

Preparando nuestro entorno

Desarrollo de interfaces

Apuntes: Jesús Redondo García

Entorno de desarrollo: VSCODE

Para este curso vamos a emplear la herramienta VSCode, el editor de código de código abierto y gratuito de Microsoft:



Ventajas:

- Ligero, rápido y totalmente gratuito.
- Permite emplear otros lenguajes y desarrollos.
- Gran integración con herramientas de control de versión.
- Tendencia actual.
- Multitud de plugins, actualizaciones, snippets, temas...

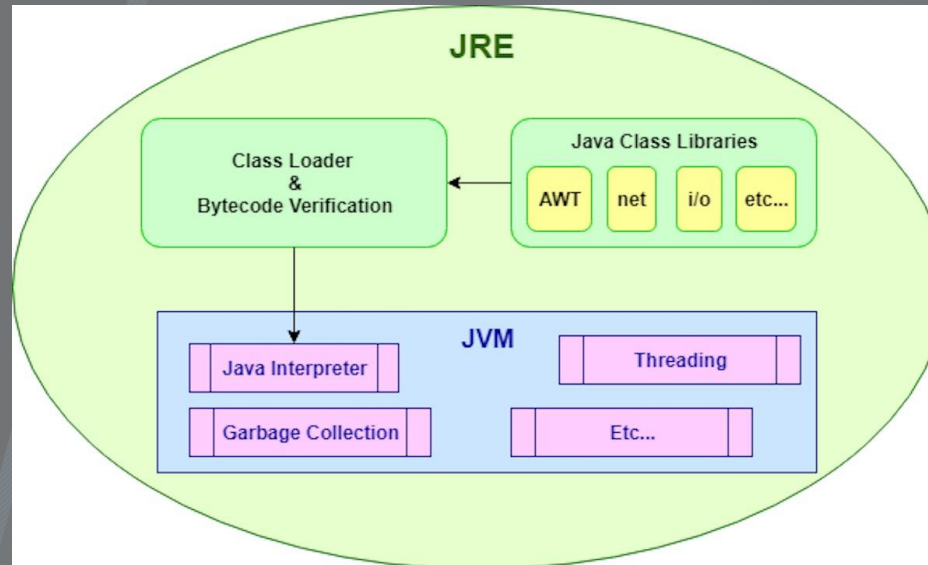
Inconvenientes:

- Entorno no familiar. Periodo de adaptación.
- Diferencias con Eclipse/Netbeans (para mejor)

JDK: Definición

Java Development Kit (JDK) es el conjunto de herramientas que necesitamos para poder desarrollar aplicaciones Java en nuestro equipo. Entre otras cosas el JDK contiene:

- **JRE:** Es la capa software que se sitúa por encima del S.O. y nos permite ejecutar programas Java. Permite utilizar la JVM (máquina virtual de Java) y lanzar un entorno de ejecución que cargue clases y las ejecute.



- **Compilador:** Convierte los archivos java en Bitcode.
- **Depurador:** Permite ejecutar paso a paso e inspeccionar memoria.
- **Otras Herramientas:** Línea de comandos, Applet viewer, Document Viewer...

JDK: Definición

Java Development Kit (JDK)

Existen varias versiones e implementaciones del JDK.

Implementaciones: [\(más info aquí\)](#)

- **Oracle JDK:** La versión oficial de Oracle. A partir de la 8 no es gratuita para desarrollo profesional. Se supone más estable, con mejor rendimiento. Mejor recolector de basura, algunas herramientas extra.
- **Open JDK:** Licencia abierta. Desarrollo profesional o personal gratuito. Buen ritmo de actualizaciones. A partir de v11 idéntico casi a Oracle JDK.

Versiones:

Ahora mismo,
la última
versión de
larga duración
recomendada
es la
Java SE 11
(LTS).

Java SE 8 (LTS)	March 2014	January 2019 for Oracle (commercial) December 2020 for Oracle (personal use) At least May 2026 for AdoptOpenJDK At least June 2023 ^[7] for Amazon Corretto	December 2030
Java SE 9	September 2017	March 2018 for OpenJDK	N/A
Java SE 10	March 2018	September 2018 for OpenJDK	N/A
Java SE 11 (LTS)	September 2018	At least August 2024 ^[7] for Amazon Corretto October 2024 for AdoptOpenJDK	September 2026
Java SE 12	March 2019	September 2019 for OpenJDK	N/A
Java SE 13	September 2019	March 2020 for OpenJDK	N/A
Java SE 14	March 2020	September 2020 for OpenJDK	N/A
Java SE 15	September 2020	March 2021 for OpenJDK	N/A
Java SE 16	March 2021	September 2021 for OpenJDK	N/A
Java SE 17 (LTS)	September 2021	TBA	TBA

Legend: ■ Old version ■ Older version, still maintained ■ Latest version ■ Future release

https://en.wikipedia.org/wiki/Java_version_history

JDK: Instalación

Linux:

Desde línea de comandos podemos instalar la versión de JDK que deseemos, en nuestro caso la open-JDK-11

```
sudo apt install openjdk-11-jdk
```

No nos podemos olvidar de instalar las fuentes, que nos permitirán ver el código original de las clases de Java así como sus comentarios:

```
sudo apt-get install openjdk-11-source
```

La instalación del JDK se encuentra en **/usr/lib/jvm**

Podemos seleccionar la versión por defecto de java de nuestra máquina con el siguiente comando:

```
sudo update-alternatives --config java
```

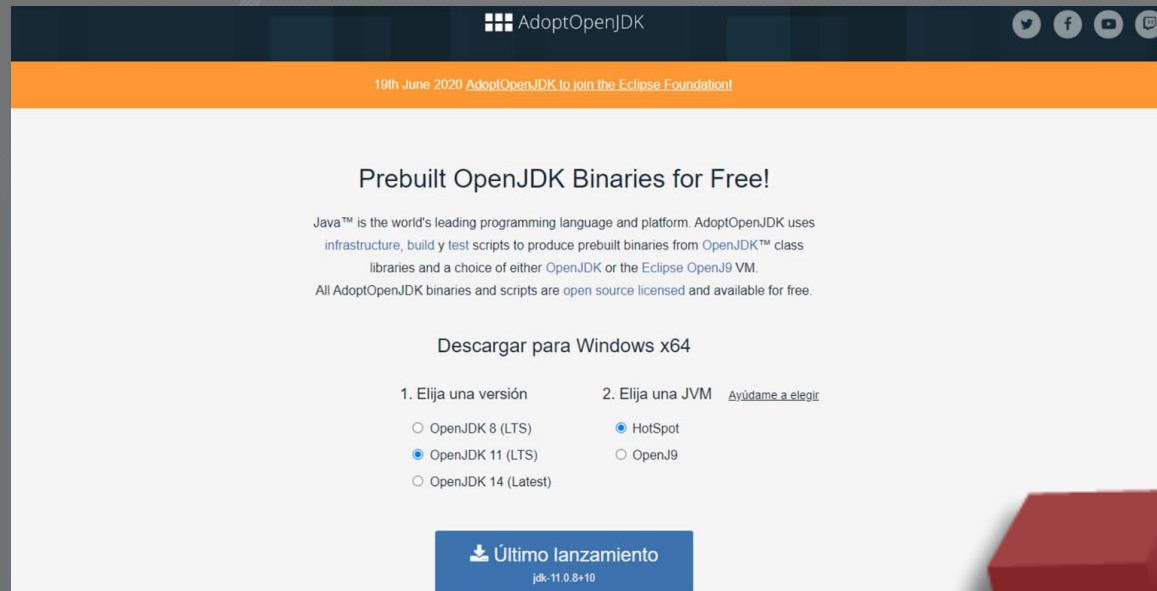

JDK: Instalación

Windows:

- 1) Lo podemos hacer manualmente, habría que descargar las fuentes, seleccionar el PATH y el JAVA_HOME y configurar el IDE.

<https://stackoverflow.com/a/52531093>

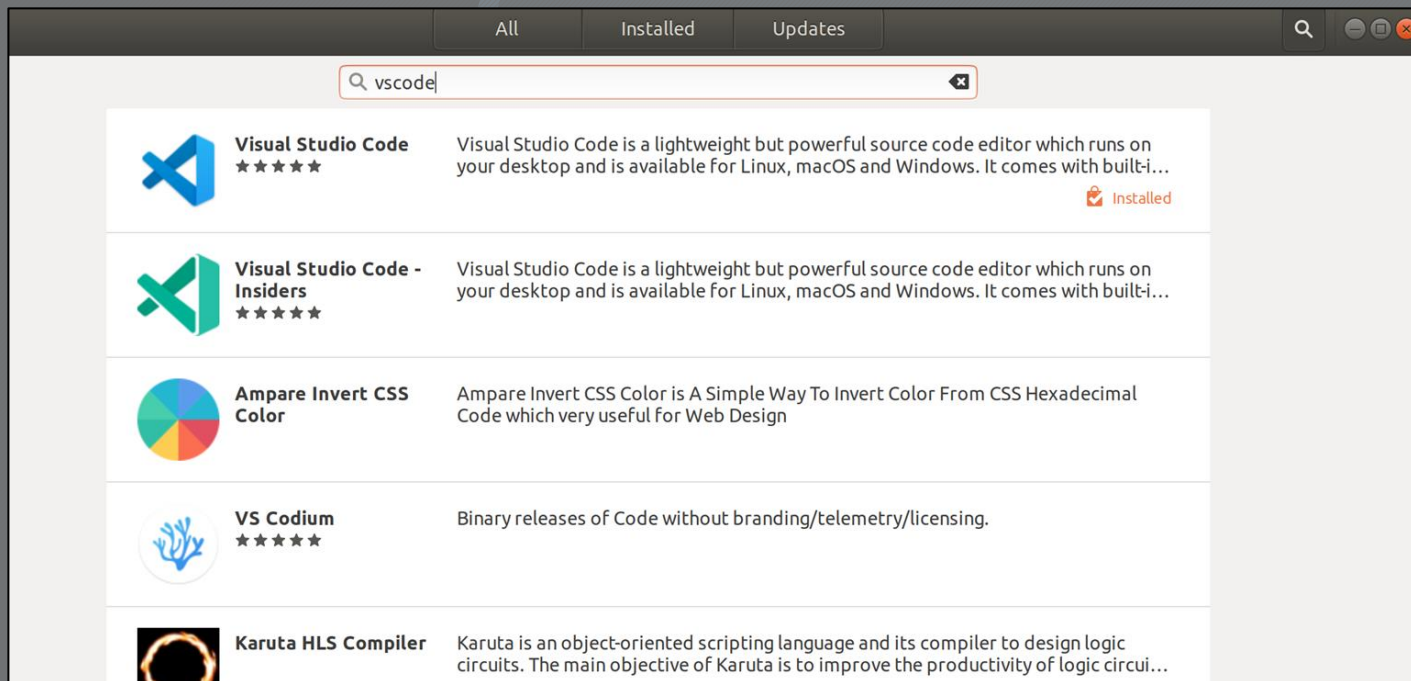
- 1) O bien podemos emplear un instalador que nos haga todos esos pasos automáticamente. Si es posible, y tenemos permisos, es la opción más sencilla. Descargar instalador desde <https://adoptopenjdk.net/>:



VSCODE: Instalación

Linux:

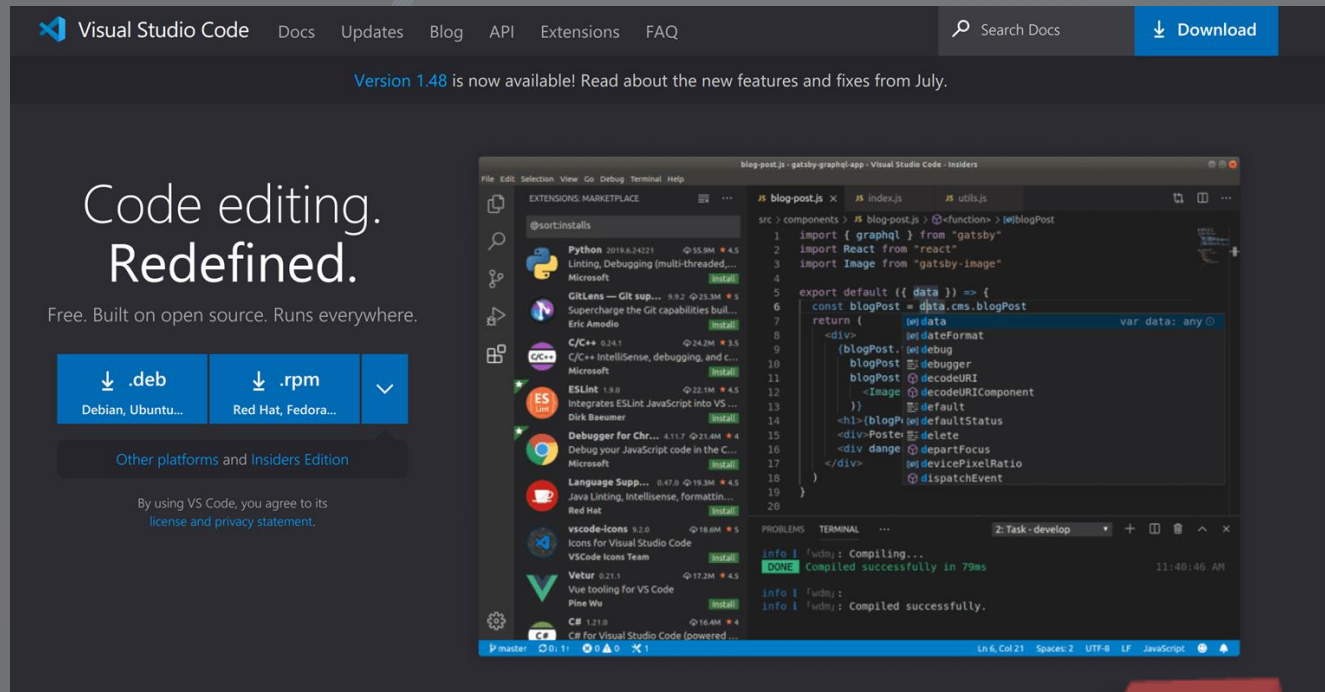
Desde el gestor de software del S.O. nos permitirá instalar VSCode.



VSCODE: Instalación

Windows:

Desde la web de VSCode podemos descargar el instalador autoejecutable:



VSCODE: Plugins básicos

Para Java:

Una de las principales virtudes de VSCode es lo sencillo que es modificar el editor. Existen plugins para casi cualquier lenguaje o comportamiento del editor que podamos pensar. Para trabajar con Java, desde la documentación de VSCode se nos recomiendan una serie de plugins básicos: <https://code.visualstudio.com/docs/languages/java>

El Plugin que recopila todos los otros plugins basicos necesarios se denomina **“Java Extension Pack”**. Puedes descargarlo desde la pestaña de extensiones y buscando Java:



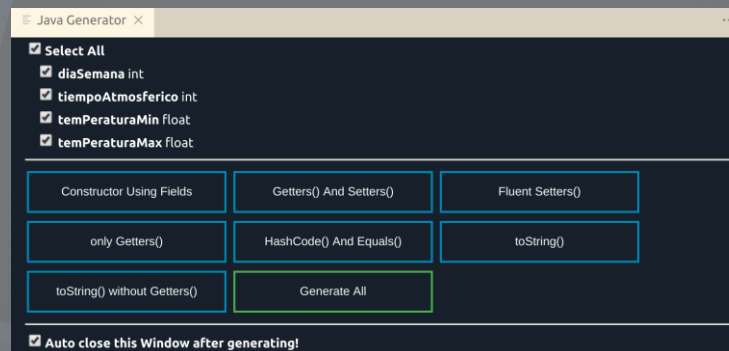
VSCODE: Plugins básicos

Para Java:

Otro plugin indispensable es la extensión Java Code Generators. Este plugin nos permite añadir los métodos básicos (getters, setters, constructores, comparadores, hash...) de manera automática a nuestra clase:




La manera más sencilla de emplearlo es con click derecho sobre el editor abierto en la clase y click “Java: Generator GUI”:



VSCODE: Plugins básicos

Para compartir el proyecto:

Podemos compartir y editar nuestro proyecto mediante el plugin “Live Share Extension Pack”:



Live Share Extension Pack 0.4.0

Collection of extensions that enable real-time ...

Microsoft

473K

★ 4.5

⚙️

Para poder emplearlo es necesario tener una cuenta de GitHub. En el caso de no disponer de una, emplead el mail del instituto para crearla:

<https://github.com/join>

Join GitHub

Create your account

Username *

Email address *

VSCODE: Plugins básicos

Para compartir el proyecto:

Para compartir nuestro workspace:

- 1) Pulsamos sobre la pestaña del plugin:
- 2) Iniciamos una colaboración (con edición o no) y compartimos el link:



✓ SESSION DETAILS

Join collaboration session...

Start collaboration session...

Start read-only collaboration session...

- 1) Los demás usuarios se podrán conectar y visualizar TODO el workspace, desafortunadamente, no podremos ejecutar el código en remoto.

NOTA: Si hubiese problemas al unirse a una sesión, con cerrar y volver a entrar en VSCode es suficiente.

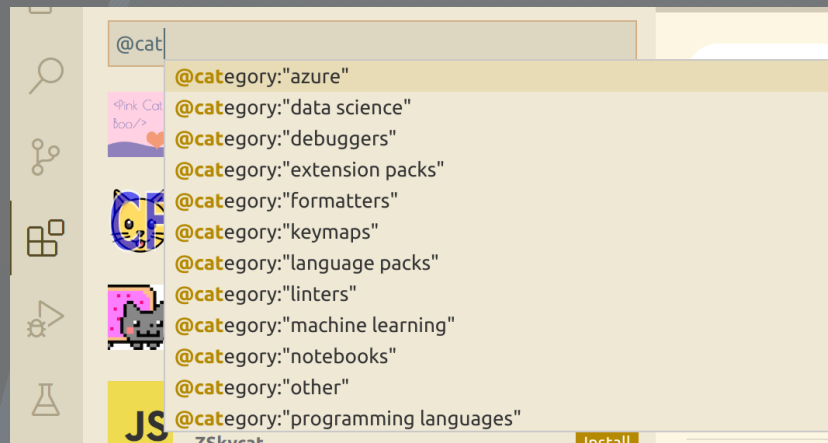
VSCODE: Plugins básicos

Otros plugins:

Existen multitud de utilidades que te podrían resultar interesantes en el marketplace.

Investiga por tu propia cuenta, busca en foros y prueba. Instalar y desinstalar plugins es extremadamente sencillo.

Recuerda que puedes filtrar la búsqueda por muchos criterios, prueba a poner el @ en el campo de búsqueda:

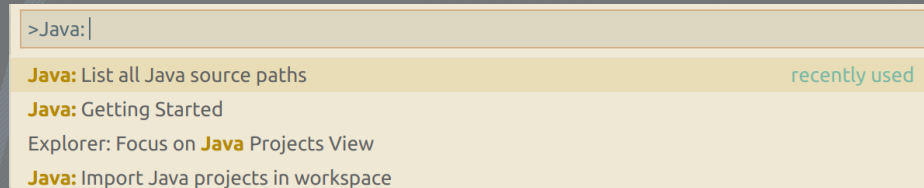


VSCODE: Tecla mágica F1

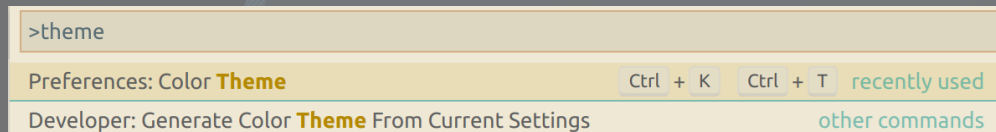
La tecla mágica

Si sólo puedes recordar una combinación de teclas en VSCode, que sea F1. Mediante esta tecla puedes buscar cualquier opción del procesador.

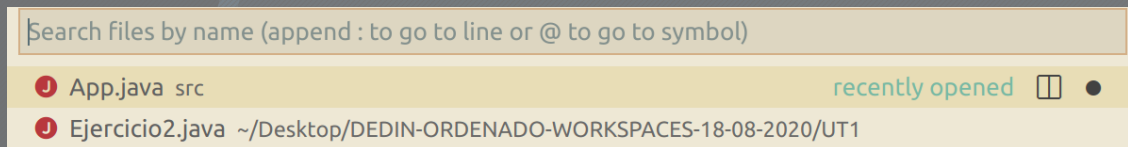
Getting Started con Java:



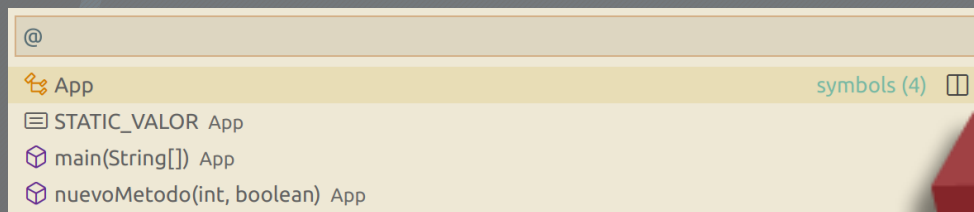
Seleccionar tema (el tema que empleo es *Solarized Light*):



Si usamos F1 y borramos el símbolo “>” buscaremos por los archivos que tengamos en el explorador:



Si colocamos una “@” buscaremos métodos, clases, atributos...



VSCODE: Otros atajos

Ctrl + click sobre símbolo: Desplazarse a la implementación.

F2: Renombrar símbolo.

F5: Compilar o depurar (si hay puntos de interrupción activos).

F8: Ir al siguiente error.

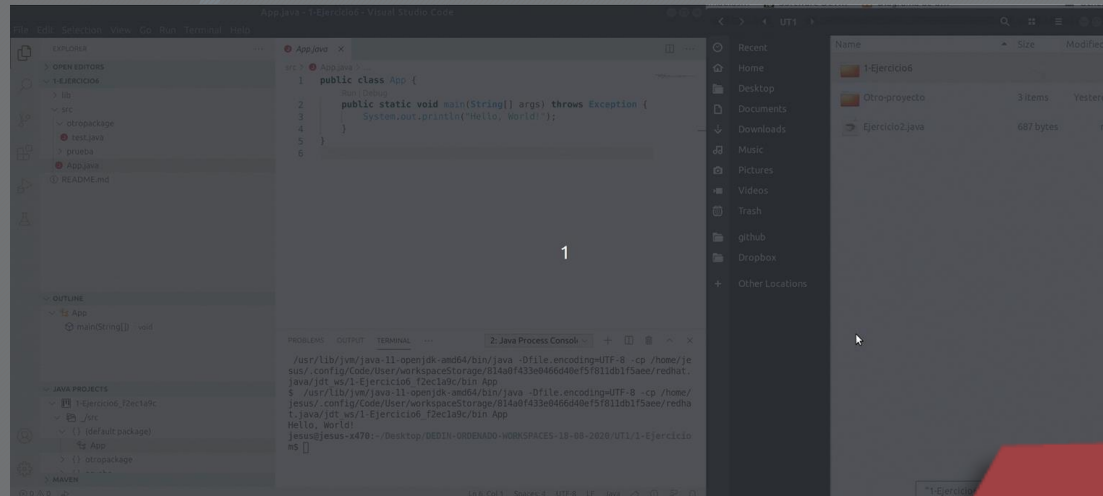
Ctrl .: Fixes para el error actual.

Alt + (tecla arriba o tecla abajo): Desplazar la línea arriba o abajo.

Ctrl + (tecla derecha o izquierda): Moverse entre palabras (rápido).

Ctrl D: Seleccionar toda la palabra actual.

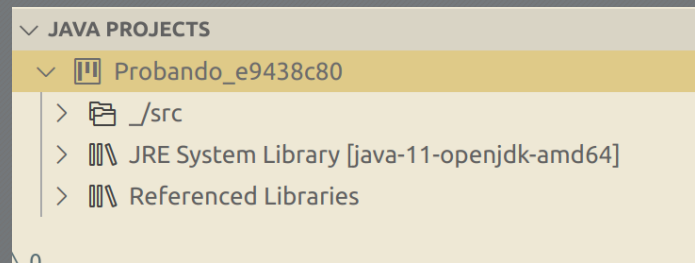
Ctrl C + Ctrl V (sin nada seleccionado): Duplica la línea



VSCODE: Gestión de proyectos

- Workspaces: Cuando se trabajan con proyectos que tienen dependencias y cierta complejidad (es lo normal). Existen dos modos diferentes:
 - Folder Workspace: El concepto es que el workspace sólo sirve para un proyecto. Es el modo de apertura por defecto si abrimos los proyectos arrastrándolos al lado derecho de VSCode. Es el modo de apertura que emplearemos por defecto al crear proyectos y el que más emplearemos a lo largo del curso.

Para crear un proyecto nuevo de JAVA: Hacemos uso de la tecla F1 -> Java: Create java -> No build tools -> Seleccionamos la ruta del proyecto -> Nombre.



VSCODE: Gestión de proyectos

- Workspaces:
 - Multi-root Workspace: VSCode permite trabajar con varios workspaces de carpeta abiertos a la vez. Esto puede ser útil por ejemplo cuando revises los proyectos de una unidad de trabajo.

Para abrir más proyectos con multi-root se puede hacer mediante File -> Add Folder to Workspace -> Seleccionar uno o varias carpetas para añadirlas a VSCode.











Otra opción es arrastrar las carpetas al **explorador** y seleccionar la opción de add to Workspace:



VSCODE: Organización

Para facilitar la labor de gestionar nuestros ejercicios se propone la siguiente estructura de carpetas y proyectos para el curso:

- Una carpeta raíz para el módulo: DEDIN.
- Cada unidad de Trabajo consta de una carpeta distinta UT1, UT2...
- Cada Ejercicio propuesto de cada unidad de trabajo es un proyecto Java único nuevo que se sitúa dentro de la carpeta correspondiente de su Unidad de Trabajo.

 UT1	4 items	11:00
 UT2	0 items	11:22
 UT3	0 items	11:22
 UT4	0 items	11:22
 UT5	0 items	11:22
 UT6	0 items	11:22
 UT7	0 items	11:22
 UT8	0 items	11:22
 UT9	0 items	11:22
 UT10	0 items	11:22

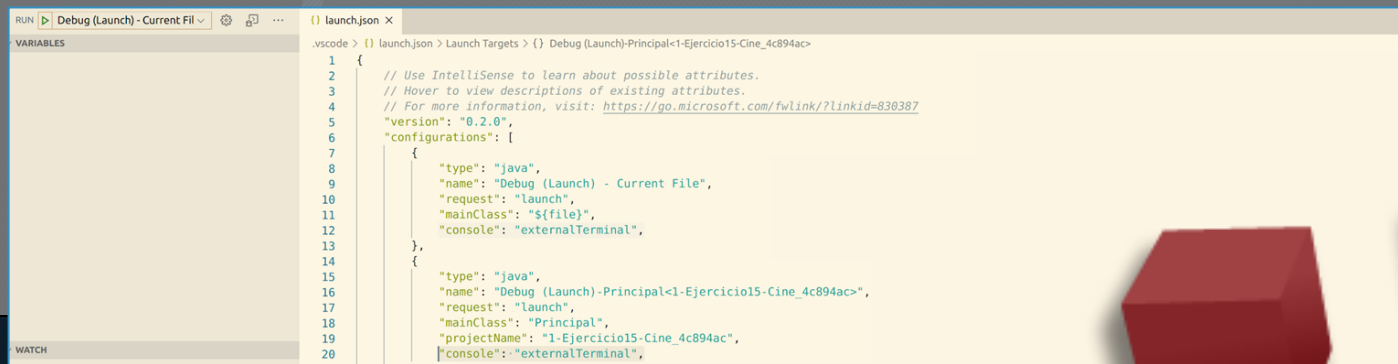
VSCODE: RESOLVIENDO PROBLEMAS

1 - NO SE PUEDEN ESCRIBIR VOCALES CON TILDE EN EL TERMINAL INTERNO DE VSCODE:

El problema es bien conocido y existen numerosos mensajes en el repositorio de VSCode de Github. Por desgracia, todavía sigue sin solución. Para nuestros ejercicios, que a partir del tema 2 van a ser gráficos no supone un problema.

Aún así se proponen las siguientes dos alternativas:

- 1) Escribir la cadena que queramos en otro editor (incluso en VSCode) para copiar y pegarlo. Los acentos se copian sin problema.
- 1) Modificar la configuración de ejecución (desde la pestaña debug o editando config.json), para añadir la línea `"console": "externalTerminal",`



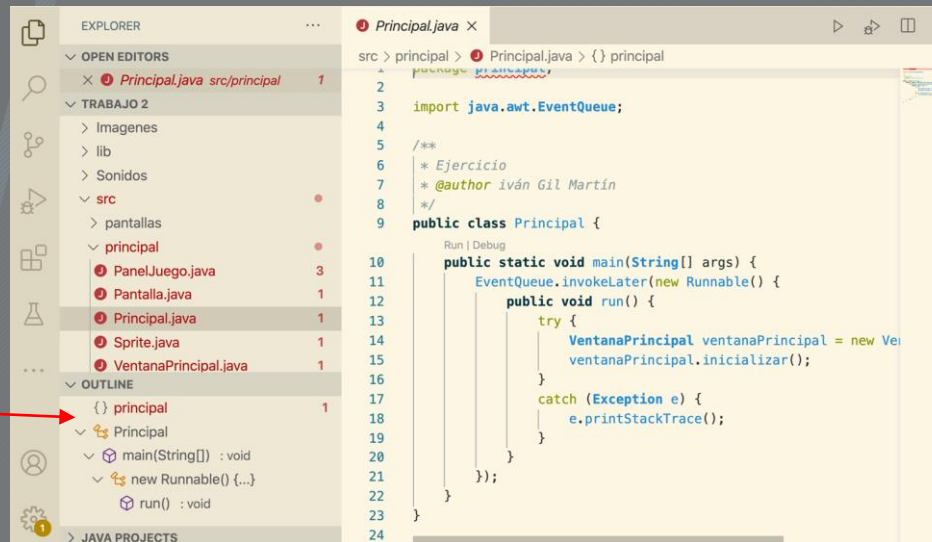
```
.vscode > {} launch.json > Launch Targets > {} Debug (Launch)-Principal<1-Ejercicio15-Cine_4c894ac>
1 {
2   // Use IntelliSense to learn about possible attributes.
3   // Hover to view descriptions of existing attributes.
4   // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5   "version": "0.2.0",
6   "configurations": [
7     {
8       "type": "java",
9       "name": "Debug (Launch) - Current File",
10      "request": "launch",
11      "mainClass": "${file}",
12      "console": "externalTerminal",
13    },
14    {
15      "type": "java",
16      "name": "Debug (Launch)-Principal<1-Ejercicio15-Cine_4c894ac>",
17      "request": "launch",
18      "mainClass": "Principal",
19      "projectName": "1-Ejercicio15-Cine_4c894ac",
20      "console": "externalTerminal",
```


VSCODE: RESOLVIENDO PROBLEMAS

2 - AL ABRIR UN PROYECTO QUE TIENE VARIOS PACKAGES, NO RECONOCE SRC COMO CARPETA DE FUENTES:

El problema ocurre cuando se abre un proyecto en un equipo distinto del que se creó y el proyecto tiene dentro uno o varios packages (por ejemplo principal):

Aquí vemos claramente el problema. Toma la carpeta de fuentes desde “principal” en lugar de “src”.



Para solucionarlo, basta con modificar el “Java Source Path” del proyecto. En la próxima diapositiva tenemos un gif explicativo

VSCODE: RESOLVIENDO PROBLEMAS

2 - AL ABRIR UN PROYECTO QUE TIENE VARIOS PACKAGES, NO RECONOCE SRC COMO CARPETA DE FUENTES:

Gif explicativo

