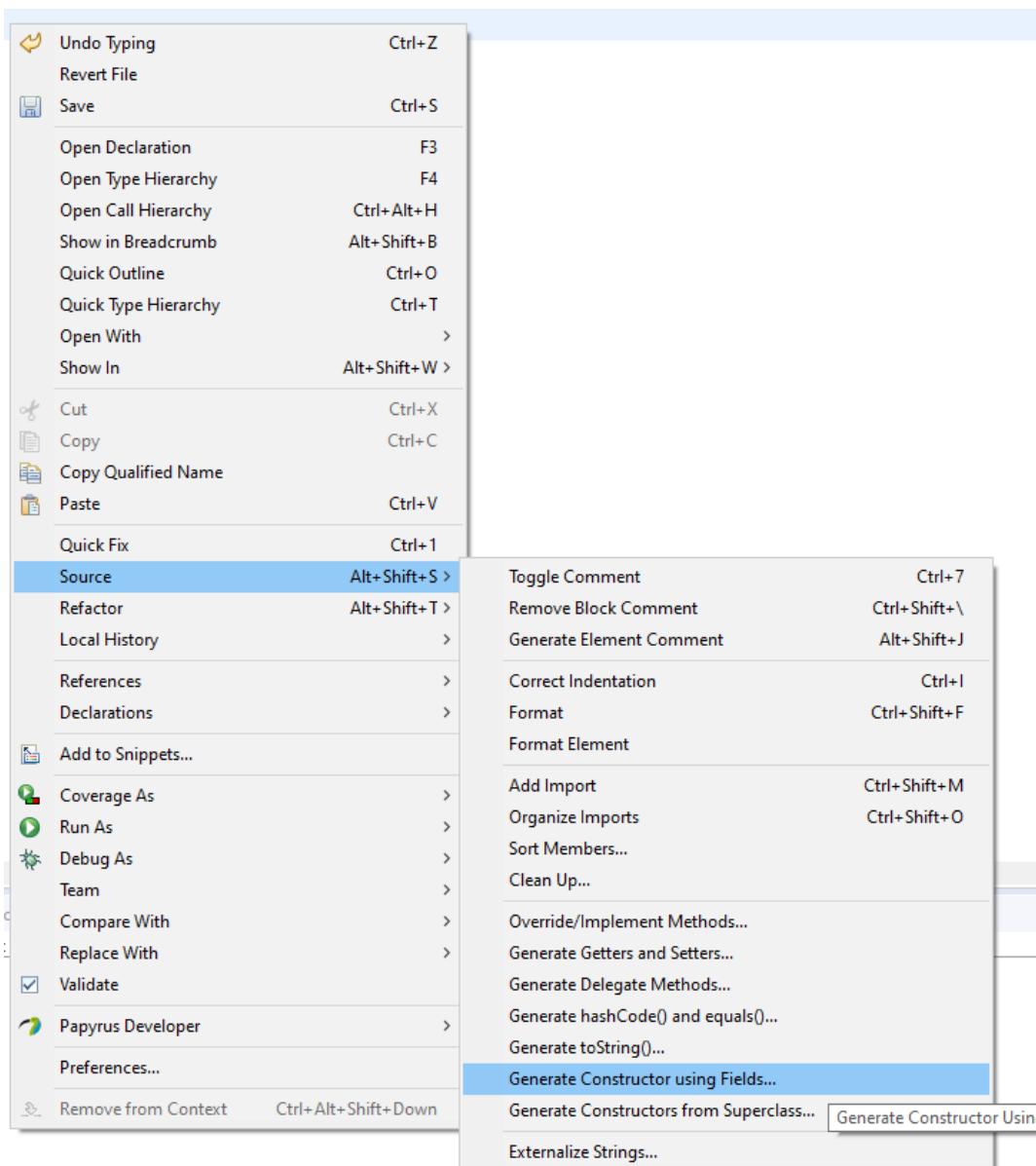
Los entornos dan también la opción de autogenerar código en casos concretos y con poco margen de variación.

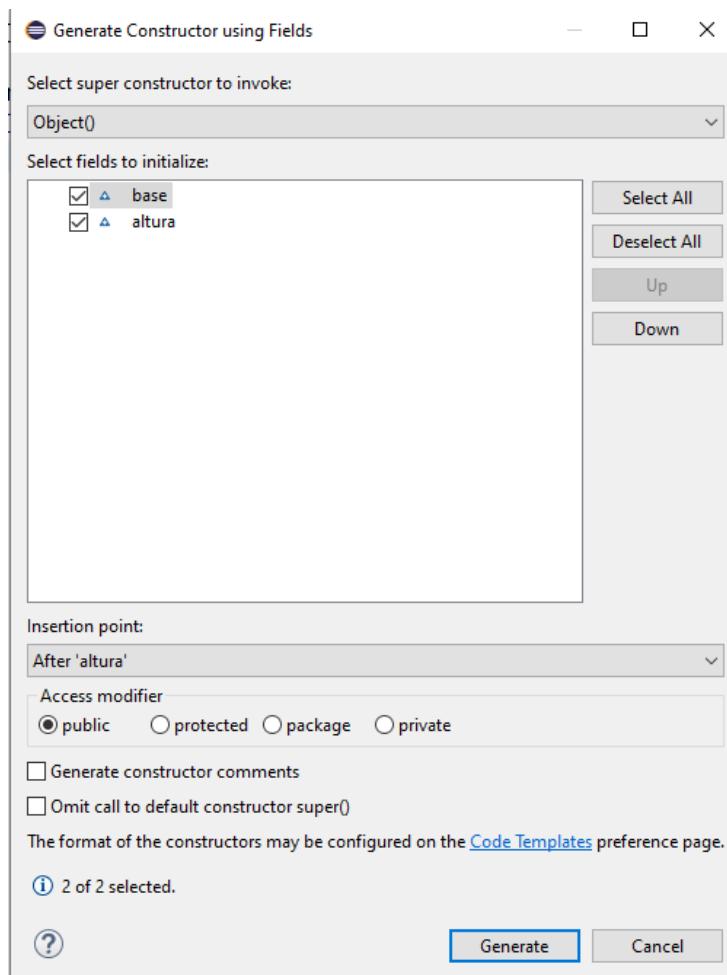
Por ejemplo, en una clase triángulo que creamos y a la cual le añadimos estos dos atributos podemos autogenerarle algunos métodos.

```

1 package proyecto.javadoc.paquete;
2
3 public class Triangulo {
4     float base, altura;
5
6 }
```

Por ejemplo, se le puede generar fácilmente el constructor.





```
public Triangulo(float base, float altura) {  
    super();  
    this.base = base;  
    this.altura = altura;  
}
```

También se pueden generar, por ejemplo, los getters y setters.

```

3 public class Triangulo {
4     private float base, altura;
5
6     public Triangulo(float base, float altura) {
7         super();
8         this.base = base;
9         this.altura = altura;
10    }
11
12    public float getBase() {
13        return base;
14    }
15
16    public void setBase(float base) {
17        this.base = base;
18    }
19
20    public float getAltura() {
21        return altura;
22    }
23
24    public void setAltura(float altura) {
25        this.altura = altura;
26    }
27 }
```

EJERCICIO 21: Crea una clase usuario que tenga los atributos nombre, apellidos, clave y correo. Autogenera un constructor vacío, uno con todos los parámetros, los getters y los setters.

5.11. Generador de documentación

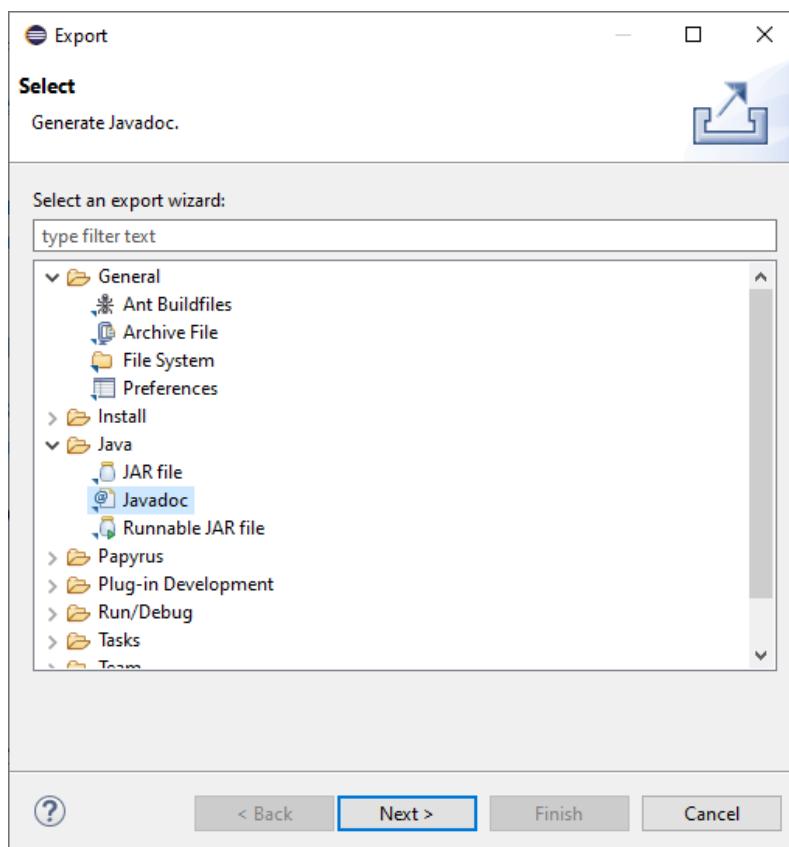
En el tema de documentación ya se explicará más a fondo cómo comentar correctamente un proyecto.

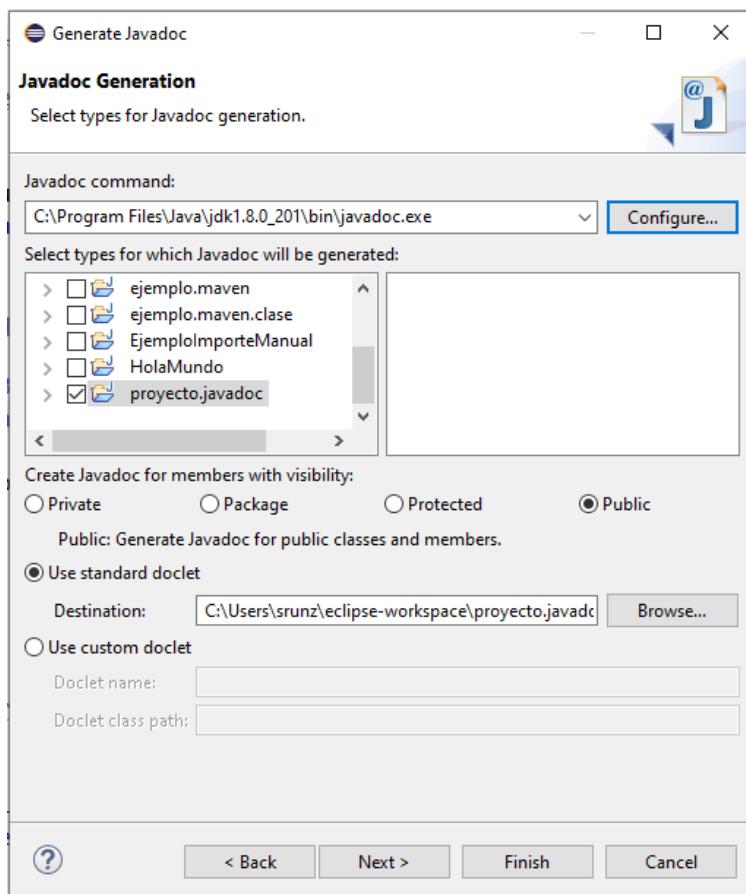
En este punto solo vamos a ver cómo extraer el Javadoc en Eclipse.

```

3 /**
4 * <h2>Principal</h2>
5 * Clase creada para probar los triángulos.
6 *
7 * @author Miguel de Unamuno
8 *
9 */
10 public class Principal {
11     public static void main(String[] args) {
12         System.out.println("Voy a crear un nuevo triángulo");
13         Triangulo triangulo = new Triangulo(10, 5);
14         System.out.println("El triángulo tiene " + triangulo.calcularArea() + " cms de área");
15     }
16 }
```

```
3o /**
4 * <h2>Triángulo</h2>
5 * Clase que permitirá realizar cálculos con triángulos.
6 *
7 * @author Miguel de Unamuno
8 *
9 */
10 public class Triangulo {
11     private float base, altura;
12
13o /**
14 * Constructor que permite generar un triángulo a partir de su base y altura.
15 * La base y la altura ha de introducirse en centímetros.
16 * @param base Base del triángulo.
17 * @param altura altura del triángulo.
18 */
19o public Triangulo(float base, float altura) {
20     super();
21     this.base = base;
22     this.altura = altura;
23 }
24
25o /**
26 * Calcula el área del triángulo en centímetros.
27 * @return área del triángulo en centímetros.
28 */
29o public float calcularArea() {
30     return (base*altura/2);
31 }
32
33 /**
34 * Calcula el área del triángulo en centímetros.
35 * @return área del triángulo en centímetros.
36 */
37 public float calcularArea() {
38     return (base*altura/2);
39 }
40
41 /**
42 * Imprime las características del triángulo.
43 */
44 public void imprimirTriángulo() {
45     System.out.println("El triángulo tiene " + altura + " cms de altura y " + base + " cms de base");
46 }
47
48 public float getBase() {
49     return base;
50 }
51
52 public void setBase(float base) {
53     this.base = base;
54 }
55
56 public float getAltura() {
57     return altura;
58 }
59
60 public void setAltura(float altura) {
61     this.altura = altura;
62 }
```





En java command hay que poner de manera manual la ubicación del programa javadoc (normalmente en la carpeta bin del jdk).

Se exportará el Javadoc en la carpeta de destino indicada y se podrá consultar la documentación realizada y navegar por ella.

index-files	15/02/2022 10:39	Carpeta de archivos
proyecto	15/02/2022 10:39	Carpeta de archivos
allclasses-frame.html	15/02/2022 10:39	Chrome HTML Do...
allclasses-noframe.html	15/02/2022 10:39	Chrome HTML Do...
constant-values.html	15/02/2022 10:39	Chrome HTML Do...
deprecated-list.html	15/02/2022 10:39	Chrome HTML Do...
help-doc.html	15/02/2022 10:39	Chrome HTML Do...
index.html	15/02/2022 10:39	Chrome HTML Do...
overview-tree.html	15/02/2022 10:39	Chrome HTML Do...
package-list	15/02/2022 10:39	Archivo
script.js	15/02/2022 10:39	Archivo JS
stylesheet.css	15/02/2022 10:39	Archivo CSS

Package proyecto.javadoc.paquete

Class Summary	
Class	Description
Principal	Principal Clase creada para probar los triángulos.
Triangulo	Triángulo Clase que permitirá realizar cálculos con triángulos.

proyecto.javadoc.paquete

Class Triangulo

java.lang.Object
proyecto.javadoc.paquete.Triangulo

```
public class Triangulo
extends java.lang.Object
```

Triángulo

Clase que permitirá realizar cálculos con triángulos.

Author:

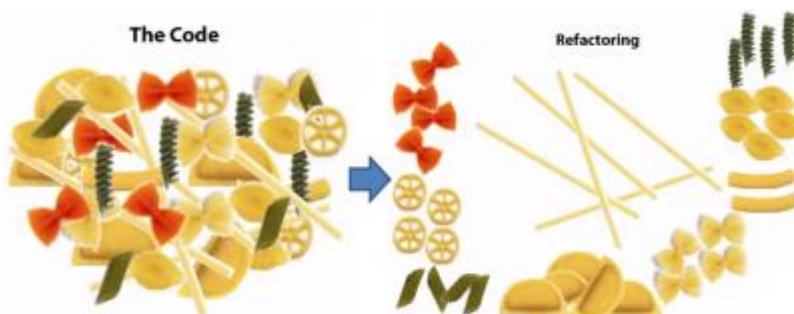
Miguel de Unamuno

EJERCICIO 22: Crea el javadoc correspondiente a un proyecto con, al menos, 3 clases (además del main) y, al menos, 4 métodos en total (podéis usar algún proyecto realizado en la asignatura de programación).

5.12. Refactorizar y buscar

Refactorizar consiste en mejorar y optimizar la estructura del programa y el código sin modificar el comportamiento observable.

Es decir, la refactorización no tiene como fin añadir una nueva funcionalidad o arreglar errores sino mejorar la comprensión del código.



La refactorización permite evitar los llamados malos olores de código (bad smells).

Los **malos olores** son errores de programación que afectan a la forma, pero no en el fondo. Es decir, un código con malos olores puede compilar y funcionar, no obstante, este código será de menor calidad y será más difícil de mantener y entender.

En esta web se recogen algunos de los malos olores

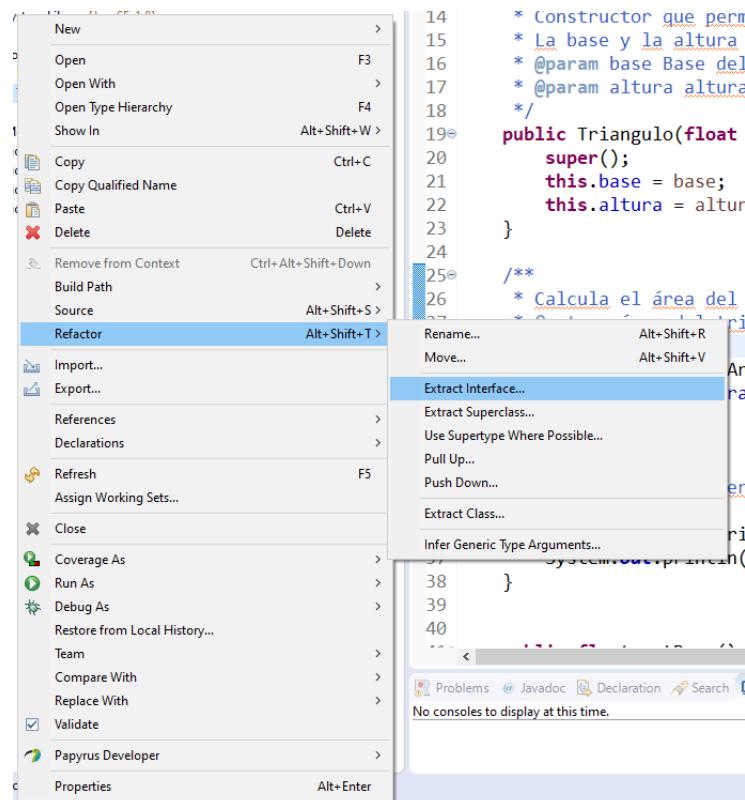
<https://refactoring.guru/es/refactoring/smells> que se pueden encontrar en el código.

Aquí se listarán algunos de los malos olores más relevantes.

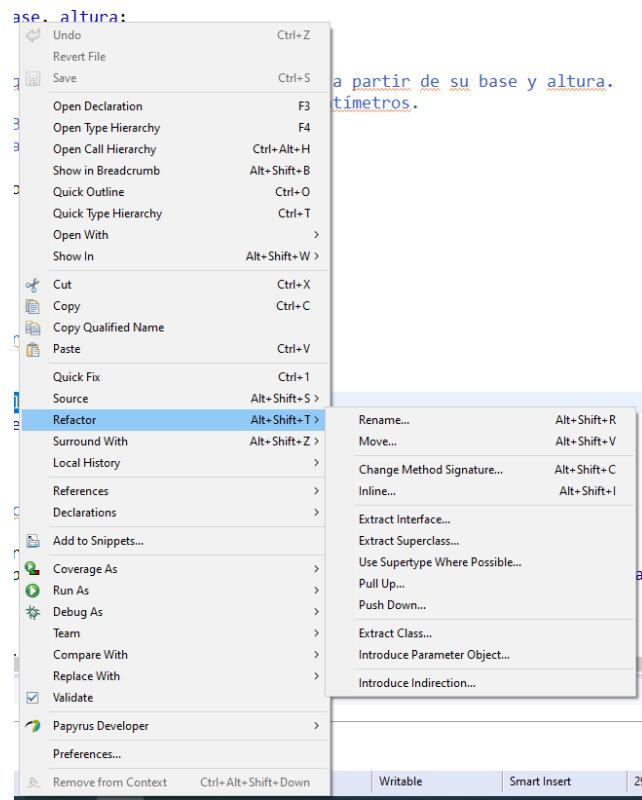
- **Hinchados:** Código, métodos y clases que tienen un tamaño muy grande y es difícil trabajar con ellos.
 - **Función larga:** Las funciones cortas suelen ser mucho más fáciles de entender y reutilizar. Las funciones muy largas (a partir de 10 líneas) normalmente pueden descomponerse en funciones más pequeñas.
<https://refactoring.guru/es/smells/long-method>
 - **Muchos parámetros:** En la programación orientada a objetos no se suelen pasar demasiados parámetros en las funciones (no más de tres o cuatro). Si pasamos muchos parámetros en una función puede simbolizar que estos se podrían encapsular en un objeto.
<https://refactoring.guru/es/smells/long-parameter-list>
 - ...
- **Abusadores de la orientación a objetos:** Malas prácticas con la programación orientada a objetos.
- **Preventores de cambio:** Si cuando se necesita cambiar algo en un lugar del código, también hay que realizar muchos cambios en otros lugares.
 - **Cambio divergente:** Surge cuando hay que realizar muchos cambios entre métodos que no tienen mucha relación al realizar cambios en una clase. Puede ser conveniente modularizar el código o incluso utilizar la herencia. <https://refactoring.guru/es/smells/divergent-change>
 - **Cirugía de escopeta:** Cuando después de un cambio en una clase se tienen que realizar muchos cambios en muchas clases diferentes. Puede ser interesante englobar todos los comportamientos en una clase.
<https://refactoring.guru/es/smells/shotgun-surgery>
- **Prescindibles:** Código no necesario y que cuya ausencia haría el código más limpio eficiente y fácil de entender.
 - **Código duplicado:** Cuando el mismo código se encuentra en más de un lugar. Es conveniente extraer el método.
<https://refactoring.guru/es/smells/duplicate-code>
- **Acopladores:** Cuando la clase está mal acoplada, ya sea por exceso o defecto.

- **Envidia:** Cuando un método está más tiempo comunicándose con funciones o datos de otro objeto que con sus propios datos. Suele ser mejor mover estos métodos a la clase con la que más se comunica.
<https://refactoring.guru/es/smells/feature-envy>
- **Grupo de datos:** Cuando hay varios datos que siempre se mueven juntos puede ser conveniente crear un objeto.
- **Datos globales:** Cuanto más restrictivo sea el acceso a un dato más seguro será y más fácil será de controlar los métodos y clases en los que tiene repercusión.
- **Olores de nombre:** Cuando un nombre (de una variable, de un método, de una clase, etc) no representa de manera concisa su contenido.'

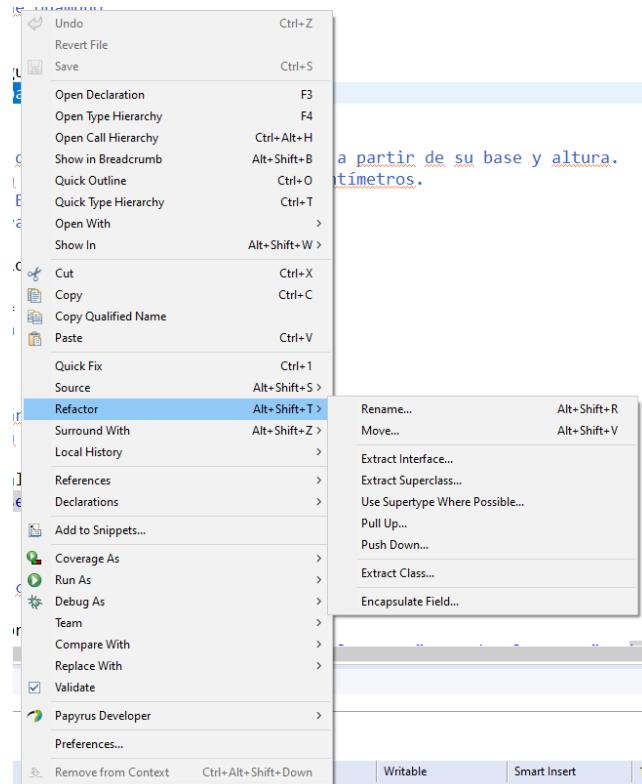
Refactorización de clases:



Refactorización de métodos:



Refactorización de variables:



5.13. Atajos de teclado

Un atajo de teclado es una tecla o secuencia de teclas que efectúa una acción definida previamente.

Si usamos un entorno de trabajo muy frecuentemente es conveniente conocer los atajos de teclado para trabajar más rápido.

- **Ctrl + espacio:** Autocompletar código.
- **Ctrl + SHIFT + F:** Formatear texto.
- **Ctrl + S:** Guardar
- **Ctrl + SHIFT + S:** Guardar todos los documentos abiertos.
- **Alt + Flechas:** Navegar.
- **Ctrl + Z:** Comenta el texto seleccionado.
- **sys0 + (ctrl + espacio):** Escribe una impresión por pantalla.
- **if, for, while, do, main, + (ctrl + espacio):** Escribe la expresión correspondiente por pantalla.
- **Ctrl + Z:** Deshacer.
- **Ctrl + Y:** Rehacer.

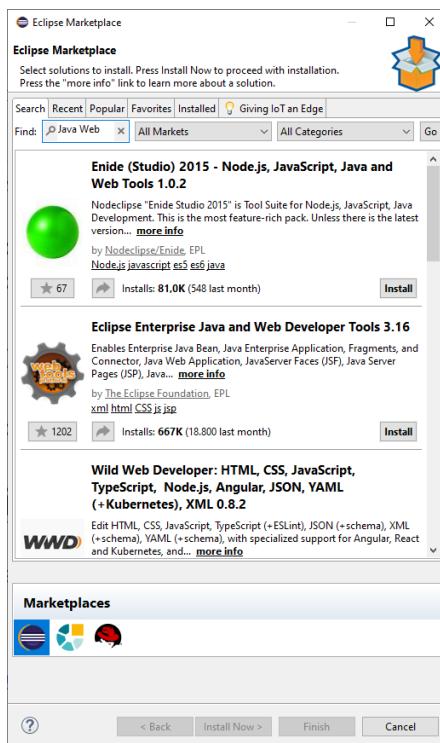
5.14. Control de versiones

Este apartado se explicará en profundidad en el tema de control de versiones.

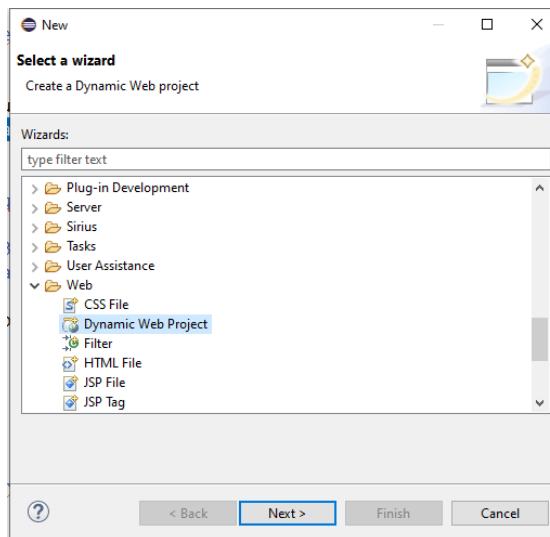
5.15. Plugins

En la pestaña de ayuda se encuentra el Marketplace de Eclipse.

Allí buscamos el plugin que necesitamos. En este caso vamos a instalar el plugin de Java EE.



Ahora dentro de los proyectos web debería aparecer la siguiente opción.



Hay que añadir un servidor local para probar el proyecto web.

<https://tomcat.apache.org/download-80.cgi> de aquí se puede descargar la versión 8.5.75 del servidor de aplicaciones Tomcat.

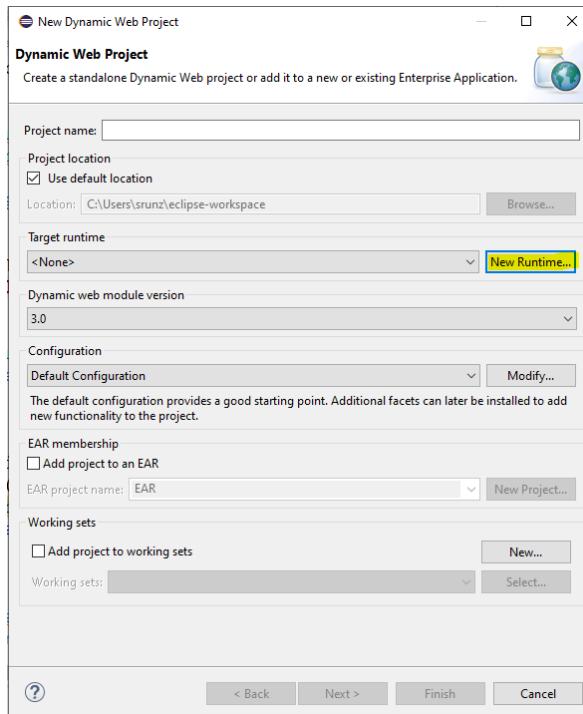
8.5.75

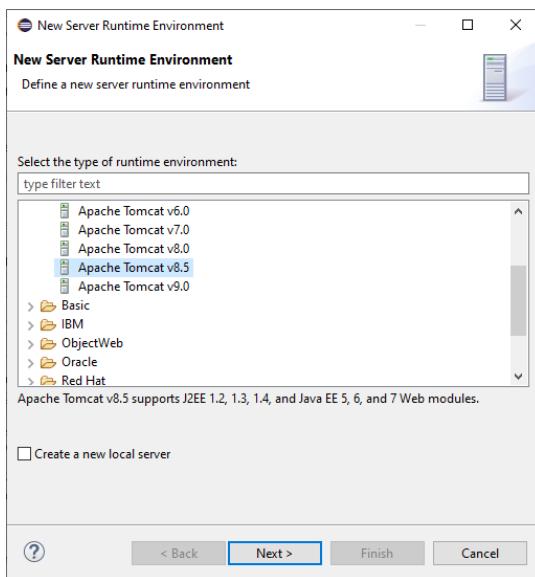
Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

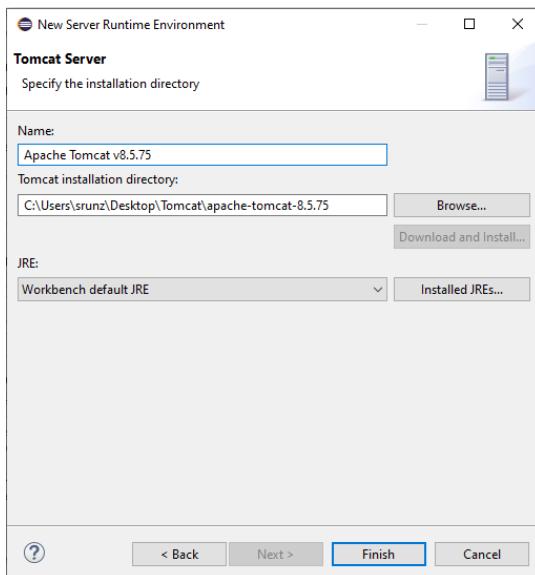
- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Extras:
 - [JMX Remote jar \(pgp, sha512\)](#)
 - [Web services jar \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

Descomprimid el Tomcat en una carpeta para tenerlo localizado ya que, cuando creamos el Dynamic Web Project, hay que establecer el servidor en el que se ejecutará el proyecto.

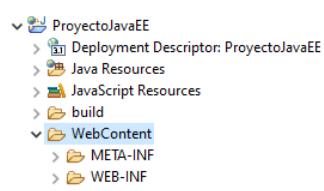


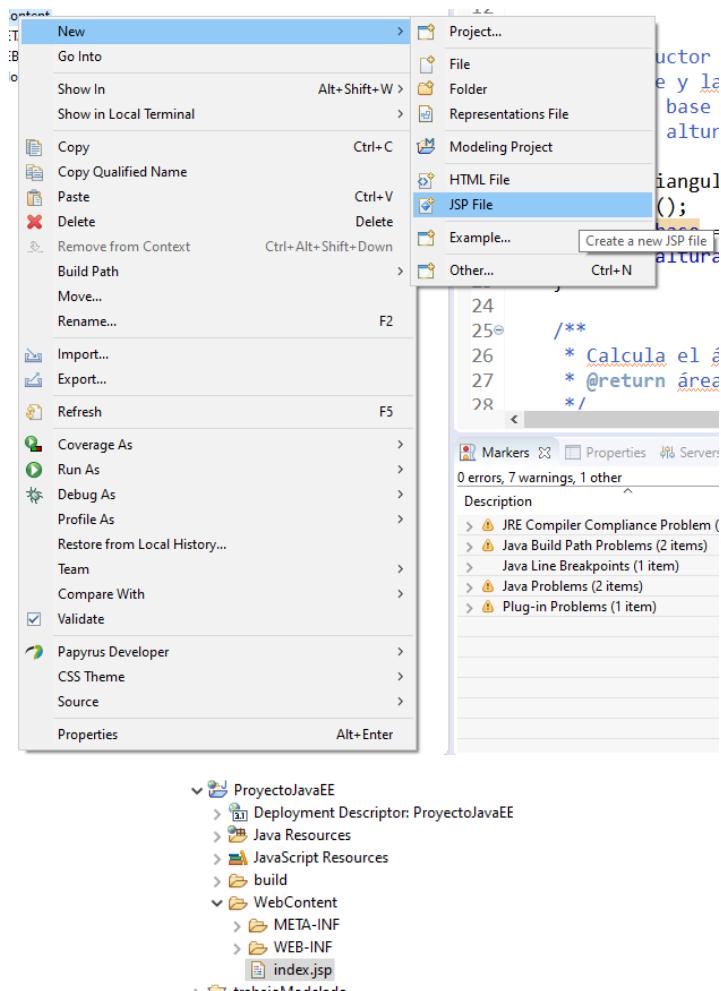


Y se selecciona la ubicación del Tomcat descargado.



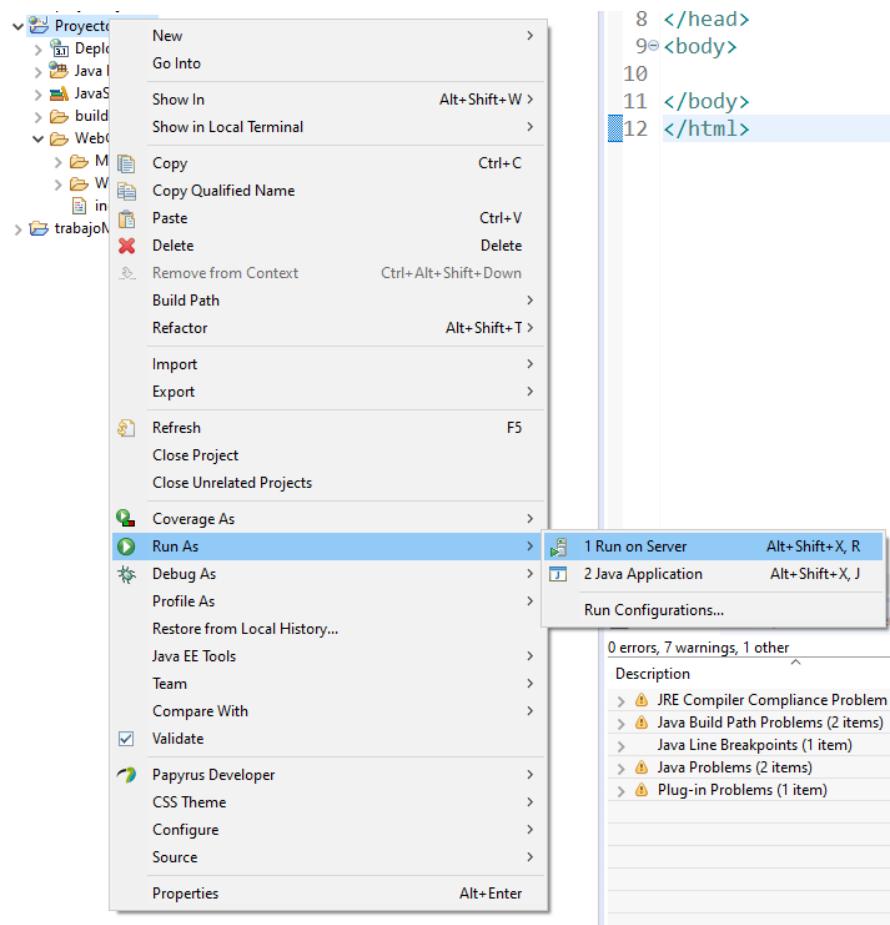
Una vez se cree el proyecto, dentro de WebContent añadimos un jsp.



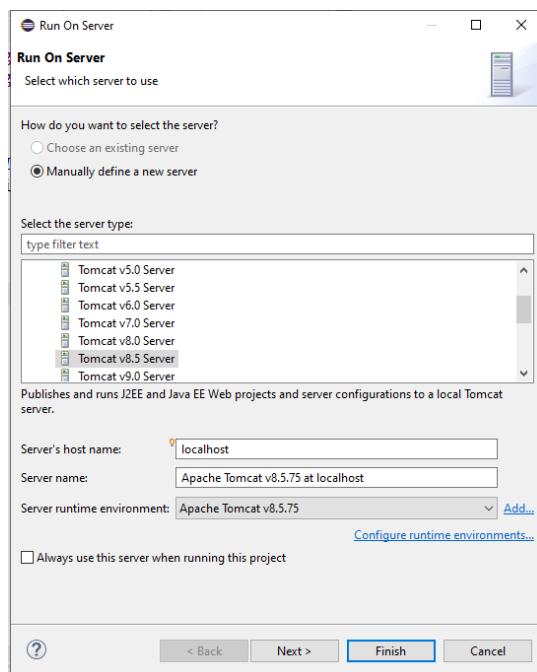


Añadid un contenido al jsp para que se imprima por pantalla (un hola mundo o algo)

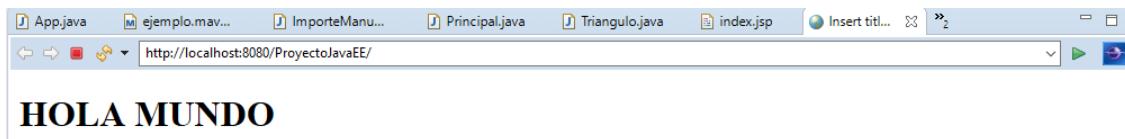
Seleccionamos el proyecto y lo ejecutamos en el servidor.



Seleccionamos el servidor.

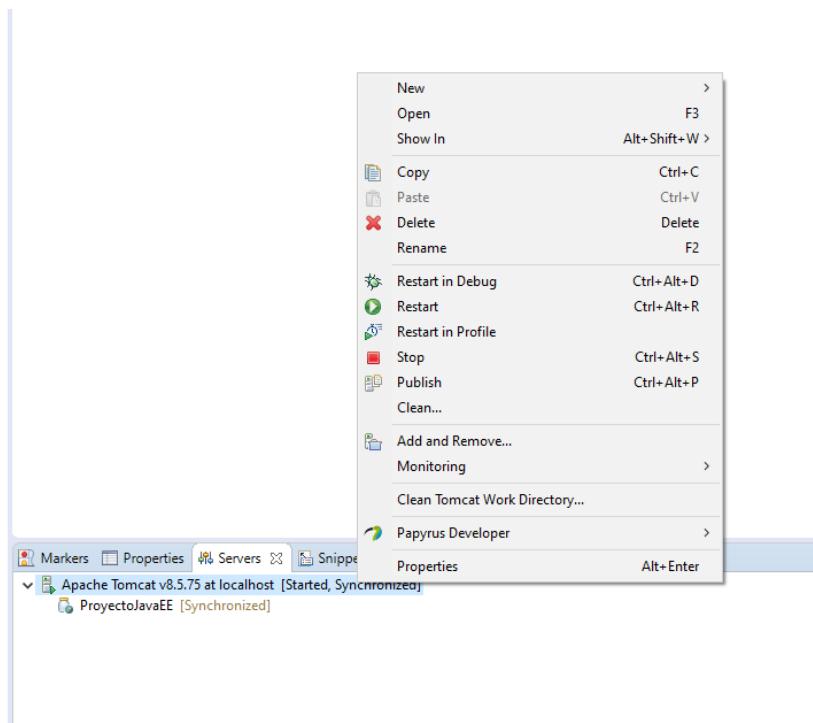


Ahora el servidor estará accesible.



Si estás conectados a la misma red que vuestros compañeros podéis acceder a sus proyectos por la ip.

Lo suyo es que, cuando manejemos servidores dentro de Eclipse, mostremos la vista de servidores.



<https://marketplace.eclipse.org/category/free-tagging/java-ee>

EJERCICIO 23: Importa el proyecto de Java EE que se encuentra en la plataforma en eclipse y desplíégalo. Explica lo que hace este proyecto.

EJERCICIO 24: Accede al proyecto de un compañero.

5.16. Bases de datos

Desde Eclipse se pueden gestionar también bases de datos.

Esta funcionalidad no está instalada por defecto en el paquete básico de Eclipse, por tanto, vamos a tener que descargar el plugin de Eclipse Data Tools Platform que permite trabajar con nuestra base de datos desde el entorno de desarrollo. De tal manera que desde el propio Eclipse vamos a poder crear tablas, lanzar consultas, tendremos editores para ficheros con sql, ...

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

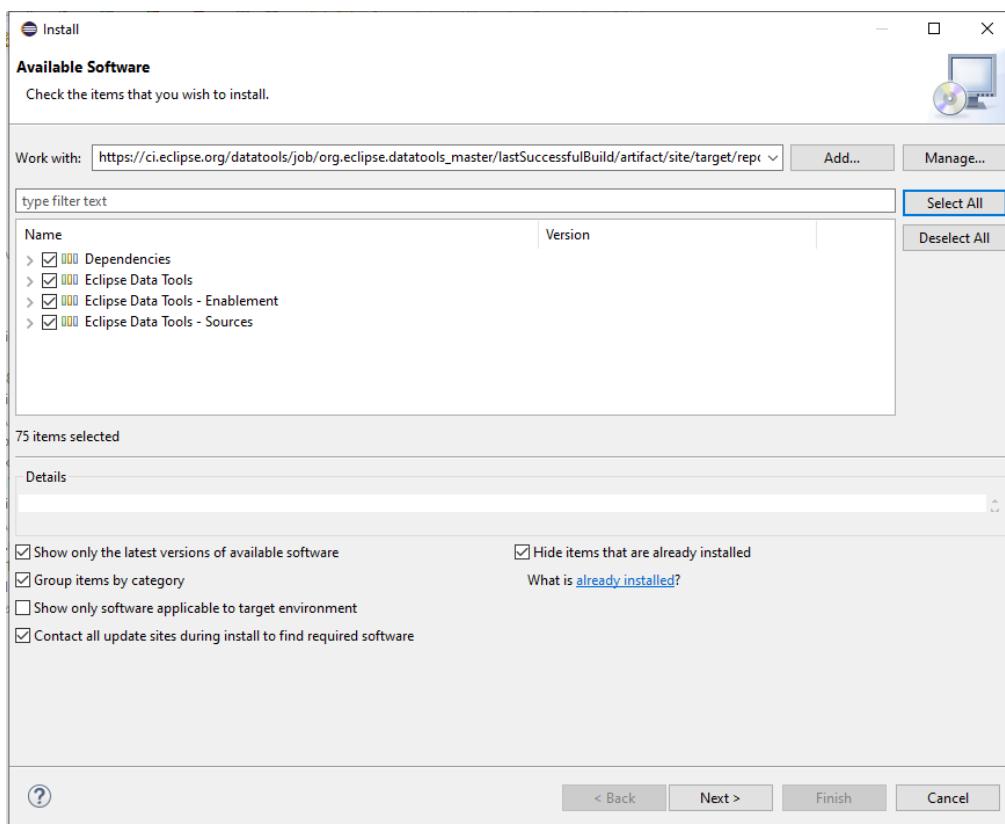
Tomamos esta URL.

Eclipse Data Tools Platform (DTP) Project Downloads

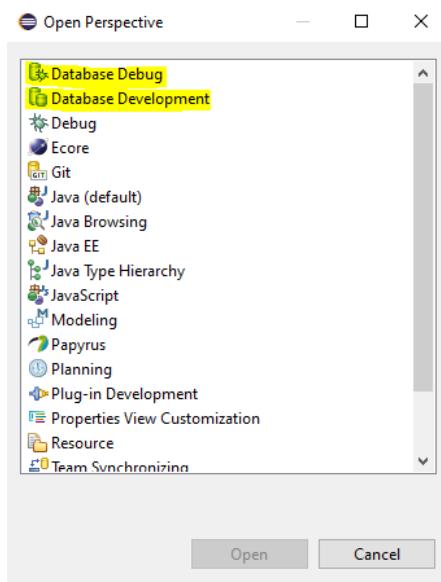
DTP 1.14.x

- 1.14.x (latest CI build): [update site](#) or repo zip
- 1.14.103 (Simrel 2018-12): [update site](#) or repo zip
- 1.14.102 (Simrel 2018-09): [update site](#) or repo zip
- 1.14.100 (Photon): [update site](#) or repo zip
- 1.14.1 (Oxygen): [update site](#) or repo zip
- 1.14.0 (Oxygen): [update site](#) or repo zip

Y la añadimos en Eclipse dentro de Install New Software.

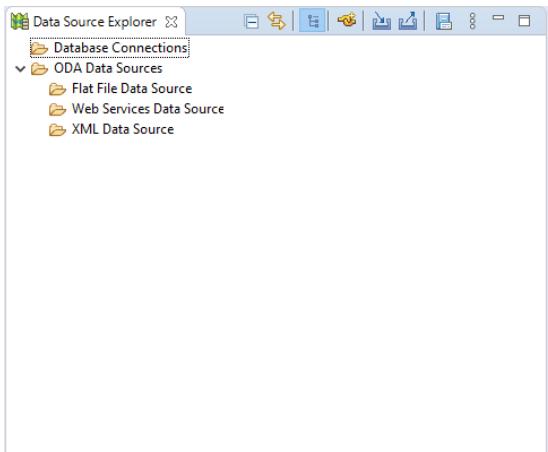


Ahora cuando le damos a abrir perspectiva aparecerán estas dos.

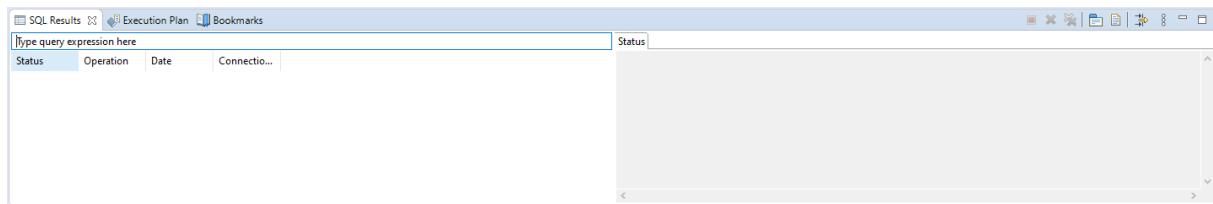


El explorador de Fuentes de Datos permite gestionar desde bases de datos hasta Servicios Web o XMLs de los que los proyectos extraen información

En este caso solo nos interesa crear una conexión a una base de datos.



En SQL Results se nos muestra el resultado de las consultas SQL que lancemos ya sean manualmente o automáticamente.



Para conectarnos a la base de datos desde un entorno de Java vamos a necesitar un Driver llamado JDBC (Java Database Connectivity).

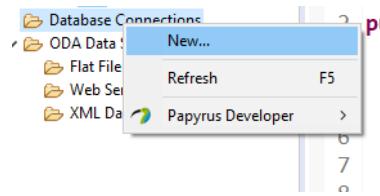
<https://dev.mysql.com/downloads/connector/j/>

File Type	Version	Size	Action
Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.28.tar.gz)	8.0.28	4.0M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.28.zip)	8.0.28	4.8M	Download

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

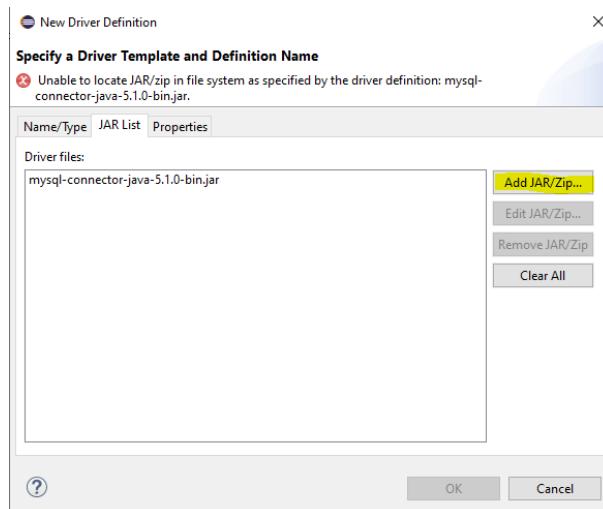
Lo descargamos y lo extraemos en una carpeta que tengamos localizada.

Ahora vamos a crear una conexión a la base de datos.

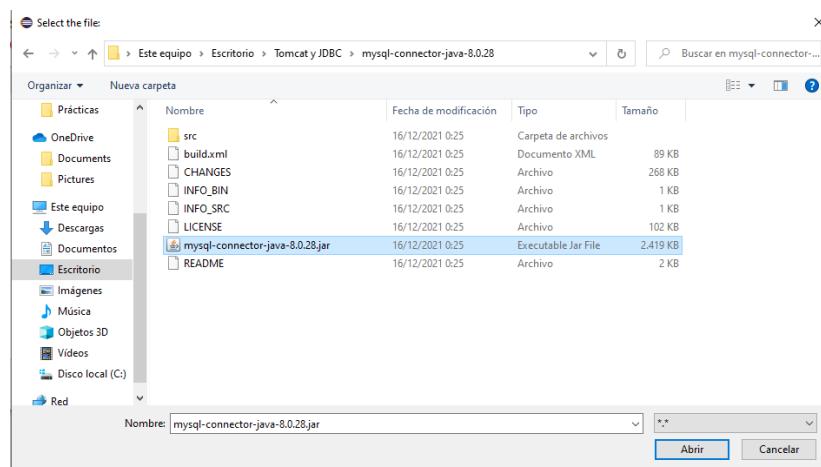


Seleccionamos le tipo de base de datos MySQL que es el que tiene XAMPP y es al que nos conectaremos.

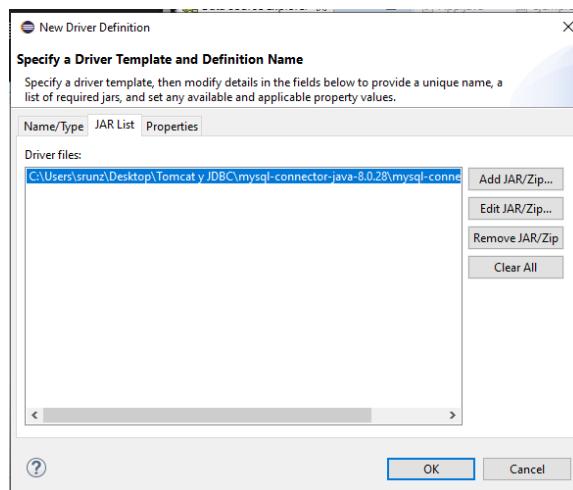
Añadimos el Driver.



Hay que seleccionar el JAR que se encuentra dentro de la biblioteca descargada.

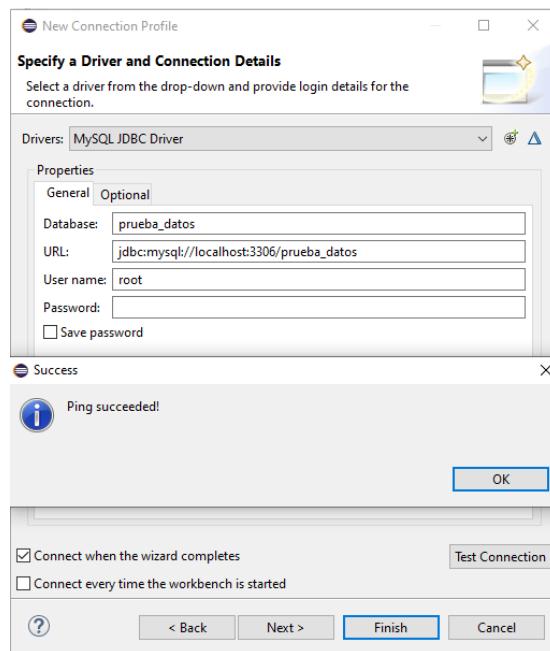


Dejamos tan solo el JAR que hemos descargado.



Como no hemos puesto contraseña en XAMPP nos podemos conectar directamente.

Para probar la conexión pinchamos en Test Connection



Ya nos vamos a poder conectar a nuestra base de datos.

Y podemos acceder a los datos que esta tiene.

The screenshot shows the MySQL Workbench interface. On the left, the 'Data Source Explorer' pane is open, displaying a tree structure of database connections, schemas, and tables. A connection to 'MySQL XAMPP (MySQL v. 5.5.0.4.21-MariaDB)' is selected, showing the 'prueba_datos' schema with its 'jugadores' table. On the right, a table editor window titled 'Main.java jugadores' displays the data from the 'jugadores' table:

id [INT]	nombre [VARCHAR(50)]	apellidos [VARCHAR(50)]	edad [INT]	equipo [INT]
1	Gerard	Moreno	29	0
2	Pau	Torres	25	0
3	Lucas	Ocampos	27	0
4	Radamel	Falcao	35	0

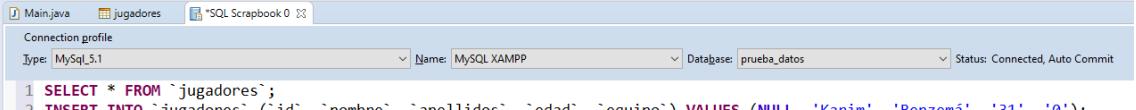
Se puede modificar la base de datos a mano.

This screenshot shows the MySQL Workbench interface with a table editor and the 'SQL Results' pane. The table editor shows the same data as before. In the 'SQL Results' pane, a successful insert operation is shown:

```
insert into `jugadores`(`id`, `nombre`, `apellidos`, `edad`, `equipo`) values(null, 'Ronaldo', 'Nazario', 50, 0)
Query OK, 1 row affected (0.00 sec)
```

También se pueden lanzar consultas SQL directamente sobre la base de datos.

This screenshot shows a context menu for the 'jugadores' table in the 'prueba_datos' schema. The 'Open SQL Scrapbook' option is highlighted. The menu also includes other options like Connect, Disconnect, Ping, Refresh, and Properties.



```
1 SELECT * FROM `jugadores`;
2 INSERT INTO `jugadores` (`id`, `nombre`, `apellidos`, `edad`, `equipo`) VALUES (NULL, 'Karim', 'Benzemá', '31', '0');
```

EJERCICIO 25: Conéctate a la base de datos de usuarios y añade algún usuario.

EJERCICIO 26: Utiliza, al menos, dos sentencias sql sobre esta base de datos.

REFERENCIAS

Libro Entornos de Desarrollo de la editorial Garceta (2^a edición), escrito por Alicia Ramos Martín y M^a Jesús Ramos Martín.

Libro Entornos de Desarrollo de José Luis Comesaña

<https://www.mclibre.org/consultar/php/otros/xampp-instalacion-windows.html#:~:text=Los%20componentes%20m%C3%ADnimos%20que%20instala,menos%20instalar%20MySQL%20y%20phpMyAdmin.>

<https://javadesdecero.es/fundamentos/diferencias-jdk-jre-jvm/>

<https://medium.com/@neossoftware/diferencias-entre-jdk-y-open-jdk-f6b8f4bac924>

<https://tutobasico.com/installar-eclipse-windows/>

<https://www.genbeta.com/desarrollo/eclipse-ide>

<http://www.comscigate.com/cs/valle/vistas.htm>

<https://es.myservername.com/what-are-pom#:~:text=Project%20Object%20Model%20o%20POM,informaci%C3%B3n%20importante%20sobre%20el%20proyecto.>

https://sites.google.com/a/colegiomiguelaleman.edu.mx/sistemas/inicio/compu-que-o_0/-que-son-los-atajos-del-teclado

<http://entornos.codeandcoke.com/doku.php?id=apuntes:refactorizacion>

<http://www.cienciamx.com/index.php/tecnologia/tic/5226-herramienta-para-deteccion-de-bad-smells-en-sistemas-computacionales-nota-informativa#:~:text=%2D%20Los%20bad%20smells%20son%20errores,dif%C3%ADcil%20de%20entender%20y%20mantener.>

<https://refactoring.guru/es/smells/duplicate-code>

<https://refactoring.guru/es/refactoring/smells>

<https://jarroba.com/configurar-formateo-de-codigo-en-eclipse/#:~:text=Lo%20primero%20que%20debemos%20hacer,a%20%22Eclipse%20%3E%20Preferences%22.&text=Hasta%20a%C3%AD%20hemos%20visto%20como,configurar%20una%20serie%20de%20par%C3%A1metros.>

<https://qastack.mx/programming/6937825/how-can-i-change-eclipse-theme#:~:text=En%20la%20versi%C3%B3n%20de%20eclipse,y%20haga%20clic%20en%20Aceptar.>

<https://www.adictosaltrabajo.com/2007/08/06/eclipse-dtp/>

http://fernando-gaitan.com.ar/wp-content/uploads/tablas_relaciones.png