

CASOS PRÁCTICOS: AUTOMATIZACIÓN DE PROYECTOS

1. CASO PRÁCTICO: ANT.....	1
TERMINAL.....	1
VISUAL STUDIO CODE.....	3
ECLIPSE.....	4
2. CASO PRÁCTICO : MAVEN.....	7
TERMINAL.....	8
VISUALSTUDIO CODE.....	12
ECLIPSE.....	13
3. CASO PRÁCTICO: GRADLE.....	17
TERMINAL.....	17
VISUAL STUDIO CODE.....	18
ECLIPSE.....	23

1. CASO PRÁCTICO: ANT

TERMINAL

NOTA: Para el correcto funcionamiento \$JAVA_HOME debe estar con JDK-17

```
/usr/lib/jvm/java-17-openjdk-amd64
```

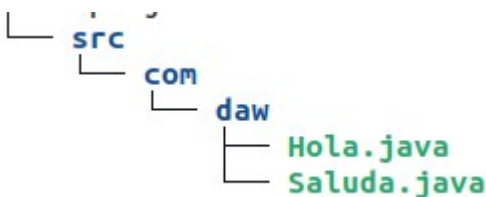
Crear la carpeta principal del proyecto: ProyectoAnt1

```
lo/Documentos$ mkdir ProyectoAnt1
```

Movernos a la carpeta

```
/Documentos$ cd ProyectoAnt1
```

Crear la siguiente estructura de carpetas con los archivos **Hola.java** y **Saluda.java**



Modifica el paquete de los ficheros para adaptarlo a la nueva estructura de carpetas

```
1 package com.daw;
2
3 public class Hola {
4     public static void main(String[] args) {
5         String nombre=args.length>0?args[0]:"";
6         Saluda.saluda(nombre);
7     }
8 }
```

```
1 package com.daw;
2
3 public class Saluda {
4     public static void saluda(String nombre) {
5         System.out.println("Hola "+nombre);
6     }
7 }
```

Modificar crea el fichero build.xml

```

<project name="saluda">
  <target name="clean">
    <delete dir="classes" />
  </target>

  <target name="compile" depends="clean">
    <mkdir dir="classes" />
    <javac includeantruntime="false" srcdir="src/com/daw"
destdir="classes" />
  </target>

  <target name="run" depends="compile">
    <property name="arg0" value="" />
    <java classpath="classes" classname="com.daw.Hola">
      <arg value="{arg0}" />
    </java>
  </target>
</project>

```

Compilación

usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT\$ ant compile
Buildfile: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/build.xml

clean:

compile:

[mkdir] Created dir: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes

[javac] Compiling 2 source files to /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes

BUILD SUCCESSFUL

Total time: 1 second

Comprobación de carpetas

usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT\$ tree -a

```

.
├── build.xml
├── classes
│   └── com
│       └── daw
│           ├── Hola.class
│           └── Saluda.class
├── .classpath
├── .project
├── src
│   └── com
│       └── daw
│           ├── Hola.java
│           └── Saluda.java

```

5 directories, 7 files

Ejecutar

Sin argumentos

```

usuario@HP-ENVY-15-Notebook:/home/usuario/Documentos/PRUEBAS/PRUEBAS ANT$ ant run
Buildfile: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/build.xml

clean:
[delete] Deleting directory /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes

compile:
[mkdir] Created dir: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes
[javac] Compiling 2 source files to /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes

run:
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by org.apache.tools.ant.types.Permissions (file:/usr/share/ant/lib/ant.jar)
WARNING: Please consider reporting this to the maintainers of org.apache.tools.ant.types.Permissions
WARNING: System::setSecurityManager will be removed in a future release
[java] Hola

BUILD SUCCESSFUL
Total time: 1 second

```

Con argumentos

```

usuario@HP-ENVY-15-Notebook:/home/usuario/Documentos/PRUEBAS/PRUEBAS ANT$ ant run -Darg0="María"
Buildfile: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/build.xml

clean:
[delete] Deleting directory /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes

compile:
[mkdir] Created dir: /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes
[javac] Compiling 2 source files to /home/usuario/Documentos/PRUEBAS/PRUEBAS ANT/classes

run:
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by org.apache.tools.ant.types.Permissions (file:/usr/share/ant/lib/ant.jar)
WARNING: Please consider reporting this to the maintainers of org.apache.tools.ant.types.Permissions
WARNING: System::setSecurityManager will be removed in a future release
[java] Hola María

BUILD SUCCESSFUL
Total time: 1 second

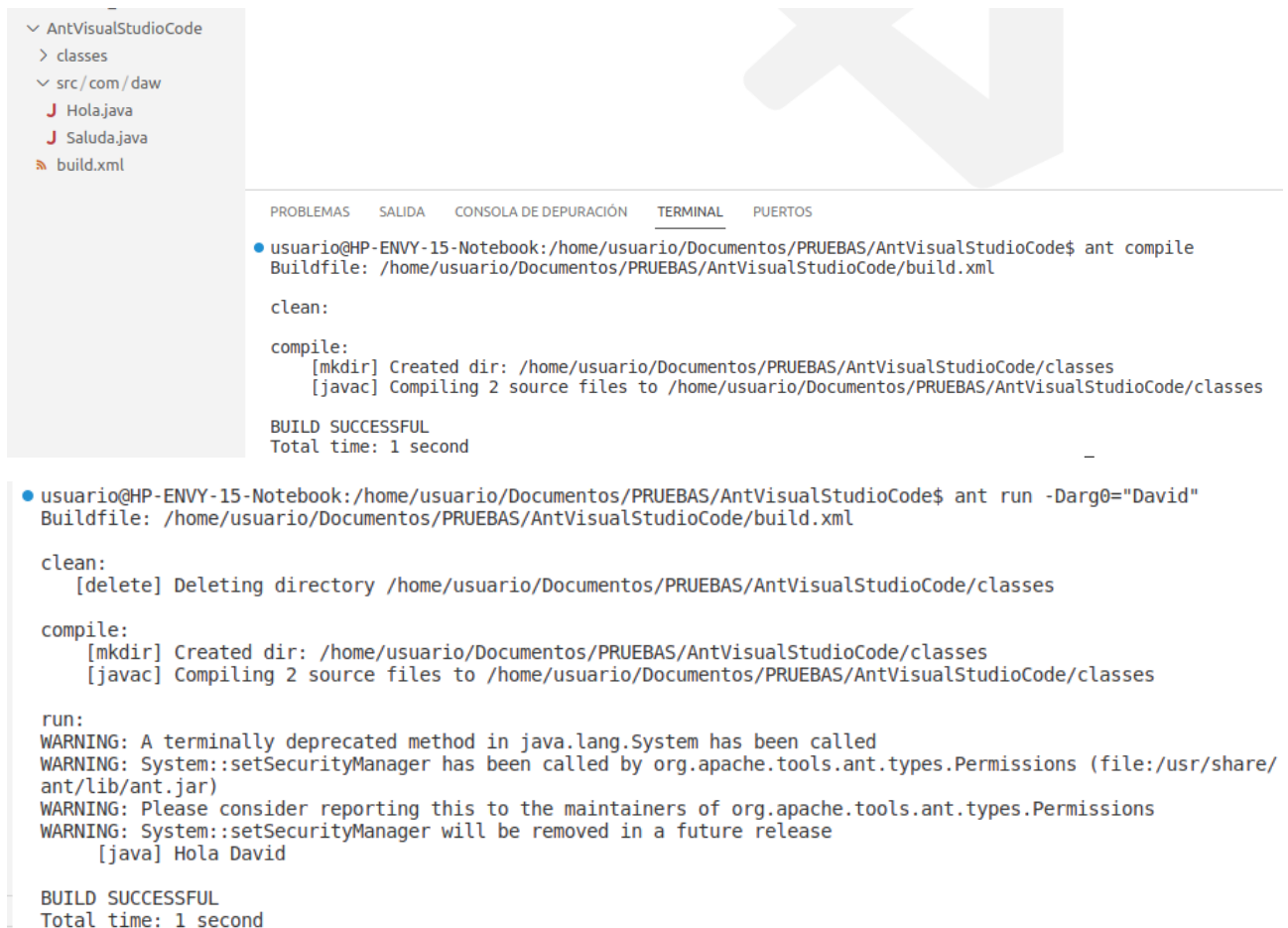
```

VISUAL STUDIO CODE

Crear la carpeta principal del proyecto y dentro de ella las carpetas y los archivos .java y el archivo build.xml



En la terminal escribir los comandos de compilación y ejecución



```
AntVisualStudioCode
├── classes
├── src/com/daw
│   ├── Holajava.java
│   └── Saluda.java
└── build.xml

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

● usuario@HP-ENVY-15-Notebook:/home/usuario/Documentos/PRUEBAS/AntVisualStudioCode$ ant compile
Buildfile: /home/usuario/Documentos/PRUEBAS/AntVisualStudioCode/build.xml

clean:

compile:
[mkdir] Created dir: /home/usuario/Documentos/PRUEBAS/AntVisualStudioCode/classes
[javac] Compiling 2 source files to /home/usuario/Documentos/PRUEBAS/AntVisualStudioCode/classes

BUILD SUCCESSFUL
Total time: 1 second

● usuario@HP-ENVY-15-Notebook:/home/usuario/Documentos/PRUEBAS/AntVisualStudioCode$ ant run -Darg0="David"
Buildfile: /home/usuario/Documentos/PRUEBAS/AntVisualStudioCode/build.xml

clean:
[delete] Deleting directory /home/usuario/Documentos/PRUEBAS/AntVisualStudioCode/classes

compile:
[mkdir] Created dir: /home/usuario/Documentos/PRUEBAS/AntVisualStudioCode/classes
[javac] Compiling 2 source files to /home/usuario/Documentos/PRUEBAS/AntVisualStudioCode/classes

run:
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by org.apache.tools.ant.types.Permissions (file:/usr/share/ant/lib/ant.jar)
WARNING: Please consider reporting this to the maintainers of org.apache.tools.ant.types.Permissions
WARNING: System::setSecurityManager will be removed in a future release
[java] Hola David

BUILD SUCCESSFUL
Total time: 1 second
```

ECLIPSE

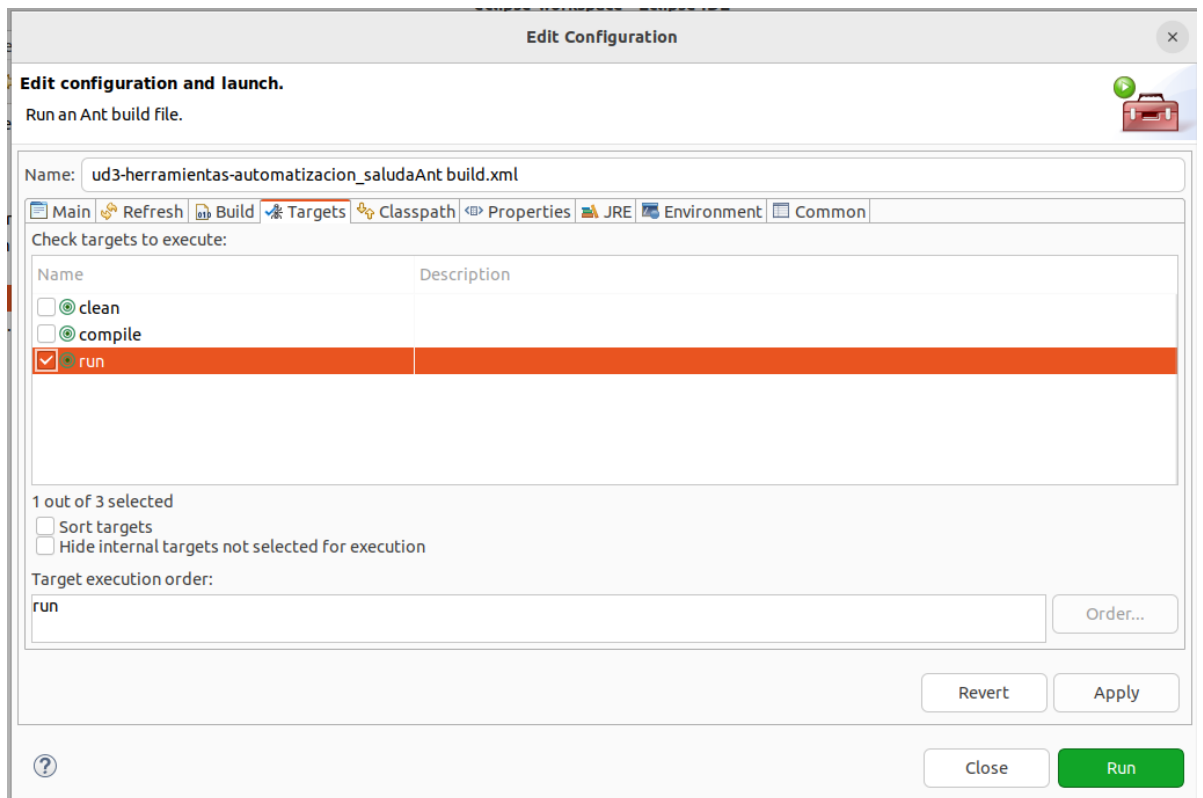
A partir del proyecto **saludaAnt** impórtalo en Eclipse y ejecuta los diferentes targets.

Añade un argumento a la aplicación (por ejemplo, tu nombre) y observa cómo aparece en los resultados.

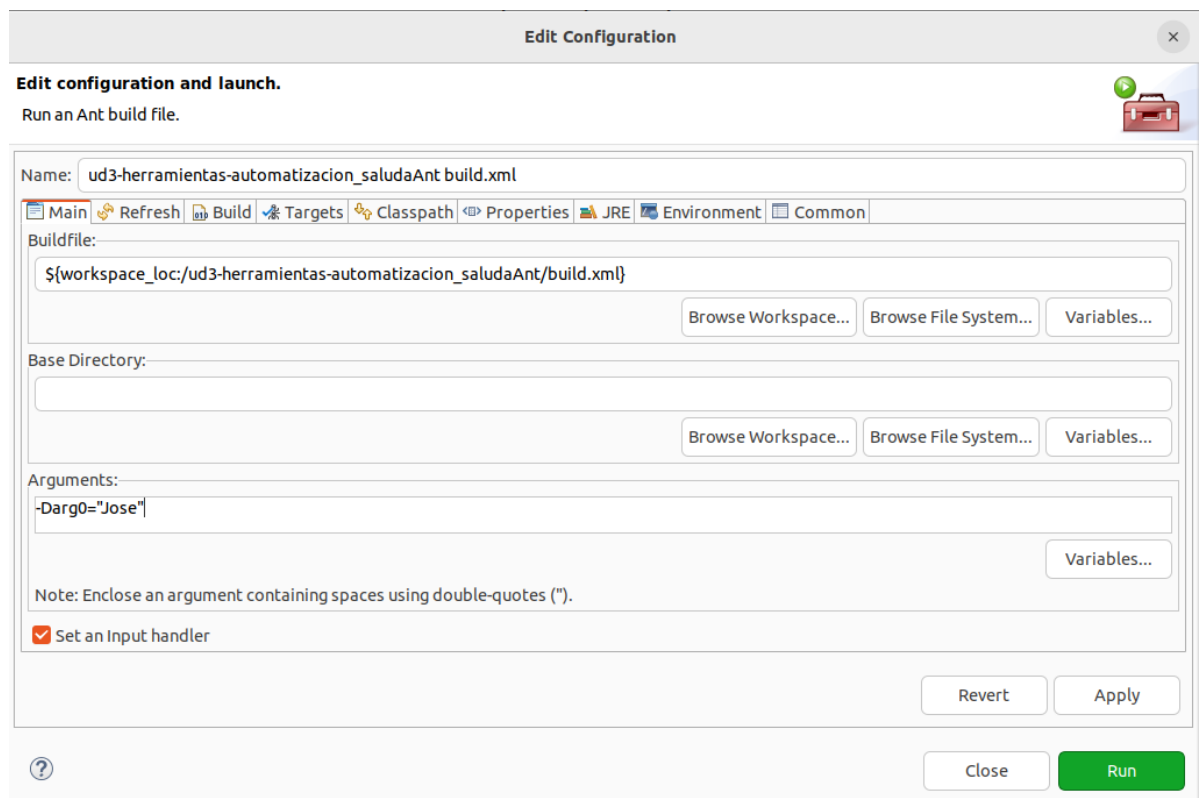
Importar el proyecto **File > Open Projects from File System** seleccionamos la carpeta **saludaAnt**

Para **ejecutar** los diferentes targets:

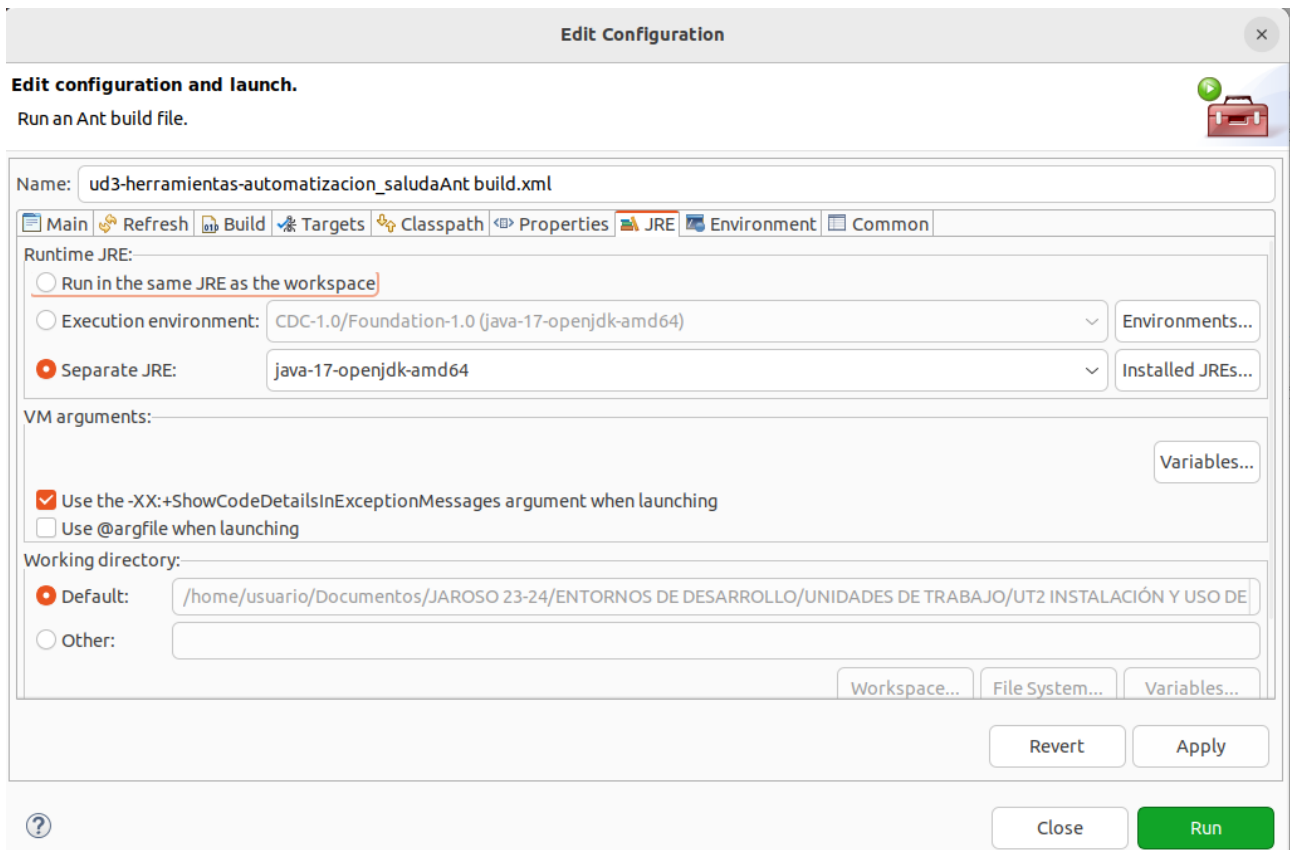
- **clic con el botón derecho** sobre el fichero **build.xml**
- seleccionar **Run As > Ant Build...**
- En la ventana tenemos diferentes pestañas.
 - Las que nos interesan son **Targets**, donde especificamos qué target vamos a lanzar, y **Main**, que nos permite añadir argumentos a la ejecución.



○

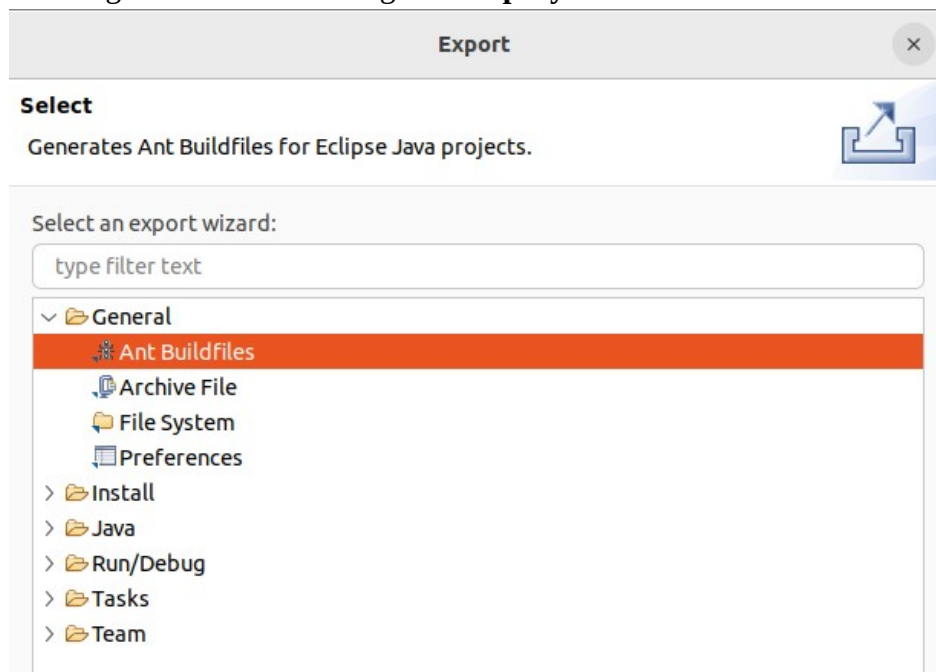


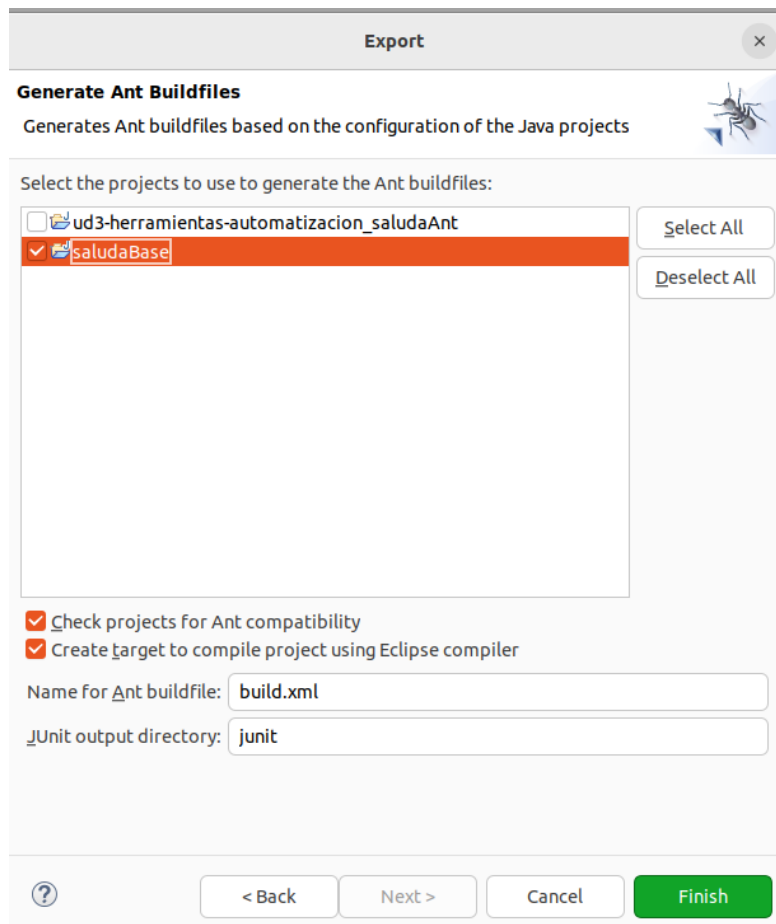
- Si hay problemas en la ejecución con el JRE, elegir un openjdk en Separat JRE.



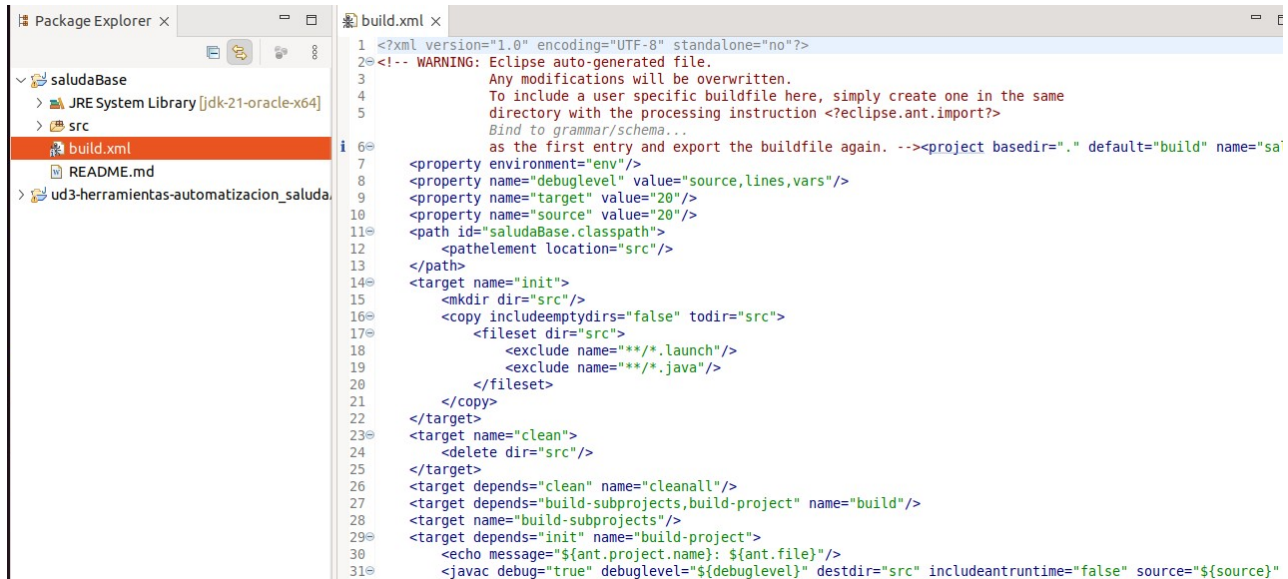
PONTE A PRUEBA: Generación del build.xml Eclipse

- Importamos el proyecto con **File > Open Projects from File System > saludaBase**
- Hacemos clic con el botón derecho sobre el proyecto **saludaBase** (en el árbol de proyectos), y buscamos la opción **Export**.
- Seleccionar el **asistente de exportación**.
- Seleccionar **Ant Build Files** de la sección **General**,
- En la siguiente ventana es cogemos el **proyecto saludaBase**





Si exploramos el fichero veremos que difiere bastante del que habíamos generado manualmente.



Se ha generado a partir de una plantilla con **varios targets más y varias opciones** que no hemos utilizado.

Deberemos tener esto en cuenta cuando nos planteemos generar ficheros de construcción de forma automática, ya que pueden incluir opciones que no utilicemos o incluso que no deseemos.

2. CASO PRÁCTICO : MAVEN

Crea un proyecto MAVEN para que tenga las clases **Saluda.java** y **Hola.java** que tienes disponible en Github (<https://git.io/JcarI>)

TERMINAL

Crear la carpeta del proyecto y movernos a ella

EN MODO INTERACTIVO

Teclear el comando **mvn archetype:generate**

```
usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/MavenTerminal$ mvn archetype:generate
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[WARNING] No archetype found in remote catalog. Defaulting to internal catalog
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.archetypes:maven-archetype-quickstart:1.0)
Choose archetype:
1: internal -> org.apache.maven.archetypes:maven-archetype-archetype (An archetype which contains a sample archetype.)
2: internal -> org.apache.maven.archetypes:maven-archetype-j2ee-simple (An archetype which contains a simplified sample J2EE application.)
3: internal -> org.apache.maven.archetypes:maven-archetype-plugin (An archetype which contains a sample Maven plugin.)
4: internal -> org.apache.maven.archetypes:maven-archetype-plugin-site (An archetype which contains a sample Maven plugin site.
   This archetype can be layered upon an existing Maven plugin project.)
5: internal -> org.apache.maven.archetypes:maven-archetype-portlet (An archetype which contains a sample JSR-268 Portlet.)
6: internal -> org.apache.maven.archetypes:maven-archetype-profiles ()
7: internal -> org.apache.maven.archetypes:maven-archetype-quickstart (An archetype which contains a sample Maven project.)
8: internal -> org.apache.maven.archetypes:maven-archetype-site (An archetype which contains a sample Maven site which demonstrates
   some of the supported document types like APT, XDoc, and FML and demonstrates how
   to i18n your site. This archetype can be layered upon an existing Maven project.)
9: internal -> org.apache.maven.archetypes:maven-archetype-site-simple (An archetype which contains a sample Maven site.)
10: internal -> org.apache.maven.archetypes:maven-archetype-webapp (An archetype which contains a sample Maven Webapp project.)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 7:
```

indicar los valores de configuración:

```
Define value for property 'groupId': com.daw
Define value for property 'artifactId': Saluda4
Define value for property 'version' 1.0-SNAPSHOT: :
Define value for property 'package' com.daw: :
Confirm properties configuration:
groupId: com.daw
artifactId: Saluda4
version: 1.0-SNAPSHOT
package: com.daw
Y: :
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.1
[INFO] -----
[INFO] Parameter: basedir, Value: /home/usuario/Documentos/PRUEBAS/MavenTerminal
[INFO] Parameter: package, Value: com.daw
[INFO] Parameter: groupId, Value: com.daw
[INFO] Parameter: artifactId, Value: Saluda4
[INFO] Parameter: packageName, Value: com.daw
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:19 min
[INFO] Finished at: 2023-11-03T17:42:09+01:00
[INFO] -----
```

Comprobar que se ha generado el fichero pom.xml


```
usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/MavenTerminal$ tree -a
```

```
├── Saluda4
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   │   ├── com
│       │   │   │   ├── daw
│       │   │   │   │   └── App.java
│       │   └── test
│       │       ├── java
│       │       │   ├── com
│       │       │   │   ├── daw
│       │       │   │   │   └── AppTest.java
```

Añadir en el **pom.xml** los valores para los tags para compilar y empaquetar

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.daw</groupId>
  <artifactId>Saluda4</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>Saluda4</name>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Compilar

```
usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.daw:Saluda4 >-----
[INFO] Building Saluda4 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Saluda4 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/src/main/resou
rces
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Saluda4 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/target/classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time:  1.565 s
[INFO] Finished at: 2023-11-03T20:42:20+01:00
[INFO] -----
```

Comprobar las carpetas y ficheros generados

```
usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4$ tree -a
```

```
.
├── pom.xml
├── src
│   ├── main
│   │   ├── java
│   │   │   └── com
│   │   │       └── daw
│   │   │           └── App.java
│   └── test
│       ├── java
│       │   ├── com
│       │   │   └── daw
│       │   │       └── AppTest.java
└── target
    ├── classes
    │   ├── com
    │   │   └── daw
    │   │       └── App.class
    ├── generated-sources
    │   └── annotations
    ├── maven-status
    │   ├── maven-compiler-plugin
    │   │   └── compile
    │   │       └── default-compile
    │   │           ├── createdFiles.lst
    │   │           └── inputFiles.lst
```

19 directories, 6 files

Ejecutar

```
usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4$ java -cp target/classes com.daw.App
Hello World!
```

Empaquetar

```
usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.daw:Saluda4 >-----
[INFO] Building Saluda4 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Saluda4 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Saluda4 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Saluda4 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Saluda4 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Saluda4 ---
[INFO] Surefire report directory: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/target/surefire-reports
```

TESTS

Running com.daw.AppTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.024 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

```
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Saluda4 ---
[INFO] Building jar: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/target/Saluda4-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 2.398 s
[INFO] Finished at: 2023-11-03T20:50:49+01:00
[INFO]
```

Comprobar los ficheros

```
usuario@HP-ENVY-15-Notebook: /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4$ tree -a
```

```
.
├── pom.xml
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── com
│   │   │   │   ├── daw
│   │   │   │   │   └── App.java
│   │   └── test
│   │       ├── java
│   │       │   ├── com
│   │       │   │   ├── daw
│   │       │   │   │   └── AppTest.java
│   └── target
│       ├── classes
│       │   ├── com
│       │   │   └── daw
│       │   │       └── App.class
│       ├── generated-sources
│       │   └── annotations
│       ├── generated-test-sources
│       │   └── test-annotations
│       ├── maven-archiver
│       │   └── pom.properties
│       ├── maven-status
│       │   └── maven-compiler-plugin
│       │       ├── compile
│       │       │   ├── default-compile
│       │       │   │   ├── createdFiles.lst
│       │       │   │   └── inputFiles.lst
│       │       └── testCompile
│       │           ├── default-testCompile
│       │           │   ├── createdFiles.lst
│       │           │   └── inputFiles.lst
│       ├── Saluda4-1.0-SNAPSHOT.jar
│       ├── surefire-reports
│       │   ├── com.daw.AppTest.txt
│       │   └── TEST-com.daw.AppTest.xml
│       └── test-classes
│           ├── com
│           │   └── daw
│           │       └── AppTest.class
```

28 directories, 13 files

Limpiar y comprobar que el proyecto sólo tiene los ficheros java

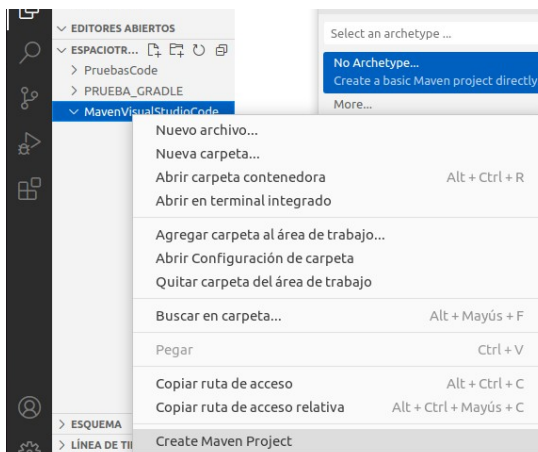
```

usuario@HP-ENVY-15-Notebook:/home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.daw:Saluda4 >-----
[INFO] Building Saluda4 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ Saluda4 ---
[INFO] Deleting /home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4/target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 0.313 s
[INFO] Finished at: 2023-11-03T21:00:21+01:00
[INFO]
[INFO] -----
usuario@HP-ENVY-15-Notebook:/home/usuario/Documentos/PRUEBAS/MavenTerminal/Saluda4$ tree -a
.
├── pom.xml
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── com
│   │   │   │   ├── daw
│   │   │   │   │   └── App.java
│   │   └── test
│   │       ├── java
│   │       │   ├── com
│   │       │   │   ├── daw
│   │       │   │   │   └── AppTest.java
└── 9 directories, 3 files

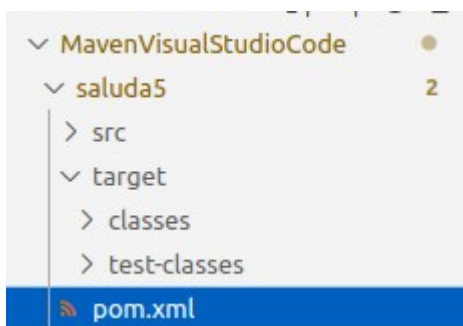
```

VISUALSTUDIO CODE

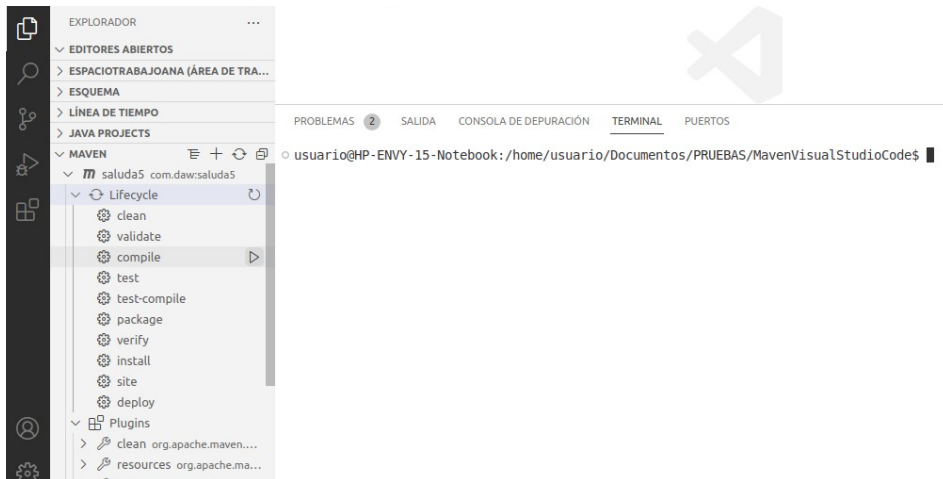
Clic derecho sobre la carpeta, **Create Maven Project** y seguir los pasos del asistente



Comprobar los ficheros en el explorador de proyectos



Abrir la **pestaña MAVEN** y **ejecutar los objetivos** del ciclo de vida: compile, package...



Ejecutar: Menú ejecutar

Hello World!

ECLIPSE

Nuevo proyecto Maven, seleccionar **arquétipo maven-archetype-quickstart**, indicar los parámetros

A screenshot of the 'New Maven project' dialog box in Eclipse. The dialog has a title bar 'New Maven project' and a subtitle 'Specify Archetype parameters'. It contains several input fields: 'Group Id:' with the value 'com.daw', 'Artifact Id:' with the value 'saluda6', 'Version:' with a dropdown menu showing '0.0.1-SNAPSHOT', and 'Package:' with the value 'com.daw.saluda6'. Below these fields, there is a checkbox labeled 'run archetype generation interactively' which is checked. At the bottom, there is a section titled 'Properties available from archetype:'.

Se generan todos los archivos del proyecto:

The Package Explorer on the left shows the project structure: PRUEBAS ANT, Saluda4, saluda6 (with subfolders src/main/java, src/test/java, JRE System Library [JavaSE-1.7], Maven Dependencies, src, target), and pom.xml (selected).

The Console on the right shows the Maven build output for pom.xml:

```

<terminated> /usr/lib/jvm/jdk-21-oracle-x64/bin/java (3 nov 2023 21:53:50) [pid: 21
package: com.daw.saluda6
Y: :
[INFO] -----
[INFO] Using following parameters for creating project from Archetype
[INFO] -----
[INFO] Parameter: groupId, Value: com.daw
[INFO] Parameter: artifactId, Value: saluda6
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: com.daw.saluda6
[INFO] Parameter: packageInPathFormat, Value: com/daw/saluda6
[INFO] Parameter: package, Value: com.daw.saluda6
[INFO] Parameter: groupId, Value: com.daw
[INFO] Parameter: artifactId, Value: saluda6
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Project created from Archetype in dir: /home/usuario/eclipse-
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:50 min
[INFO] Finished at: 2023-11-03T21:55:43+01:00
[INFO] -----

```

Compilar: clic derecho en **pom.xml**, **Run as, maven build, compile**

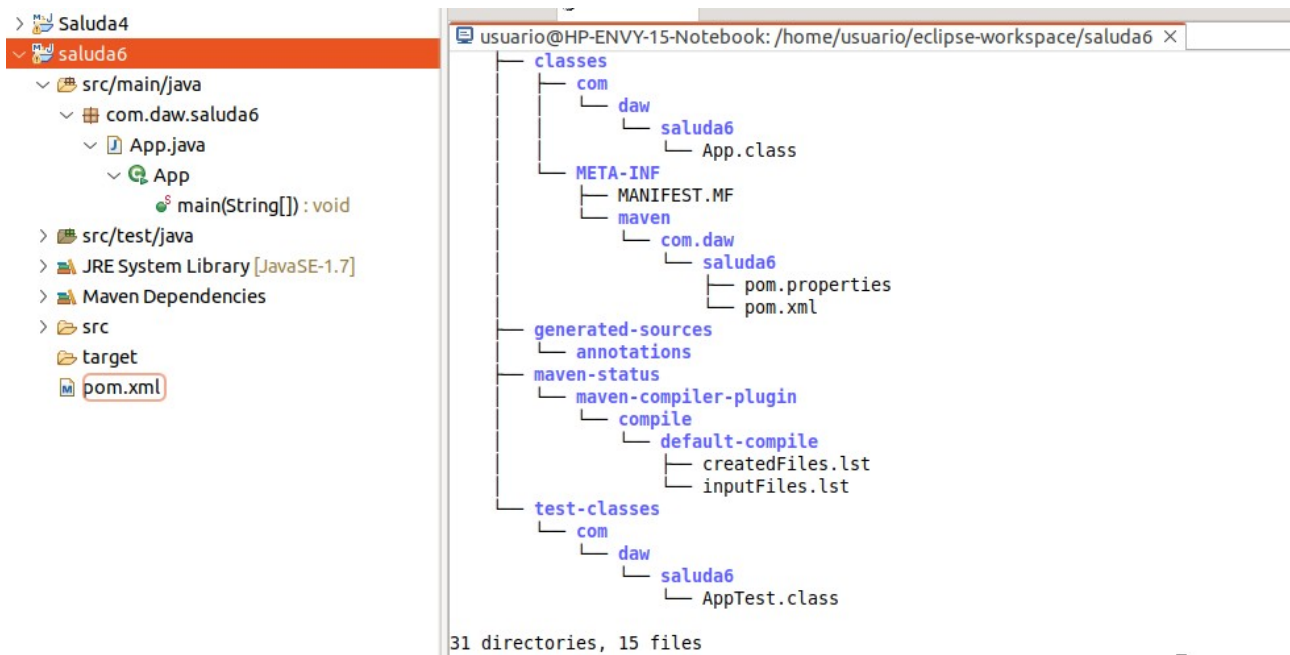


Edit configuration and launch.

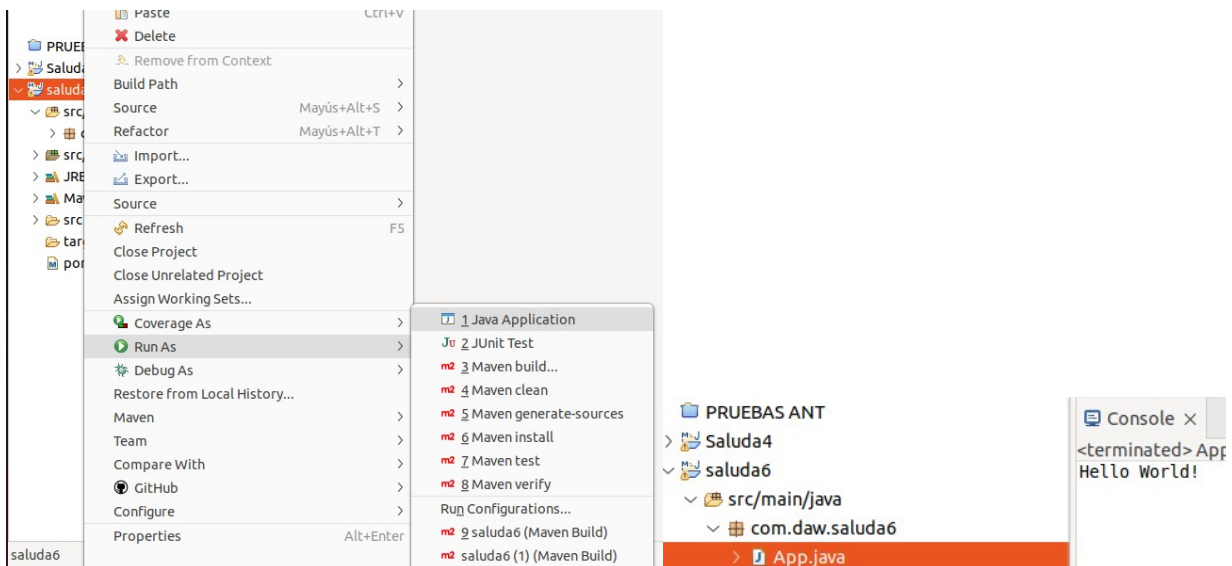
The 'Main' tab is selected in the 'Edit Configuration' dialog box. The configuration details are as follows:

- Name: saluda6 (1)
- Base directory: `${project_loc:saluda6}`
- Goals: `compile`
- Profiles: (empty)
- User settings: `/home/usuario/.m2/settings.xml`

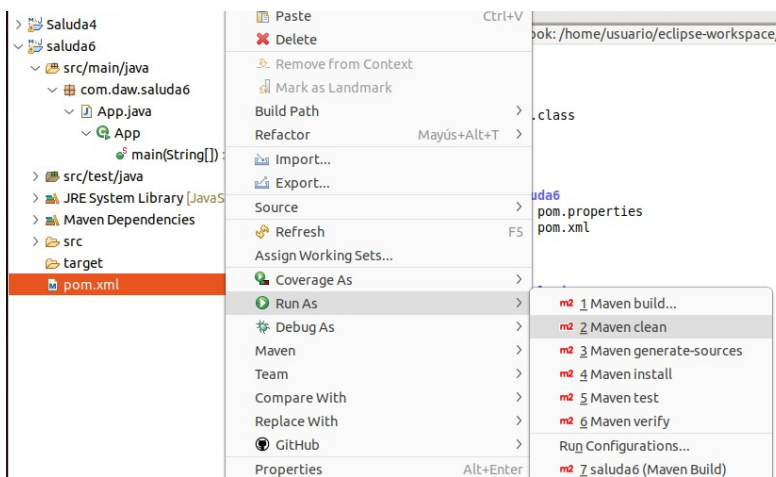
Comprobar ficheros: `tree -a`



Ejecutar: Run as Java Application

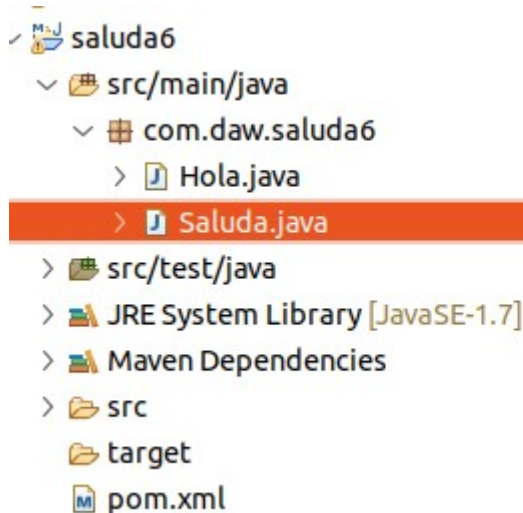


Limpiar: Run as, Maven clean

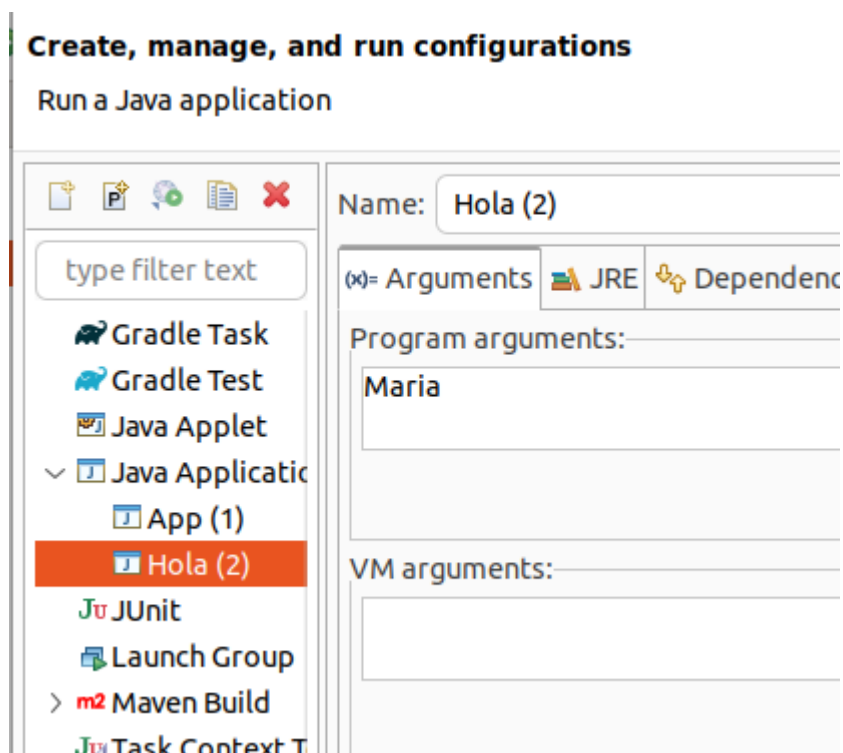


Una vez limpio el proyecto y comprobado que todo funciona correctamente, se **elimina el fichero App.java** y se añaden los ficheros **Hola.java** y **Saluda.java**(pegar en la carpeta del proyecto)

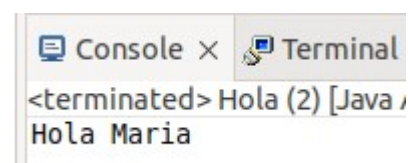
Corregir los posibles errores, compilar y ejecutar



Indicar los parámetros en Run as, run Configurations



Resultado:



3. CASO PRÁCTICO: GRADLE

TERMINAL

Crea un nuevo proyecto Gradle utilizando los ficheros proporcionados en el ejemplo saludaBase (*Hola.java* y *Saluda.java*), que tienes disponible en Github (<https://git.io/Jcarl>)

gradle init, gradle build, gradle run

SOLUCIÓN

Para crear el proyecto, lo haremos mediante gradle init y seguiremos el asistente:



```
Starting a Gradle Daemon, 2 incompatible and 1 stopped Daemons could not be reused,  
use --status for details
```

```
Select type of project to generate:
```

- 1: basic
- 2: application
- 3: library
- 4: Gradle plugin



```
Select implementation language:
```

- 1: C++
- 2: Groovy
- 3: Java
- 4: Kotlin
- 5: Scala
- 6: Swift



```
Split functionality across multiple subprojects?:
```

- 1: no - only one application project
- 2: yes - application and library projects



```
Select build script DSL:
```

- 1: Groovy
- 2: Kotlin



```
Select test framework:
```

```
1: JUnit 4
2: TestNG
3: Spock
4: JUnit Jupiter
```



```
> Task :init
...
```

Con esto ya tenemos el proyecto creado. Ahora vamos a copiar en él los fuentes de SaludaBase, y a eliminar el fichero App.java creado por defecto:

```
$ cp ../saludaBase/src/com/mgh/edd/*.java app/src/main/java/com/mgh/edd/
$ rm app/src/main/java/com/mgh/edd/App.java
```

Como el método main ha cambiado de ubicación (estaba en App.java y ahora está en Hola.java), deberemos modificar el valor de mainClass, dentro de application del fichero app/build.gradle:

```
application {
    // Define the main class for the application.
    mainClass = 'com.mgh.edd.Hola'
}
```

Finalmente, si queremos construirlo o ejecutarlo, lo haríamos mediante gradle build y gradle run respectivamente, o bien utilizando el wrapper de Gradle ./gradlew build y ./gradlew run.



```
> Task :app:run
Hola
...
```

Añadiendo argumentos

Si intentamos incorporar argumentos a la línea de ejecución del gradle run, veremos que nos lanza un error, ya que el comando que estamos ejecutando es gradle, y no java. Para pasar los argumentos a nuestra aplicación java, a través de Gradle deberemos indicarlos con el argumento —args de Gradle:



```
> Task :app:run
Hola Jose
...
```

VISUAL STUDIO CODE

Añadir una carpeta al workspace

En terminal Movernos a la carpeta

o **usuario@HP-ENVY-15-Notebook:~/Documentos/PRUEBAS/gradleVSC\$** gradle init

Select type of project to generate:

- 1: basic
- 2: application
- 3: library
- 4: Gradle plugin

Enter selection (default: basic) [1..4] 2

Select implementation language:

- 1: C++
- 2: Groovy
- 3: Java
- 4: Kotlin
- 5: Scala
- 6: Swift

Enter selection (default: Java) [1..6] 3

Generate multiple subprojects for application? (default: no) [yes, no]

Select build script DSL:

- 1: Kotlin
- 2: Groovy

Enter selection (default: Kotlin) [1..2] 2

Select test framework:

- 1: JUnit 4
- 2: TestNG
- 3: Spock
- 4: JUnit Jupiter

Enter selection (default: JUnit Jupiter) [1..4] 2

Project name (default: gradleVSC): saluda10

Source package (default: saluda10): saluda10



> Task :init

To learn more about Gradle by exploring our Samples at <https://docs.gradle.org/5.6.2/samples/>

BUILD SUCCESSFUL in 2m 56s

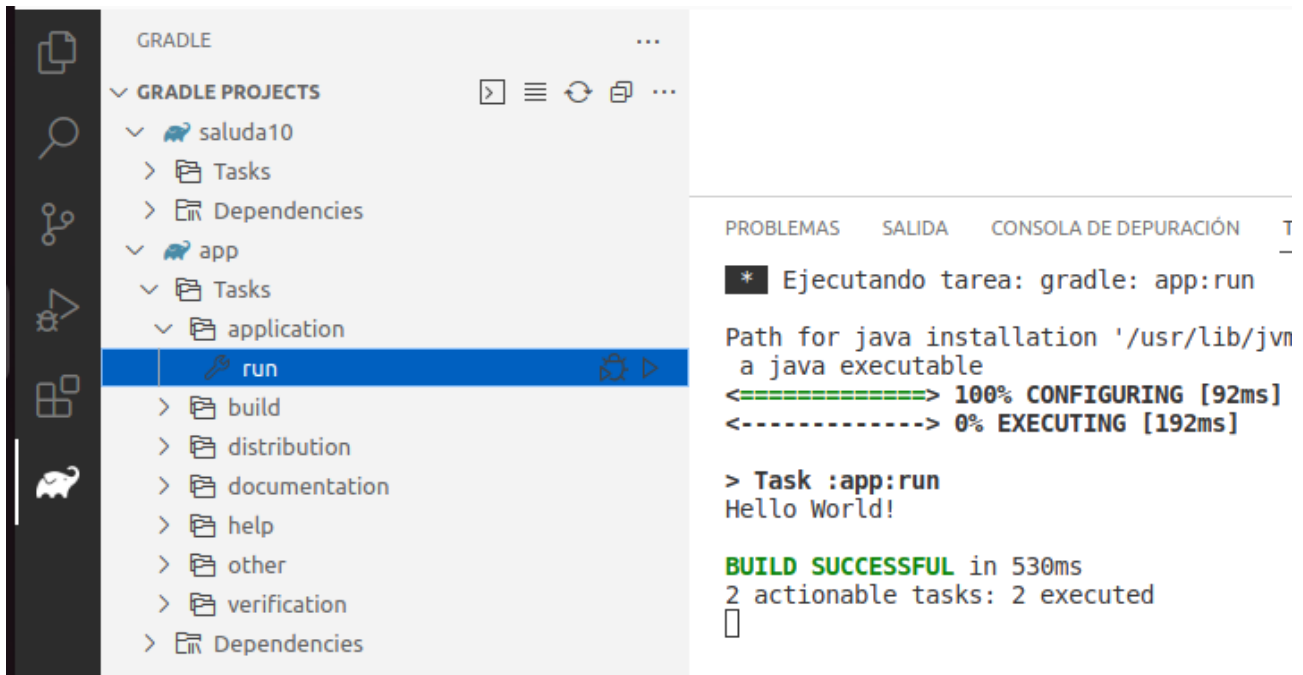
2 actionable tasks: 2 executed

● **usuario@HP-ENVY-15-Notebook:~/Documentos/PRUEBAS/gradleVSC\$** tree -a

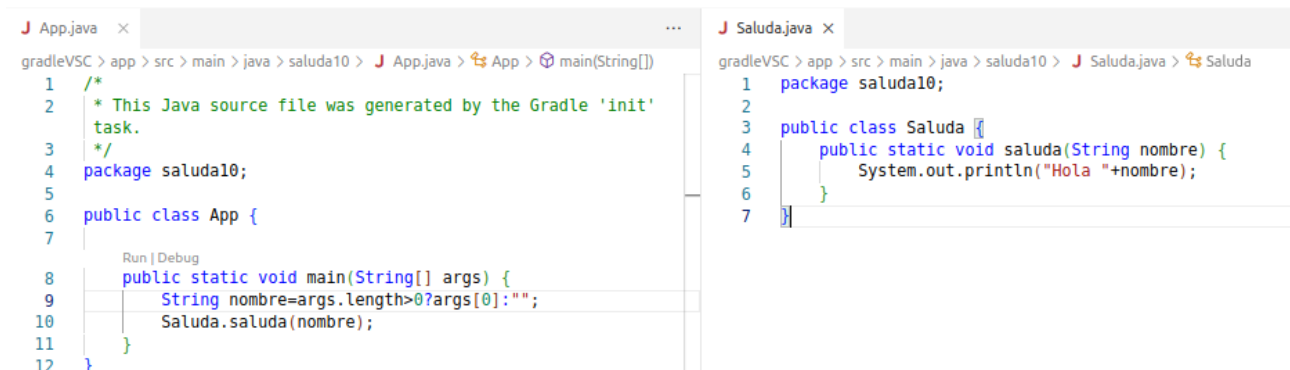
```
.
├── app
│   ├── build.gradle
│   └── src
│       ├── main
│       │   ├── java
│       │   │   └── saluda10
│       │   │       └── App.java
│       │   └── resources
│       └── test
│           ├── java
│           │   └── saluda10
│           │       └── AppTest.java
│           └── resources
├── .gitattributes
├── .gitignore
├── .gradle
├── file-system.probe
├── gradle
│   └── wrapper
│       ├── gradle-wrapper.jar
│       └── gradle-wrapper.properties
├── gradlew
├── gradlew.bat
└── settings.gradle
```

13 directories, 11 files

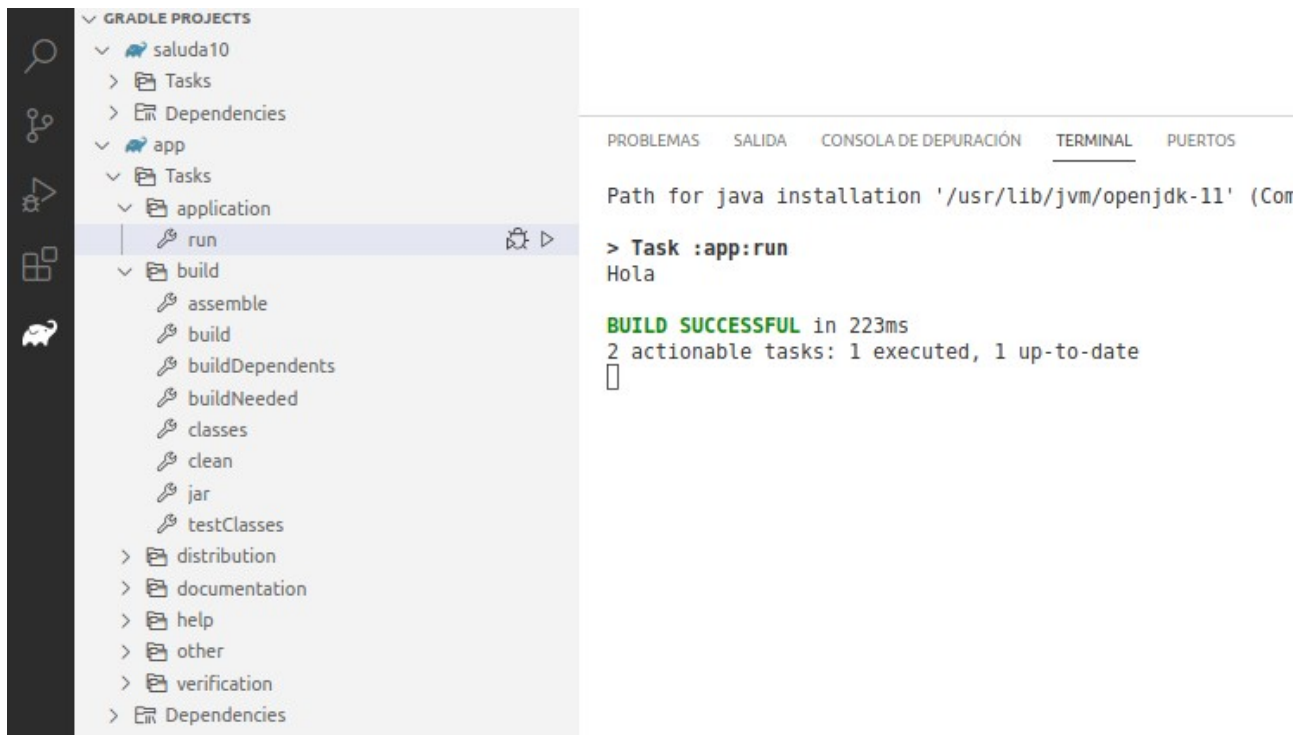
Abrir la pestaña de Gradle y ejecutar las task de la aplicación



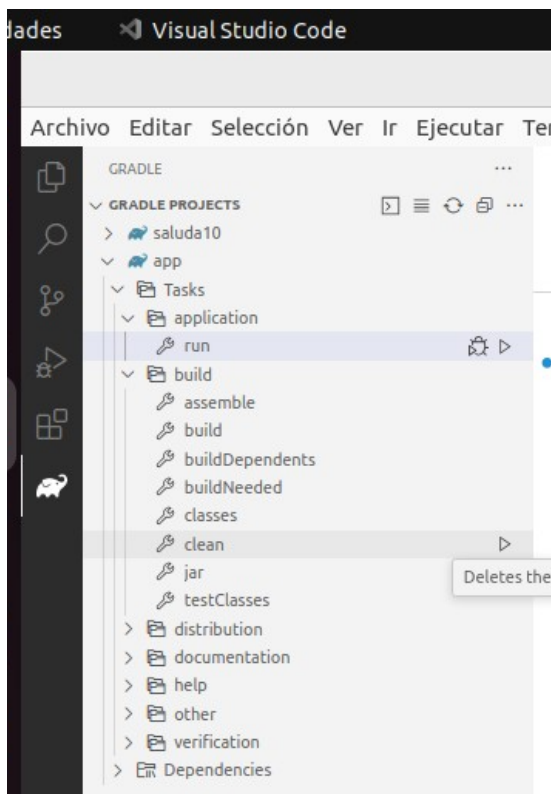
Modificar los ficheros por Hola.java y Saluda.java



Ejecutar



Realizar task: clean, build, jar



Comprobar los ficheros generados

• usuario@HP-ENVY-15-Notebook:~/Documentos/PRUEBAS/gradleVSC\$ tree -a

```
.
├── app
│   ├── build
│   │   ├── classes
│   │   │   ├── java
│   │   │   │   ├── main
│   │   │   │   │   ├── saluda10
│   │   │   │   │   │   ├── App.class
│   │   │   │   │   │   └── Saluda.class
│   │   ├── generated
│   │   │   ├── sources
│   │   │   │   ├── annotationProcessor
│   │   │   │   │   ├── java
│   │   │   │   │   │   ├── main
│   │   │   │   ├── headers
│   │   │   │   │   ├── java
│   │   │   │   │   │   ├── main
│   │   ├── tmp
│   │   │   ├── compileJava
│   │   │   │   ├── previous-compilation-data.bin
│   ├── build.gradle
│   ├── src
│   │   ├── main
│   │   │   ├── java
│   │   │   │   ├── saluda10
│   │   │   │   │   ├── App.java
│   │   │   │   │   └── Saluda.java
│   │   │   └── resources
│   │   └── test
│   │       ├── java
│   │       │   ├── saluda10
│   │       └── resources
│   └── .gitattributes
├── .gitignore
├── .gradle
│   └── 8.4
```


ECLIPSE

```
usuario@HP-ENVY-15-Notebook: ~/eclipse-workspace/saluda9 ×
usuario@HP-ENVY-15-Notebook:~/eclipse-workspace$ mkdir saluda9
usuario@HP-ENVY-15-Notebook:~/eclipse-workspace$ cd saluda9
usuario@HP-ENVY-15-Notebook:~/eclipse-workspace/saluda9$ gradle init

Select type of project to generate:
  1: basic
  2: application
  3: library
  4: Gradle plugin
Enter selection (default: basic) [1..4] 2

Select implementation language:
  1: C++
  2: Groovy
  3: Java
  4: Kotlin
  5: Scala
  6: Swift
Enter selection (default: Java) [1..6] 3

Generate multiple subprojects for application? (default: no) [yes, no]
```

Generate multiple subprojects for application? (default: no) [yes, no]

Select build script DSL:

- 1: Kotlin
- 2: Groovy

Enter selection (default: Kotlin) [1..2] 2

Select test framework:

- 1: JUnit 4
- 2: TestNG
- 3: Spock
- 4: JUnit Jupiter

Enter selection (default: JUnit Jupiter) [1..4] 1

Project name (default: saluda9):

Source package (default: saluda9):

Enter target version of Java (min. 7) (default: 17):

Generate build using new APIs and behavior (some features may change in the next minor release)? (default: no) [yes, no]
] n
o

BUILD SUCCESSFUL in 51s

2 actionable tasks: 2 executed

usuario@HP-ENVY-15-Notebook:~/eclipse-workspace/saluda9\$ tree -a

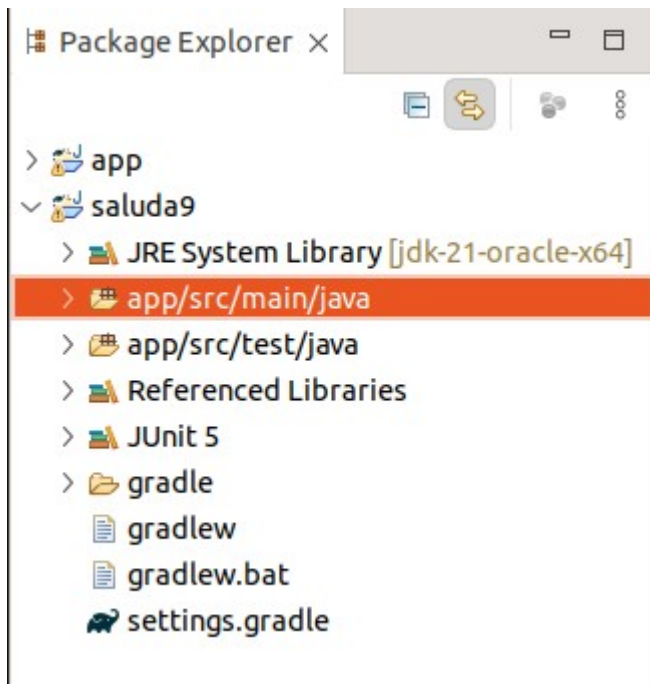
```
.
├── app
│   ├── build.gradle
│   └── src
│       ├── main
│       │   ├── java
│       │   │   └── saluda9
│       │   │       └── App.java
│       │   └── resources
│       └── test
│           ├── java
│           │   └── saluda9
│           │       └── AppTest.java
│           └── resources
├── .gitattributes
├── .gitignore
├── .gradle
│   └── file-system.probe
├── gradle
│   └── wrapper
│       ├── gradle-wrapper.jar
│       └── gradle-wrapper.properties
├── gradlew
├── gradlew.bat
└── settings.gradle
```

13 directories, 11 files

Modificar el código de App.java por Hola.java y añadir Saluda.java



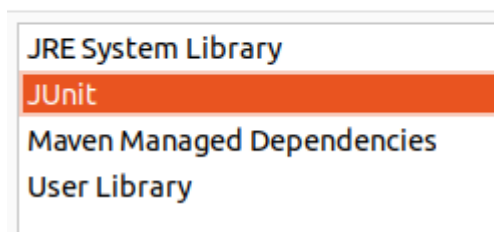
Importar proyecto desde Sistema de ficheros



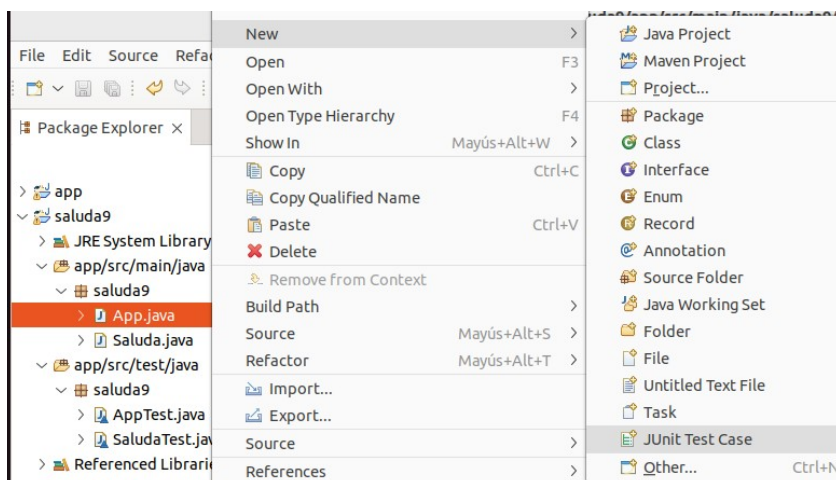
Si es necesario añadir la librería de JUnit

Add Library

Select the library type to add.



Crear los ficheros de test en app/src/test/java: clic derecho sobre los archivos java y new, case junit



Ejecutar

Console Terminal Gradle Tasks ×	
Name	Description
▼ saluda9	
▼ app	
▼ application	
run	Runs this project

Comprobar los resultados

```

app -run (1) [Gradle Project] run in /home/usuario/eclip
Working Directory: /home/usuario/eclipse-workspace
Gradle user home: /home/usuario/.gradle
Gradle Distribution: Local installation at /opt/gradle
Gradle Version: 8.4
Java Home: /usr/lib/jvm/java-17-openjdk-amd64
JVM Arguments: None
Program Arguments: None
Build Scans Enabled: false
Offline Mode Enabled: false
Gradle Tasks: run

Path for java installation '/usr/lib/jvm/openjdk-17'
> Task :app:compileJava UP-TO-DATE
> Task :app:processResources NO-SOURCE
> Task :app:classes UP-TO-DATE

> Task :app:run
Hola

BUILD SUCCESSFUL in 235ms
2 actionable tasks: 1 executed, 1 up-to-date

```

Comprobar ficheros generados: `tree -a`

```
|
|  └─ buildOutputCleanup
|      └─ buildOutputCleanup.lock
|      └─ cache.properties
|      └─ outputFiles.bin
|  └─ file-system.probe
|  └─ vcs-1
|      └─ gc.properties
└─ gradle
    └─ wrapper
        └─ gradle-wrapper.jar
        └─ gradle-wrapper.properties
└─ gradlew
└─ gradlew.bat
└─ .project
└─ .settings
    └─ org.eclipse.buildship.core.prefs
└─ settings.gradle
```

46 directories, 45 files