

Entrega Tarea UD02.5. Herramientas de desarrollo

David Mata Martínez

1. Ejecuta el programa "Hola mundo" en los siguientes lenguajes:

- bash
- python
- php
- javascript (nodejs)
- c
- c++
- java
- ruby
- go
- rust
- lisp
- ensamblador (nasm)

Crear archivo de texto y ejecutar BASH

The screenshot shows a terminal window titled "david@david-VirtualBox: ~/Escritorio/Codigos". The command history shows:

```
david@david-VirtualBox:~/Escritorio/Codigos$ chmod +x hola.sh
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola.sh
Hola mundo
david@david-VirtualBox:~/Escritorio/Codigos$
```

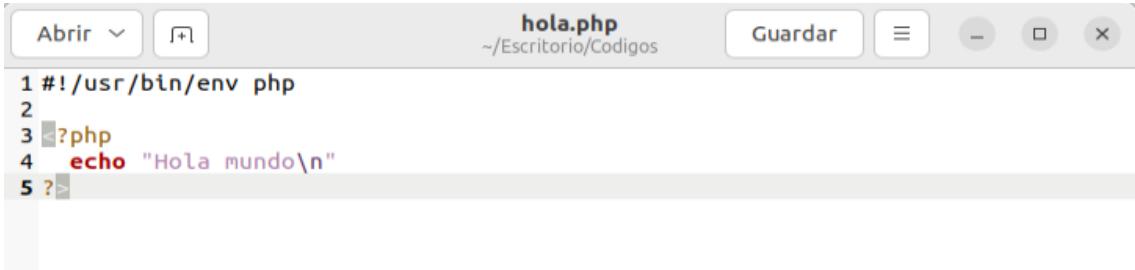
Crear archivo de texto y ejecutar PYTHON

The screenshot shows a terminal window titled "david@david-VirtualBox: ~/Escritorio/Codigos". The command history shows:

```
david@david-VirtualBox:~/Escritorio/Codigos$ python3
3 print ("Hola mundo")
```

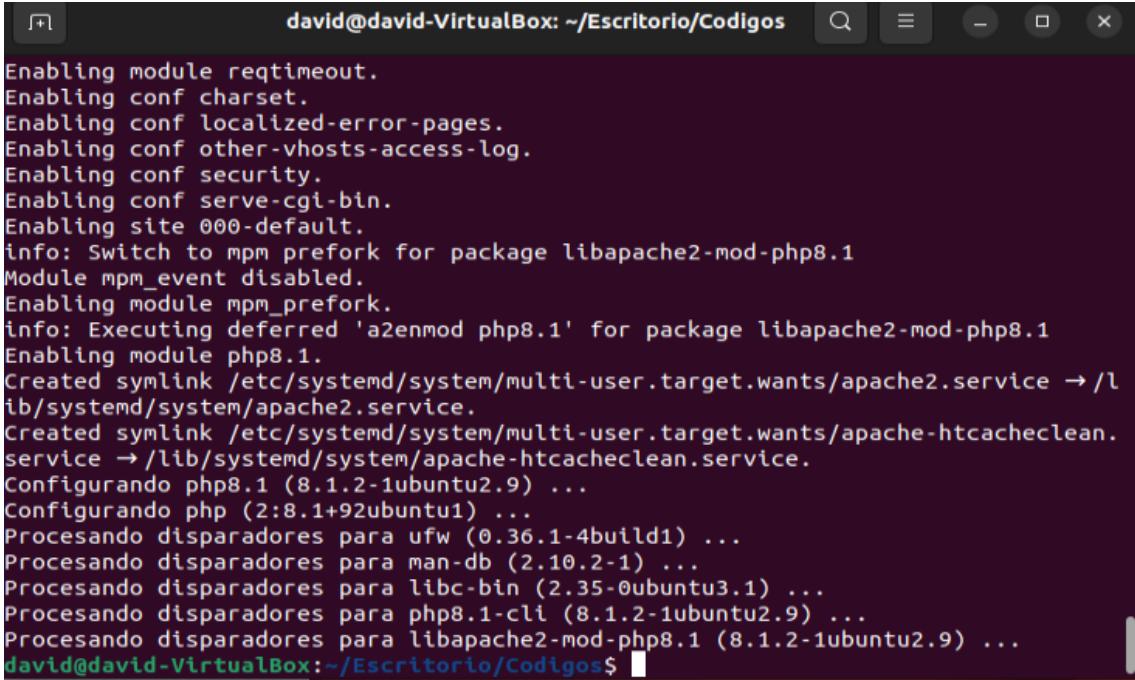
```
david@david-VirtualBox:~/Escritorio/Codigos$ gedit hola.py
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola.py
Hola mundo
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Crear archivo de texto PHP



```
Abrir ▾ Guardar
hola.php
~/Escritorio/Codigos
1 #!/usr/bin/env php
2
3 <?php
4 echo "Hola mundo\n"
5 ?>
```

⊕ Hacemos la instalación (todas las instalaciones serán sudo apt-get install <nombre del paquete>)

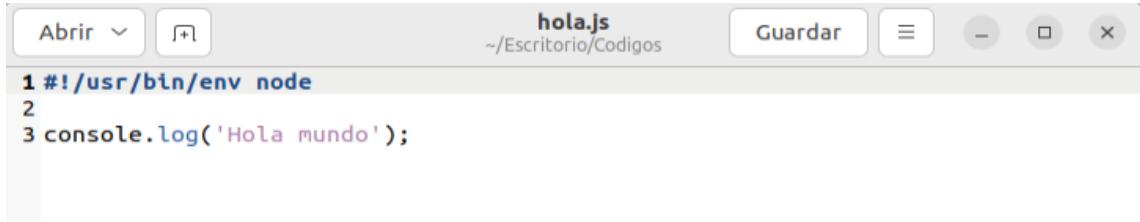


```
david@david-VirtualBox: ~/Escritorio/Codigos
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
info: Switch to mpm prefork for package libapache2-mod-php8.1
Module mpm_event disabled.
Enabling module mpm_prefork.
info: Executing deferred 'a2enmod php8.1' for package libapache2-mod-php8.1
Enabling module php8.1.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Configurando php8.1 (8.1.2-1ubuntu2.9) ...
Configurando php (2:8.1+92ubuntu1) ...
Procesando disparadores para ufw (0.36.1-4build1) ...
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.1) ...
Procesando disparadores para php8.1-cli (8.1.2-1ubuntu2.9) ...
Procesando disparadores para libapache2-mod-php8.1 (8.1.2-1ubuntu2.9) ...
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Ejecutamos

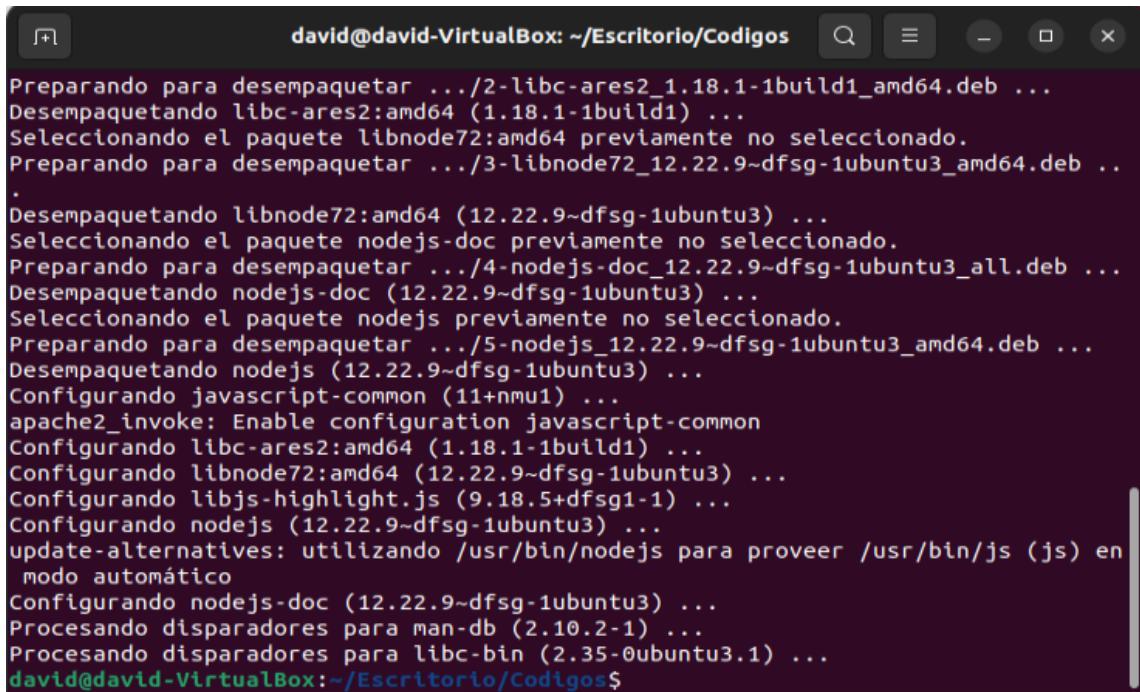
```
david@david-VirtualBox:~/Escritorio/Codigos$ chmod +x hola.php
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola.php
Hola mundo
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Javascript (nodejs) archivo de texto



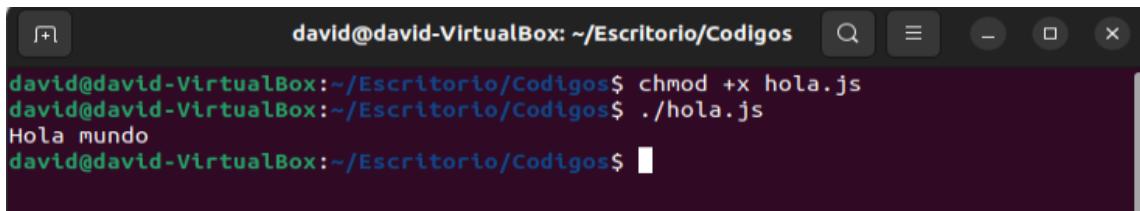
```
1 #!/usr/bin/env node
2
3 console.log('Hola mundo');
```

✍ Instalación



```
Preparando para desempaquetar .../2-libc-ares2_1.18.1-1build1_amd64.deb ...
Desempaquetando libc-ares2:amd64 (1.18.1-1build1) ...
Seleccionando el paquete libnode72:amd64 previamente no seleccionado.
Preparando para desempaquetar .../3-libnode72_12.22.9~dfsg-1ubuntu3_amd64.deb ...
.
Desempaquetando libnode72:amd64 (12.22.9~dfsg-1ubuntu3) ...
Seleccionando el paquete nodejs-doc previamente no seleccionado.
Preparando para desempaquetar .../4-nodejs-doc_12.22.9~dfsg-1ubuntu3_all.deb ...
Desempaquetando nodejs-doc (12.22.9~dfsg-1ubuntu3) ...
Seleccionando el paquete nodejs previamente no seleccionado.
Preparando para desempaquetar .../5-nodejs_12.22.9~dfsg-1ubuntu3_amd64.deb ...
Desempaquetando nodejs (12.22.9~dfsg-1ubuntu3) ...
Configurando javascript-common (11+nmu1) ...
apache2_invoke: Enable configuration javascript-common
Configurando libc-ares2:amd64 (1.18.1-1build1) ...
Configurando libnode72:amd64 (12.22.9~dfsg-1ubuntu3) ...
Configurando libjs-highlight.js (9.18.5+dfsg1-1) ...
Configurando nodejs (12.22.9~dfsg-1ubuntu3) ...
update-alternatives: utilizando /usr/bin/nodejs para proveer /usr/bin/js (js) en
 modo automático
Configurando nodejs-doc (12.22.9~dfsg-1ubuntu3) ...
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.1) ...
david@david-VirtualBox:~/Escritorio/Codigos$
```

✍ Ejecutamos



```
david@david-VirtualBox:~/Escritorio/Codigos$ chmod +x hola.js
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola.js
Hola mundo
david@david-VirtualBox:~/Escritorio/Codigos$
```

✍ Archivo de texto en C

A screenshot of a code editor window titled "hola.c" located at "/Escritorio/Codigos". The window has standard OS X-style controls at the top. The code editor displays the following C code:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("¡Hola, mundo!");
6     return 0;
7 }
```

⊕ Instalamos

A screenshot of a terminal window titled "david@david-VirtualBox: ~/Escritorio/Codigos". The window shows the output of a series of package configurations, likely for a build system like CMake or a package manager like apt. The output includes many "Configurando" messages followed by package names and version numbers. The terminal prompt "david@david-VirtualBox:~/Escritorio/Codigos\$" is visible at the bottom.

```
Configurando rpcsvc-proto (1.4.2-0ubuntu6) ...
Configurando libquadmath0:amd64 (12.1.0-2ubuntu1~22.04) ...
Configurando libatomic1:amd64 (12.1.0-2ubuntu1~22.04) ...
Configurando libubsan1:amd64 (12.1.0-2ubuntu1~22.04) ...
Configurando libnsl-dev:amd64 (1.3.0-2build2) ...
Configurando libcrypt-dev:amd64 (1:4.4.27-1) ...
Configurando libbinutils:amd64 (2.38-4ubuntu2.1) ...
Configurando libc-dev-bin (2.35-0ubuntu3.1) ...
Configurando libcc1-0:amd64 (12.1.0-2ubuntu1~22.04) ...
Configurando liblsan0:amd64 (12.1.0-2ubuntu1~22.04) ...
Configurando libitm1:amd64 (12.1.0-2ubuntu1~22.04) ...
Configurando libc-devtools (2.35-0ubuntu3.1) ...
Configurando libtsan0:amd64 (11.3.0-1ubuntu1~22.04) ...
Configurando libctf0:amd64 (2.38-4ubuntu2.1) ...
Configurando cpp-11 (11.3.0-1ubuntu1~22.04) ...
Configurando libgcc-11-dev:amd64 (11.3.0-1ubuntu1~22.04) ...
Configurando libc6-dev:amd64 (2.35-0ubuntu3.1) ...
Configurando binutils-x86-64-linux-gnu (2.38-4ubuntu2.1) ...
Configurando binutils (2.38-4ubuntu2.1) ...
Configurando gcc-11 (11.3.0-1ubuntu1~22.04) ...
Configurando gcc (4:11.2.0-1ubuntu1) ...
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.1) ...
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Ejecutamos

A screenshot of a terminal window titled "david@david-VirtualBox: ~/Escritorio/Codigos\$". The user runs the command "gcc -o hola hola.c" to compile the C program. Then, they run the executable "./hola", which outputs "¡Hola, mundo!". The terminal prompt "david@david-VirtualBox:~/Escritorio/Codigos\$" is visible at the bottom.

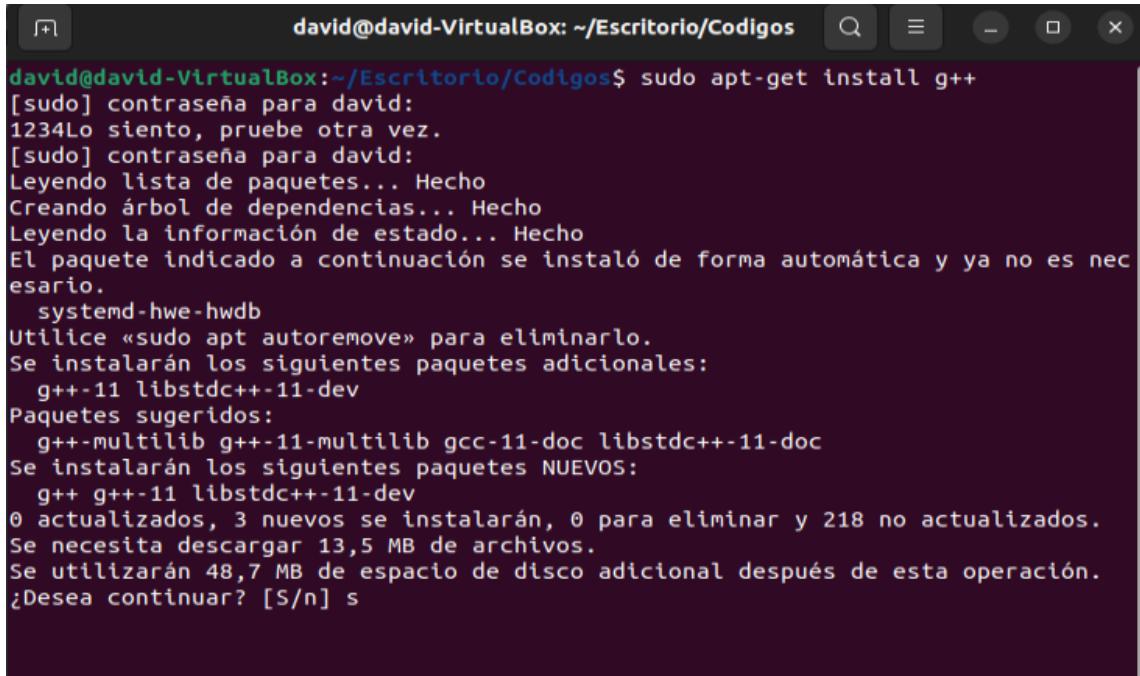
```
Procesando disparadores para libc-bin (2.35-0ubuntu3.1) ...
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -o hola hola.c
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola
¡Hola, mundo!david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Hola mundo en C++



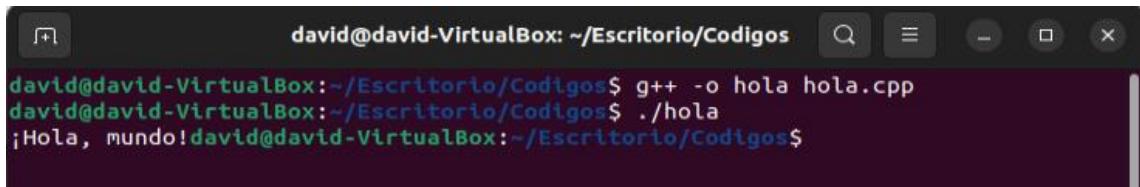
```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "¡Hola, mundo!";
8     return 0;
9 }
```

⊕ Instalación



```
david@david-VirtualBox:~/Escritorio/Codigos$ sudo apt-get install g++
[sudo] contraseña para david:
1234Lo siento, pruebe otra vez.
[sudo] contraseña para david:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  g++-11 libstdc++-11-dev
Paquetes sugeridos:
  g++-multilib g++-11-multilib gcc-11-doc libstdc++-11-doc
Se instalarán los siguientes paquetes NUEVOS:
  g++ g++-11 libstdc++-11-dev
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 218 no actualizados.
Se necesita descargar 13,5 MB de archivos.
Se utilizarán 48,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

⊕ Ejecutar



```
david@david-VirtualBox:~/Escritorio/Codigos$ g++ -o hola hola.cpp
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola
¡Hola, mundo!david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Hola Mundo JAVA

```
Hola.java
~/Escritorio/Codigos
Guarda
- □ ×

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

✚ Instalación

```
david@david-VirtualBox: ~/Escritorio/Codigos
esario.
systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
ca-certificates-java default-jdk-headless default-jre default-jre-headless
fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-jni
libICE-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev
libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre
openjdk-11-jre-headless x11proto-dev xorg-sgml-doctools xtrans-dev
Paquetes sugeridos:
libICE-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo
openjdk-11-source visualvm fonts-ipafont-gothic fonts-ipafont-mincho
fonts-wqy-microhei | fonts-wqy-zenhei
Se instalarán los siguientes paquetes NUEVOS:
ca-certificates-java default-jdk default-jdk-headless default-jre
default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java
libatk-wrapper-jni libICE-dev libpthread-stubs0-dev libsm-dev
libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk
openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless x11proto-dev
xorg-sgml-doctools xtrans-dev
0 actualizados, 24 nuevos se instalarán, 0 para eliminar y 218 no actualizados.
Se necesita descargar 262 MB de archivos.
Se utilizarán 413 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

✚ Ejecutamos

```
Configurando libxt-dev:amd64 (1:1.2.1-1) ...
david@david-VirtualBox:~/Escritorio/Codigos$ javac Hola.java
david@david-VirtualBox:~/Escritorio/Codigos$ java Hola
Hola Mundo
david@david-VirtualBox:~/Escritorio/Codigos$
```

✚ Nos crea el class



✚ Empaquetamos

```
david@david-VirtualBox: ~/Escritorio/Codigos$ jar cvfe hola.jar Hola Hola.class  
manifiesto agregado  
agregando: Hola.class(entrada = 412) (salida = 283)(desinflado 31%)  
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Nos devuelve el Hola Mundo

```
david@david-VirtualBox: ~/Escritorio/Codigos$ jar cvfe hola.jar Hola Hola.class  
manifiesto agregado  
agregando: Hola.class(entrada = 412) (salida = 283)(desinflado 31%)  
david@david-VirtualBox:~/Escritorio/Codigos$ java -jar hola.jar  
Hola Mundo  
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Hola Mundo en NASM

```
Abrir ▾ F+          hola.asm      ~/Escritorio/Codigos  Guardar  ⌂  ×  
1 section .data  
2  
3 msg      db ";Hola Mundo!", 0Ah  
4 len      equ      $ - msg  
5  
6 section .text  
7  
8 global _start  
9  
10 _start:  
11     mov      eax, 04h  
12     mov      ebx, 01h  
13     mov      ecx, msg  
14     mov      edx, len  
15     int      80h  
16     mov      eax, 01h  
17     mov      ebx, 00h  
18     int      80h|
```

⊕ Instalamos

```
[sudo] contraseña para david:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.  
    systemd-hwe-hwdb  
Utilice «sudo apt autoremove» para eliminarlo.  
Se instalarán los siguientes paquetes NUEVOS:  
    nasm  
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 218 no actualizados.  
Se necesita descargar 375 kB de archivos.  
Se utilizarán 3.345 kB de espacio de disco adicional después de esta operación.  
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 nasm amd64 2.15.0-5-1 [375 kB]  
Descargados 375 kB en 0s (853 kB/s)  
Seleccionando el paquete nasm previamente no seleccionado.  
(Leyendo la base de datos ... 208279 ficheros o directorios instalados actualmente.)  
Preparando para desempaquetar .../nasm_2.15.05-1_amd64.deb ...  
Desempaquetando nasm (2.15.05-1) ...  
Configurando nasm (2.15.05-1) ...  
Procesando disparadores para man-db (2.10.2-1) ...  
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Ejecutamos

```
Procesando disparadores para man-db (2.10.2-1) ...  
david@david-VirtualBox:~/Escritorio/Codigos$ nasm -f elf64 hola.asm Id hola.o -o  
hola  
hola.asm: fatal: more than one input file specified: Id  
  
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola  
;Hola, mundo!david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Hola Mundo en RUBY

```
Abrir ▾  ⌂  hola.rb  
~/Escritorio/Codigos  Guardar  ⌂  ×  
1 #!/usr/bin/env ruby  
2  
3 puts "Hola Mundo"
```

⊕ Instalamos

```
david@david-VirtualBox: ~/Escritorio/Codigos
1234
Lo siento, pruebe otra vez.
[sudo] contraseña para david:
sudo: apt.get: orden no encontrada
david@david-VirtualBox:~/Escritorio/Codigos$ sudo apt-get install ruby
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  fonts-lato libjs-jquery libruby3.0 rake ruby-net-telnet ruby-rubygems
  ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration
Paquetes sugeridos:
  ri ruby-dev bundler
Se instalarán los siguientes paquetes NUEVOS:
  fonts-lato libjs-jquery libruby3.0 rake ruby ruby-net-telnet ruby-rubygems
  ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration
0 actualizados, 11 nuevos se instalarán, 0 para eliminar y 218 no actualizados.
Se necesita descargar 8.571 kB de archivos.
Se utilizarán 37,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

⊕ Ejecutamos

```
david@david-VirtualBox:~/Escritorio/Codigos$ chmod +x hola.rb
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola.rb
Hola Mundo
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Hola Mundo en GO

```
Abrir ▾ F+
holago
~/Escritorio/Codigos Guardar ⌂ ×
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Hola mundo desde Go")
7 }
```

⊕ Instalamos

```
david@david-VirtualBox:~/Escritorio/Codigos$ sudo apt-get install golang
[sudo] contraseña para david:
1234
Lo siento, pruebe otra vez.
[sudo] contraseña para david:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  golang-1.18 golang-1.18-doc golang-1.18-go golang-1.18-src golang-doc
  golang-go golang-src libdpkg-perl libfile-fcntllock-perl pkg-config
Paquetes sugeridos:
  bzr | brz git mercurial subversion debian-keyring bzip2 dpkg-dev
Se instalarán los siguientes paquetes NUEVOS:
  golang golang-1.18 golang-1.18-doc golang-1.18-go golang-1.18-src golang-doc
  golang-go golang-src libdpkg-perl libfile-fcntllock-perl pkg-config
0 actualizados, 11 nuevos se instalarán, 0 para eliminar y 218 no actualizados.
Se necesita descargar 82,7 MB de archivos.
Se utilizarán 440 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

⊕ Ejecutamos

```
Procesando disparadores para man-db (2.10.2-1) ...
david@david-VirtualBox:~/Escritorio/Codigos$ go build hola.go
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola
Hola mundo desde Go
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Hola Mundo en RUST

```
Abrir ▾  FN
holo.rs
~/Escritorio/Codigos Guardar ⌂ ×
1 fn main() {
2     println!("¡Hola, mundo! Desde RUST ");
3 }
```

⊕ Instalación

```
david@david-VirtualBox:~/Escritorio/Codigos$ sudo apt-get install rustc
[sudo] contraseña para david:
1234
Lo siento, pruebe otra vez.
[sudo] contraseña para david:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libllvm14 libstd-rust-1.61 libstd-rust-dev
Paquetes sugeridos:
  cargo llvm-14 lld-14 clang-14
Se instalarán los siguientes paquetes NUEVOS:
  libllvm14 libstd-rust-1.61 libstd-rust-dev rustc
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 218 no actualizados.
Se necesita descargar 86,2 MB de archivos.
Se utilizarán 384 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

⊕ Ejecutamos

```
Procesando disparadores para libc-bin (2.35-0ubuntu3.1) ...
david@david-VirtualBox:~/Escritorio/Codigos$ rustc hola.rs
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola
¡Hola, mundo! Desde RUST
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Hola Mundo en LISP

```
Abrir ▾  [+]
holalisp
~/Escritorio/Codigos Guardar ⌂ ⌂ ⌂ ⌂
1 #!/usr/bin/env clisp
2
3 (format t "¡Hola, mundo!")
```

⊕ Instalamos

```
david@david-VirtualBox:~/Escritorio/Codigos$ sudo apt-get install clisp
[sudo] contraseña para david:
1234
Lo siento, pruebe otra vez.
[sudo] contraseña para david:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libffcallib libsigsegv2
Paquetes sugeridos:
  clisp-doc slime clisp-module-berkeley-db clisp-module-clx clisp-module-dbus
  clisp-module-fastcgi clisp-module-gdbm clisp-module-libsvm clisp-module-pari
  clisp-module-pcre clisp-module-postgresql clisp-module-zlib hyperspec
Se instalarán los siguientes paquetes NUEVOS:
  clisp libffcallib libsigsegv2
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 218 no actualizados.
Se necesita descargar 6.438 kB de archivos.
Se utilizarán 33,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

 Ejecutamos

```
Procesando disparadores para man-db (2.10.2-1) ...
david@david-VirtualBox:~/Escritorio/Codigos$ chmod +x hola.lisp
david@david-VirtualBox:~/Escritorio/Codigos$ ./hola.lisp
;Hola, mundo!
david@david-VirtualBox:~/Escritorio/Codigos$
```

2. Para cada uno de los lenguajes anteriores, indica el proceso realizado para conseguir ejecutar el código: ¿compilación o interpretación?

Para ejecutar el código como podemos ver anteriormente utilizamos chmod +x y el nombre del archivo, esto sirve para gestionar los permisos de los archivos o directorios del sistema. Cuando usamos este comando debemos tener presentes los tres niveles de gestión de permisos que existen: Lectura, Escritura y Ejecución.

Por consiguiente, utilizamos ./ y nombre del archivo.

Un compilador genera un programa “stand-alone” es decir que tiene sentido y se puede ejecutar por sí solo, mientras que un programa interpretado siempre necesita su intérprete correspondiente para poder ejecutarse. Por lo tanto, es más fácil distribuir un programa compilado.

3. Para cada uno de los lenguajes anteriores, indica el nombre del compilador o intérprete utilizado en GNU/Linux.

Bash (intérprete) se utiliza para ejecutar scripts escritos en este lenguaje.

Python (intérprete) se utiliza para ejecutar código escrito en este lenguaje.

PHP (compilador) compila código escrito en este lenguaje.

JavaScript es un intérprete Node.js y ejecuta código escrito en este lenguaje.

C: El compilador GCC se utiliza para compilar código escrito en C en GNU/Linux.

C ++: El compilador GCC también se utiliza para utilizar y compilar código escrito en C++ en GNU/Linux.

Java: Se puede utilizar tanto el compilador javac y la intérprete java, que vienen incluidos en el paquete de java.

Ruby: Se utiliza el intérprete ruby.

Go: Se puede utilizar el compilador go.

Para ruts: Se utiliza el compilador rustc.

Para Lisp, se pueden utilizar diferentes intérpretes, como clisp o sbcl.

4. Investiga y averigua que extensión tienen los archivos de código fuente de los siguientes lenguajes:

- a. Bash ".sh"
- b. Python ".py"
- c. Php ".php"
- d. Javascript ".js"
- e. C ".c"
- f. c++ ".cpp"
- g. java ".java"
- h. ensamblador ".asm"
- i. ruby ".rb"
- j. go ".go"
- k. rust ".rs"
- l. Lisp ".lisp"

5. Scripts ejecutables para los lenguajes interpretados. Edita los scripts para los siguientes lenguajes:

- bash

- python
- php
- javascript
- java
- ruby
- go
- rust
- lisp

En el ejercicio 1 se encuentran los scripts ejecutables interpretados:

[Principio del documento](#)

6. ¿Qué extensión tienen los archivos de código objeto?

Los archivos de código objeto tienen diferentes extensiones, dependiendo del lenguaje de programación y del sistema operativo en el que se esté trabajando. Algunas de las extensiones más comunes para los archivos de código objeto son:

.o

.obj

.out

Es importante tener en cuenta que la extensión de un archivo de código objeto no es una indicación de su contenido o de cómo se debe utilizar. Por lo tanto, si necesita abrir un archivo con una de estas extensiones, es recomendable utilizar un programa adecuado para ese fin, como un compilador o un depurador.

7. Lenguaje C. Código en varios archivos. Obtener el código objeto a partir del código fuente de los 3 archivos siguientes:

- ⊕ Creamos 3 archivos diferentes de texto con los siguientes códigos:
- ⊕ datos.c

A screenshot of a text editor window titled "datos.c". The file path is shown as "~/Escritorio/Codigos". The code in the editor is:

```
1 //-----
2 // datos.c
3 //-----
4
5 char *mensaje="Hola a todos y todas";
6 int num1 = 8;
7 int num2 = 10;
```

- ⊕ suma.c

A screenshot of a text editor window titled "suma.c". The file path is shown as "~/Escritorio/Codigos". The code in the editor is:

```
1 //-----
2 // suma.c
3 //-----
4
5 int suma (int a, int b) {
6     return a + b;
7 }
```

- ⊕ main.c

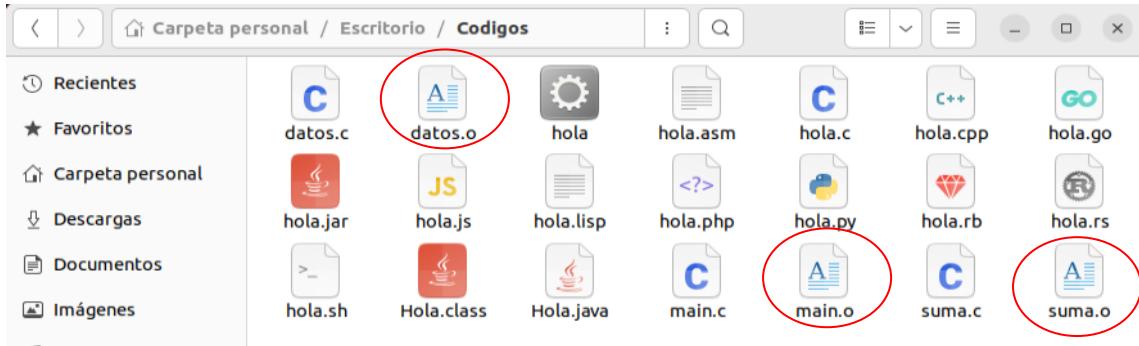
A screenshot of a text editor window titled "main.c". The file path is shown as "~/Escritorio/Codigos". The code in the editor is:

```
1 //-----
2 // main.c
3 //-----
4
5 #include <stdio.h>
6
7 int suma (int a, int b);
8
9 extern char *mensaje;
10 extern int num1, num2;
11
12 int main(){
13     printf("%s\n", mensaje);
14     printf("%d\n", suma (num1, num2) );
15     return 0;
16 }
```

- ⊕ Obtenemos el código objeto con gcc -c main.c datos.c suma.c

```
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -c main.c
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -c datos.c
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -c suma.c
david@david-VirtualBox:~/Escritorio/Codigos$
```

- Como podemos comprobar nos ha generado los .o



8. Lenguaje C. Código en varios archivos. Obtener el código binario ejecutable a partir del código objeto de los 3 archivos anteriores:

- Lo haremos con gcc -o programa main.o datos.o suma.o
- Y ./programa

```
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -o programa datos.o suma.o main.o
david@david-VirtualBox:~/Escritorio/Codigos$ ./programa
Hola a todos y todas
18
david@david-VirtualBox:~/Escritorio/Codigos$
```

9. Lenguaje C++. Código en varios archivos. Obtener el código objeto a partir del código fuente de los 3 archivos siguientes:

- Creamos 3 archivos diferentes de texto

✚ datos.cpp

```
1 //-----
2 // datos.cpp
3 //-----
4 #include <string>
5
6 std::string mensaje = "Hola a todos y todas";
7
8 int num1 = 8;
9 int num2 = 10;
```

✚ main.cpp

```
1 //-----
2 // main.cpp
3 //-----
4 #include <iostream>
5
6 using namespace std;
7
8 int suma (int a, int b);
9
10 extern string mensaje;
11 extern int num1, num2;
12
13 int main()
14 {
15     cout << mensaje << endl;
16     cout << suma (num1, num2) << endl;
17     return 0;
18 }
```

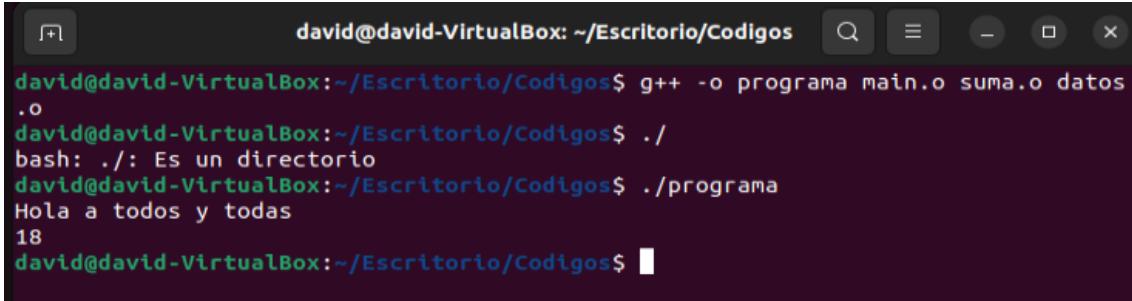
✚ suma.cpp

```
1 //-----
2 // suma.cpp
3 //-----
4 int suma (int a, int b)
5 {
6     return a + b;
7 }
```

✚ Ejecutamos los archivos con g++ -c y luego ./programa

```
david@david-VirtualBox:~/Escritorio/Codigos$ g++ -c main.cpp suma.cpp datos.cpp
david@david-VirtualBox:~/Escritorio/Codigos$ ./programa
Hola a todos y todas
18
david@david-VirtualBox:~/Escritorio/Codigos$
```

10. Lenguaje C++. Código en varios archivos. Obtener el código binario ejecutable a partir del código objeto de los 3 archivos anteriores:



The screenshot shows a terminal window titled "david@david-VirtualBox: ~/Escritorio/Codigos". The user runs the command "g++ -o programa main.o suma.o datos.o" to link three object files into an executable named "programa". When executed with "./programa", it prints "Hola a todos y todas" followed by the number "18".

```
david@david-VirtualBox:~/Escritorio/Codigos$ g++ -o programa main.o suma.o datos.o
david@david-VirtualBox:~/Escritorio/Codigos$ ./
bash: ./: Es un directorio
david@david-VirtualBox:~/Escritorio/Codigos$ ./programa
Hola a todos y todas
18
david@david-VirtualBox:~/Escritorio/Codigos$
```

11. Bibliotecas. Define que se entiende por biblioteca o librería y los tipos que existen.

Una biblioteca es un conjunto de códigos pre-escritos que pueden ser utilizados en un programa para agregar funcionalidades específicas. Por ejemplo, una biblioteca de matemáticas puede contener funciones como raíz cuadrada, logaritmo y trigonometría, que se pueden utilizar en un programa para realizar cálculos matemáticos.

- Biblioteca estática
- Biblioteca dinámica
- Biblioteca dinámica como plugin

12. Bibliotecas. ¿Qué tipo es el más utilizado actualmente? ¿Por qué?

No se puede decir que una librería sea mejor que otra de manera general. Cada librería tiene sus propias características y ventajas, y la mejor opción depende del contexto en el que se vaya a utilizar.

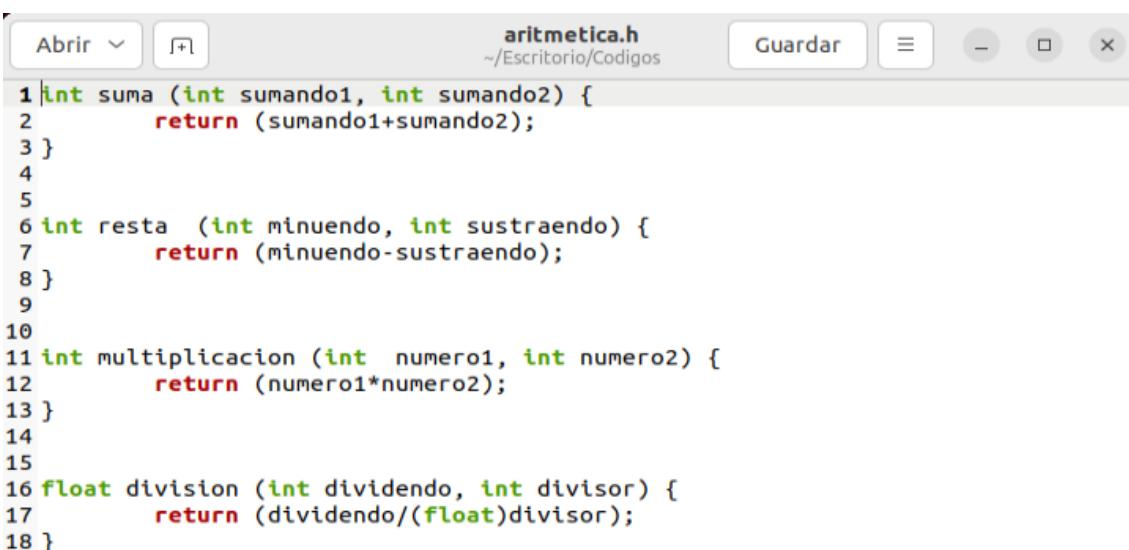
Una librería estática es un conjunto de código que se compila junto con el código de la aplicación, lo que aumenta el tamaño del ejecutable final. Sin embargo, al estar integrada en el ejecutable, la aplicación puede ejecutarse sin depender de la librería externamente.

Por otro lado, una librería dinámica es un archivo que se carga en memoria cuando se ejecuta la aplicación, lo que permite compartir la librería entre diferentes programas. Esto puede reducir el tamaño del ejecutable, pero requiere que la librería esté disponible en el sistema en el que se ejecute la aplicación.

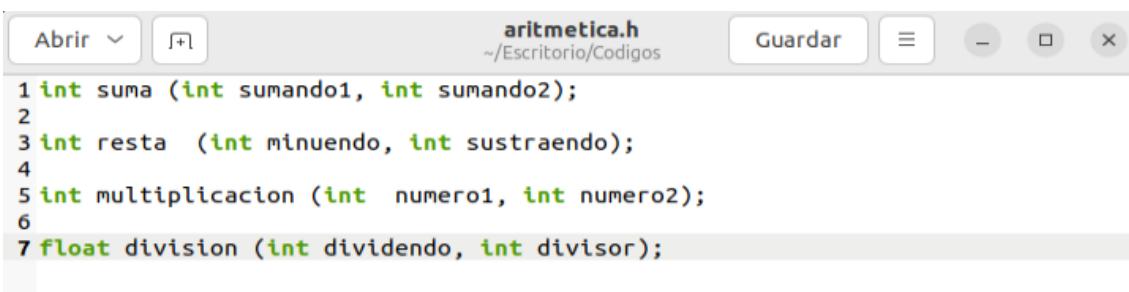
En resumen, cada tipo de librería tiene sus propias ventajas y desventajas, y la mejor opción depende de las necesidades de cada aplicación.

13. Bibliotecas. Crea una biblioteca dinámica en C que proporcione las funciones para sumar, restar, multiplicar y dividir 2 números enteros. Crea un programa que haga uso de ella y comprueba que se ejecuta correctamente.

⊕ Creamos biblioteca dinámica como aritmetica.h



```
1 int suma (int sumando1, int sumando2) {
2     return (sumando1+sumando2);
3 }
4
5
6 int resta (int minuendo, int sustraendo) {
7     return (minuendo-sustraendo);
8 }
9
10
11 int multiplicacion (int numero1, int numero2) {
12     return (numero1*numero2);
13 }
14
15
16 float division (int dividendo, int divisor) {
17     return (dividendo/(float)divisor);
18 }
```



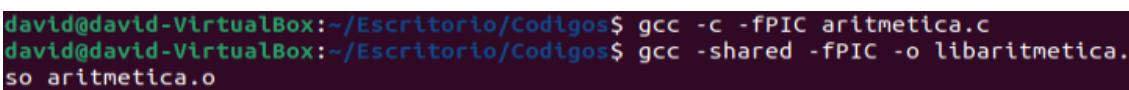
```
1 int suma (int sumando1, int sumando2);
2
3 int resta (int minuendo, int sustraendo);
4
5 int multiplicacion (int numero1, int numero2);
6
7 float division (int dividendo, int divisor);
```

⊕ Compilamos a código objeto



```
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -c aritmetica.c
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Empaqueamos en biblioteca estática libaritmetica.a



```
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -c -fPIC aritmetica.c
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -shared -fPIC -o libaritmetica.a
so aritmetica.o
```

- Crearemos un main.c con los siguientes parámetros

```

1 #include <stdio.h>
2 #include "aritmetica.h"
3
4 #define NUM1      5
5 #define NUM2      2
6
7
8 int main (){
9     printf ("Dados los números %d y %d\n", NUM1, NUM2);
10    printf ("La suma es %d\n", suma (NUM1, NUM2));
11    printf ("La resta es %d\n", resta (NUM1, NUM2));
12    printf ("La multiplicación es %d\n", multiplicacion (NUM1, NUM2));
13    printf ("La división es %f\n", division (NUM1, NUM2));
14    return 0;
15 }

```

- Distribución de binario junto a biblioteca en la misma carpeta

```
david@david-VirtualBox:~/Escritorio/Codigos$ ldd main
    linux-vdso.so.1 (0x000007ffe6e360000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000007fb69effc000)
    /lib64/ld-linux-x86-64.so.2 (0x000007fb69f23a000)
david@david-VirtualBox:~/Escritorio/Codigos$ 
```

```
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -L. -Wl,-rpath=. -Wall -o main
main.c -laritmetica
david@david-VirtualBox:~/Escritorio/Codigos$ ldd main
    linux-vdso.so.1 (0x000007ffccbb5000)
    libaritmetica.so => ./libaritmetica.so (0x000007feec1e1c000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000007feec1be5000)
    /lib64/ld-linux-x86-64.so.2 (0x000007feec1e28000)
david@david-VirtualBox:~/Escritorio/Codigos$ 
```

- Creamos ejecutable estático

```
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -o main main.c libaritmetica.a
david@david-VirtualBox:~/Escritorio/Codigos$ ldd main
    linux-vdso.so.1 (0x000007ffceb36e000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000007fe0014fa000)
    /lib64/ld-linux-x86-64.so.2 (0x000007fe001738000)
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -o main main.c libaritmetica.so
david@david-VirtualBox:~/Escritorio/Codigos$ ldd main
    linux-vdso.so.1 (0x000007ffd4f0ae000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000007f8268beb000)
    /lib64/ld-linux-x86-64.so.2 (0x000007f8268e29000)
david@david-VirtualBox:~/Escritorio/Codigos$ 
```

- Y ya podríamos ejecutar con ./main

```
La suma de 10 y 2 es 12
La multiplicación de 10 y 2 es 20
La resta de 10 y 2 es 8
La división de 10 y 2 es 5
```

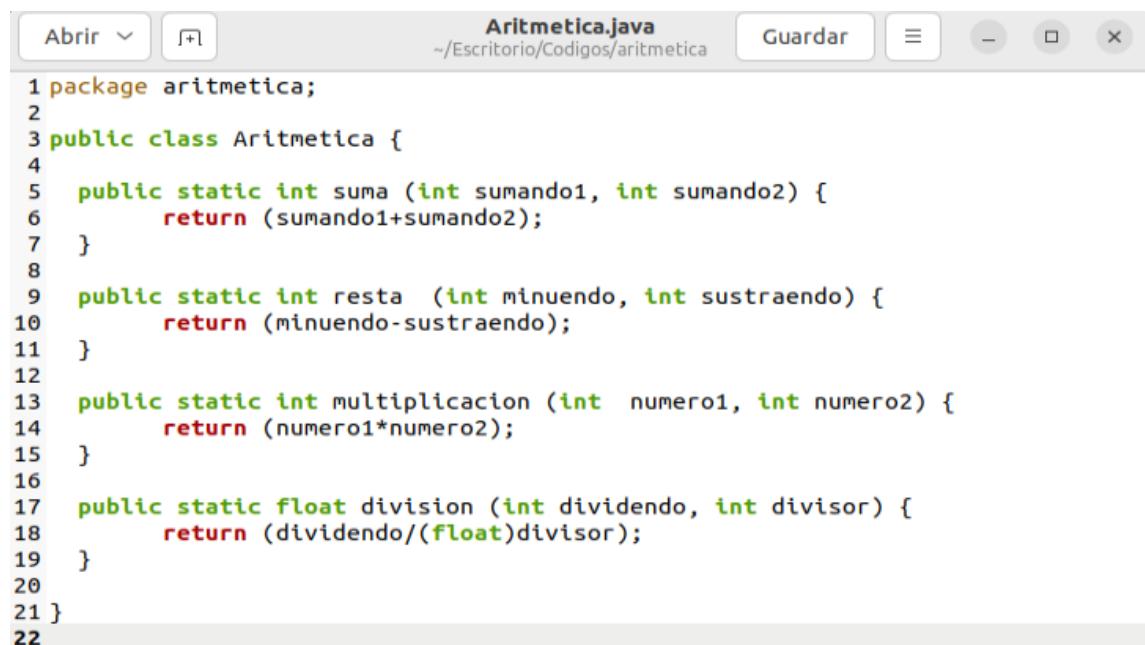
- Bibliotecas. Crea una biblioteca dinámica en Java que proporcione las funciones para sumar, restar, multiplicar y dividir 2 números

enteros. Crea un programa que haga uso de ella y comprueba que se ejecuta correctamente.

- Crearemos directorio aritmética

```
david@david-VirtualBox:~/Escritorio/Codigos$ mkdir aritmetica
```

- Compilamos este código:

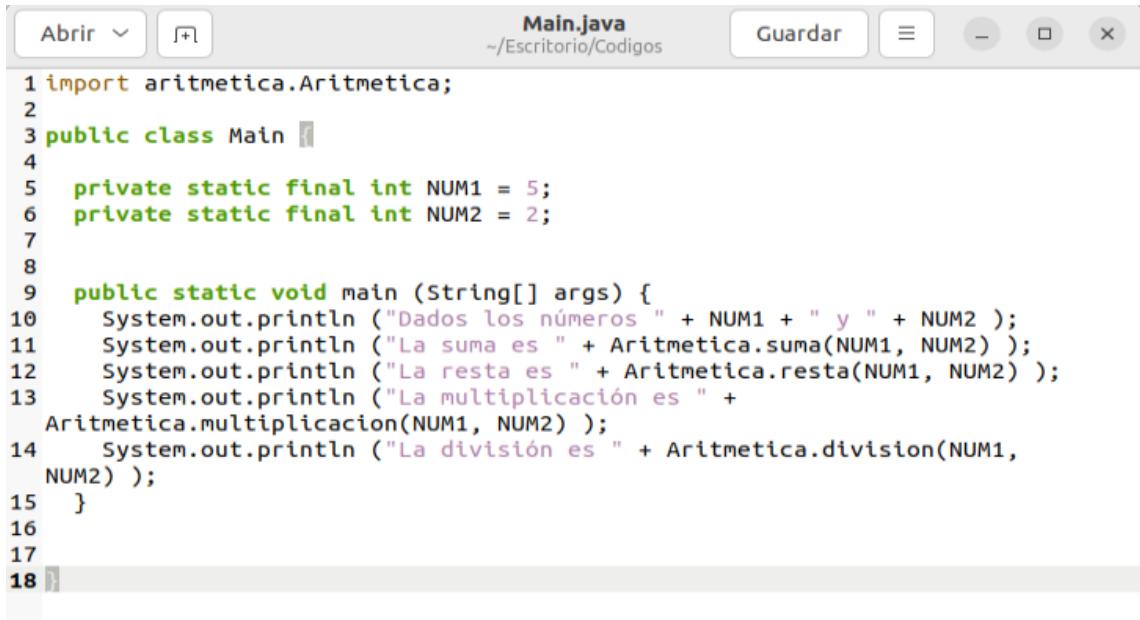


```
Abrir + Aritmetica.java ~/Escritorio/Codigos/aritmetica Guardar ... x
1 package aritmetica;
2
3 public class Aritmetica {
4
5     public static int suma (int sumando1, int sumando2) {
6         return (sumando1+sumando2);
7     }
8
9     public static int resta (int minuendo, int sustraendo) {
10        return (minuendo-sustraendo);
11    }
12
13    public static int multiplicacion (int numero1, int numero2) {
14        return (numero1*numero2);
15    }
16
17    public static float division (int dividendo, int divisor) {
18        return (dividendo/(float)divisor);
19    }
20
21 }
22
```

- Compilamos y creamos paquete jar

```
david@david-VirtualBox:~/Escritorio/Codigos$ javac aritmetica/Aritmetica.java
david@david-VirtualBox:~/Escritorio/Codigos$ jar cvf aritmetica.jar aritmetica/*.class
manifiesto agregado
agregando: aritmetica/Aritmetica.class(entrada = 434) (salida = 265)(desinflado
38%)
david@david-VirtualBox:~/Escritorio/Codigos$
```

- Creamos Main.java



```
1 import aritmetica.Aritmetica;
2
3 public class Main {
4
5     private static final int NUM1 = 5;
6     private static final int NUM2 = 2;
7
8
9     public static void main (String[] args) {
10        System.out.println ("Dados los números " + NUM1 + " y " + NUM2 );
11        System.out.println ("La suma es " + Aritmetica.suma(NUM1, NUM2) );
12        System.out.println ("La resta es " + Aritmetica.resta(NUM1, NUM2) );
13        System.out.println ("La multiplicación es " +
14            Aritmetica.multiplicacion(NUM1, NUM2) );
15        System.out.println ("La división es " + Aritmetica.division(NUM1,
16            NUM2) );
17    }
18 }
```

Compilamos

```
david@david-VirtualBox:~/Escritorio/Codigos$ javac Main.java
david@david-VirtualBox:~/Escritorio/Codigos$
```

Ejecutamos y como podemos comprobar se realizó correctamente.

```
david@david-VirtualBox:~/Escritorio/Codigos$ java Main
Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.5
```

15. Bibliotecas. Busca información y explica las ventajas y desventajas de usar bibliotecas estáticas.

Ventajas:

- Programa autocontenido.
- Las actualizaciones de la biblioteca no le afectan.

Desventajas:

- Mayor tamaño del programa.
- El programa no se beneficia de las actualizaciones de la biblioteca.

16. Bibliotecas. Busca información y explica las ventajas y desventajas de usar bibliotecas dinámicas.

Ventajas:

- Menor tamaño del programa.
- El programa se beneficia de las actualizaciones de la biblioteca.

Desventajas:

- Programa NO autocontenido.
- Las actualizaciones de la biblioteca le afectan para bien y para mal.

17. Build. Automatiza el proceso de compilación de ejecutable y biblioteca, su enlazado y la generación del archivo ejecutable para código fuente en C con make. Haz uso de un buildfile.

 Creamos un makefile con los siguientes elementos.

The screenshot shows a code editor window with the following details:

- Title Bar:** The title bar displays "makefile" and the path "~/Escritorio/Codigos".
- Toolbar:** The toolbar includes standard icons for "Abrir" (Open), "Guardar" (Save), and window controls.
- Code Area:** The main area contains a Makefile script with numbered lines from 1 to 45. The code is color-coded for readability, with blue for comments and red for variables like \$(CC) and \$(CFLAGS).

```
1 ##### EJEMPLO DE ARCHIVO Makefile (cc0) jamj2000
2 #
3 #   EJEMPLO DE ARCHIVO Makefile (cc0) jamj2000
4 #
5 #####
6
7
8 ##### MACROS
9
10 # Compilador de C
11 CC      = gcc
12
13 # Opciones del compilador, en este caso Optimización
14 CFLAGS = -O
15
16 # Directorio de instalación
17 PREFIX = /usr/local
18
19 # Ejecutable resultante
20 OUTPUT = programa
21
22
23 ##### REGLAS
24
25 # .PHONY indica objetivos especiales, que no corresponden a archivos
26 #
27 # Las reglas tienen la forma
28 # objetivo(target) : dependencias
29 # <TAB> comando1
30 # <TAB> comando2
31 # <TAB> ...
32 #
33 # La primera regla es la regla por defecto, puede invocarse simplemente con
34 # la orden make
35 all: main.o libaritmetica.so
36     $(CC) $(CFLAGS) -Wl,-rpath=$(PREFIX)/lib main.o libaritmetica.so
37     -o $(OUTPUT)
38 .PHONY: all
39
40 main.o: main.c aritmetica.h
41     $(CC) $(CFLAGS) -c main.c
42
43
44 aritmetica.o: aritmetica.c
45     $(CC) $(CFLAGS) -c -fPIC aritmetica.c
46
```

```

44 aritmetica.o: aritmetica.c
45     $(CC) $(CFLAGS) -c -fPIC aritmetica.c
46
47
48 libaritmetica.so: aritmetica.o
49     $(CC) $(CFLAGS) -shared -fPIC -o libaritmetica.so aritmetica.o
50
51
52 clean:
53     rm *.o *.so $(OUTPUT)
54 .PHONY: clean
55
56
57 install: all
58     [ -d $(PREFIX) ] || mkdir $(PREFIX)
59     [ -d $(PREFIX)/bin ] || mkdir $(PREFIX)/bin
60     [ -d $(PREFIX)/lib ] || mkdir $(PREFIX)/lib
61     install -m 0755 $(OUTPUT) $(PREFIX)/bin
62     install -m 0644 libaritmetica.so $(PREFIX)/lib
63 .PHONY: install
64
65
66 help:
67     @echo "Objetivos válidos para make:"
68     @echo ""
69     @echo "  all (el objetivo por defecto si no se indica nada)"
70     @echo "  clean"
71     @echo "  install"
72     @echo "  help"
73     @echo ""
74 .PHONY: help

```

Makefile ▼ Anchura del tabulador: 8 ▼ Ln 1, Col 1 ▼ INS

- En el caso del archivo *Makefile*, la primera regla es la regla por defecto, en este caso la hemos llamado *all*. Sirve para compilar y enlazar todo, generando el ejecutable y la biblioteca asociada. Si ejecutamos `make` sin argumentos, se ejecutará la regla por defecto y obtendremos la siguiente salida:

```

david@david-VirtualBox:~/Escritorio/Codigos$ make
make: *** No hay objetivos. Alto.
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -O -c main.c
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -O -c -fPIC aritmetica.c
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -O -shared -fPIC -o libaritmetica.so aritmetica.o
david@david-VirtualBox:~/Escritorio/Codigos$ gcc -O -Wl,-rpath=/usr/local/lib main.o libaritmetica.so -o programa

```

```

david@david-VirtualBox:~/Escritorio/Codigos$ gcc -O -Wl,-rpath=/usr/local/lib main.o libaritmetica.so -o programa
david@david-VirtualBox:~/Escritorio/Codigos$ 

```

- Instalamos aunque lo teníamos ya de forma automática.

```
david@david-VirtualBox:~/Escritorio/Codigos$ sudo apt install make
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
make ya está en su versión más reciente (4.3-4.1build1).
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 222 no actualizados.
david@david-VirtualBox:~/Escritorio/Codigos$
```

18. Build. Automatiza el proceso de compilación de ejecutable y biblioteca, su enlazado y la generación del archivo .jar para código fuente en Java con Ant. Haz uso de un buildfile.

Creamos el xml



```
*build.xml
~/Escritorio/Codigos
Guardar
×
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project name="programa" default="jar" basedir=".">
3   <description>Programa que usa biblioteca aritmética.</description>
4   <!-- Ejemplo de archivo de construcción (buildfile)
5     Para crear este archivo se ha consultado:
6       https://ant.apache.org/manual/tutorial-HelloWorldWithAnt.html
7       (cc0) jamj2000
8   -->
9   <property name="src.dir"      value="."/>
10  <property name="build.dir"    value="build"/>
11  <property name="classes.dir"  value="${build.dir}/classes"/>
12  <property name="jar.dir"      value="${build.dir}/jar"/>
13  <property name="main-class"   value="Main"/>
14
15
16  <!-- Creamos directorios para el resultado de la compilación -->
17  <target name="init">
18    <mkdir dir="${classes.dir}"/>
19    <mkdir dir="${jar.dir}"/>
20  </target>
21
22
23  <!-- Indicamos directorio donde se hallan las clases -->
24  <path id="compile.classpath">
25    <fileset dir="aritmética" />
26  </path>
27
28
29  <!-- Compilamos -->
30  <target name="compile" depends="init" >
31    <javac srcdir="${src.dir}" destdir="${classes.dir}"
32      includeantruntime="false" debug="true" >
33      <classpath refid="compile.classpath"/>
34    </javac>
35  </target>
36
37  <!-- Creamos archivo .jar -->
38  <target name="jar" depends="compile" >
39    <jar destfile="${jar.dir}/${ant.project.name}.jar" basedir="${classes.dir}">
40      <manifest>
41        <attribute name="Main-Class" value="${main-class}"/>
42      </manifest>
43    </jar>
44  </target>
45
46
47  <!-- Ejecutamos -->
48  <target name="run" depends="jar" >
49    <java jar="${jar.dir}/${ant.project.name}.jar" fork="true" />
50  </target>
51
52
```

```

57
58      <!-- Mostramos ayuda -->
59      <target name="help">
60          <echo level="info">
61              Objetivos válidos para ant:
62
63          jar (el objetivo por defecto si no se indica nada)
64          compile
65          run
66          clean
67          help
68      </echo>
69  </target>
70
71 </project>
72

```

Instalamos ant

```

david@david-VirtualBox:~/Escritorio/Codigos$ sudo apt install ant
[sudo] contraseña para david:
1234
Lo siento, pruebe otra vez.
[sudo] contraseña para david:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  systemd-hwe-hwdb
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  ant-optional
Paquetes sugeridos:
  ant-doc antlr javacc junit junit4 jython libactivation-java libbcel-java
  libbsf-java libcommons-logging-java libcommons-net-java libmail-java
  libjaxp1.3-java libjdepend-java libjsch-java liblog4j1.2-java liboro-java
  libregexp-java libxalan2-java libxml-commons-resolver1.1-java libxz-java
Se instalarán los siguientes paquetes NUEVOS:
  ant ant-optional
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 222 no actualizados.

```

Introducimos ant

```

david@david-VirtualBox:~/Escritorio/Codigos$ ant
Buildfile: /home/david/Escritorio/Codigos/build.xml

init:
  [mkdir] Created dir: /home/david/Escritorio/Codigos/build/classes
  [mkdir] Created dir: /home/david/Escritorio/Codigos/build/jar

compile:
  [javac] Compiling 2 source files to /home/david/Escritorio/Codigos/build/classes

BUILD FAILED
/home/david/Escritorio/Codigos/build.xml:31: /home/david/Escritorio/Codigos/aritmetica does not exist.

Total time: 1 second

```

Ant help

```
david@david-VirtualBox:~/Escritorio/Codigos$ ant help
Buildfile: /home/david/Escritorio/Codigos/build.xml

help:
[echo]
[echo]      Objetivos válidos para ant:
[echo]
[echo]          jar (el objetivo por defecto si no se indica nada)
[echo]          compile
[echo]          run
[echo]          clean
[echo]          help
[echo]

BUILD SUCCESSFUL
Total time: 1 second
```

⊕ Ant run para correrlo y listo.

```
david@david-VirtualBox:~/Escritorio/Codigos$ ant run
Buildfile: /home/david/Escritorio/Codigos/build.xml

init:

compile:
[javac] Compiling 2 source files to /home/david/Escritorio/Codigos/build/classes

BUILD FAILED
/home/david/Escritorio/Codigos/build.xml:31: /home/david/Escritorio/Codigos/aritmetica does not exist.

Total time: 1 second
```

19. Build. Automatiza el proceso de compilación de ejecutable y biblioteca, su enlazado y la generación del archivo jar para código fuente en Java con Maven. Haz uso de un buildfile.

⊕ Instalamos tree

```
libwagon-http-shaded-java libwagon-provider-api-java
Paquetes sugeridos:
libaopalliance-java-doc libatinject-jsr330-api-java-doc libel-api-java
libcommons-io-java-doc libcommons-lang3-java-doc libasm-java libcglib-java
libjsr305-java-doc libmaven-shared-utils-java-doc liblogback-java
libplexus-classworlds-java-doc libplexus-sec-dispatcher-java-doc
libplexus-utils2-java-doc junit4 testng libcommons-logging-java
liblog4j1.2-java
Se instalarán los siguientes paquetes NUEVOS:
libaopalliance-java libapache-pom-java libatinject-jsr330-api-java
libcdi-api-java libcommons-cli-java libcommons-io-java libcommons-lang3-java
libcommons-parent-java libgeronimo-annotation-1.3-spec-java
libhawtjni-runtime-java libjansi-java libjansi-native-java libjsr305-java
libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java
libmaven3-core-java libplexus-cipher-java libplexus-classworlds-java
libplexus-component-annotations-java libplexus-interpolation-java
libplexus-sec-dispatcher-java libplexus-utils2-java libsisu-inject-java
libsisu-plexus-java libslf4j-java libwagon-file-java
libwagon-http-shaded-java libwagon-provider-api-java maven
0 actualizados, 33 nuevos se instalarán, 0 para eliminar y 222 no actualizados.
Se necesita descargar 10,2 MB de archivos.
Se utilizarán 13,3 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

```
david@david-VirtualBox: ~/Escritorio/Codigos
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.jar (4.3 kB at 51 kB/s)
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: /home/david/Escritorio/Codigos
[INFO] Parameter: package, Value: com.mycompany.app
[INFO] Parameter: groupId, Value: com.mycompany.app
[INFO] Parameter: artifactId, Value: my-app
[INFO] Parameter: packageName, Value: com.mycompany.app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/david/Escritorio/Codigos/my-app
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  52.637 s
[INFO] Finished at: 2022-12-15T14:52:06+01:00
[INFO] -----
david@david-VirtualBox:~/Escritorio/Codigos$
```

⊕ Creamos el proyecto

```
david@david-VirtualBox: ~/Escritorio/Codigos
david@david-VirtualBox:~/Escritorio/Codigos$ mvn archetype:generate -DgroupId=com.mimpresa.app -DartifactId=mi-app -Dversion=1.0.0 \
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
[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]
[INFO] Generating project in Batch mode
```

⊕ Obtenemos el pom

```
david@david-VirtualBox:~/Escritorio/Codigos/mi-app$ cd mi-app && ls
cat pom.xml
pom.xml src
<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/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.miempresa.app</groupId>
    <artifactId>mi-app</artifactId>
    <packaging>jar</packaging>
    <version>1.0.0</version>
    <name>mi-app</name>
    <url>http://maven.apache.org</url>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
david@david-VirtualBox:~/Escritorio/Codigos/mi-app$
```

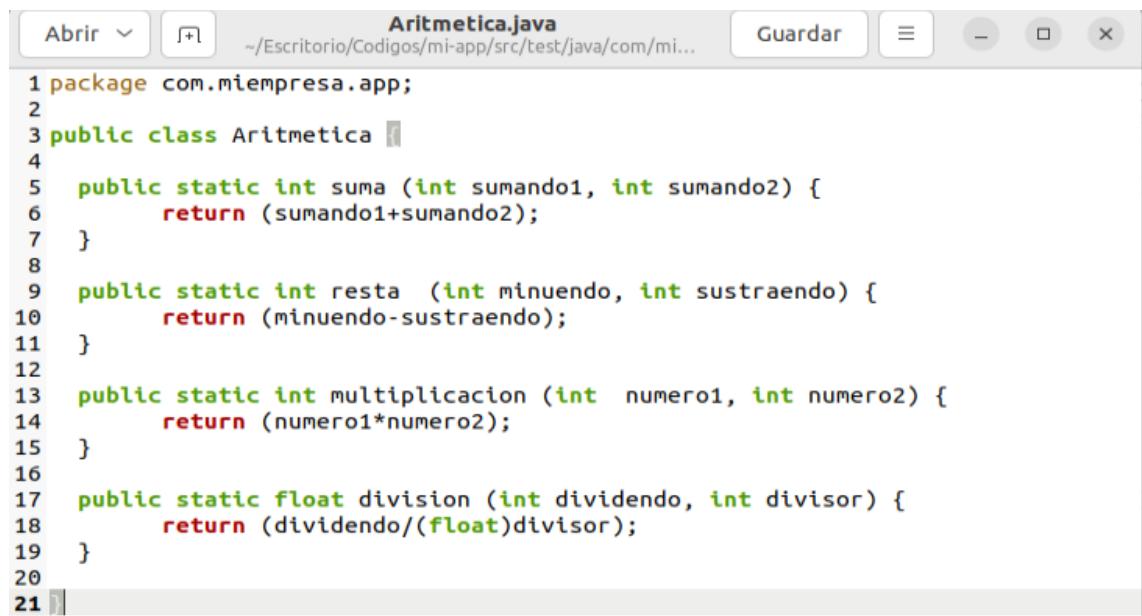
⊕ Creamos los siguientes archivos



The screenshot shows a Java code editor window with the following details:

- Title Bar:** Shows "App.java" as the active file.
- File Path:** ~/Escritorio/Codigos/my-app/src/main/java/com/miempresa/app
- Toolbar:** Includes buttons for "Abrir" (Open), "Guardar" (Save), and other standard file operations.
- Code Area:** Displays the following Java code:

```
1 package com.miempresa.app;
2
3 public class App {
4
5     private static final int NUM1 = 5;
6     private static final int NUM2 = 2;
7
8
9     public static void main (String[] args) {
10        System.out.println ("Dados los números " + NUM1 + " y " + NUM2 );
11        System.out.println ("La suma es " + Aritmetica.suma(NUM1, NUM2) );
12        System.out.println ("La resta es " + Aritmetica.resta(NUM1, NUM2) );
13        System.out.println ("La multiplicación es " +
14            Aritmetica.multiplicacion(NUM1, NUM2) );
15        System.out.println ("La división es " + Aritmetica.division(NUM1,
16            NUM2) );
16    }
17}
```

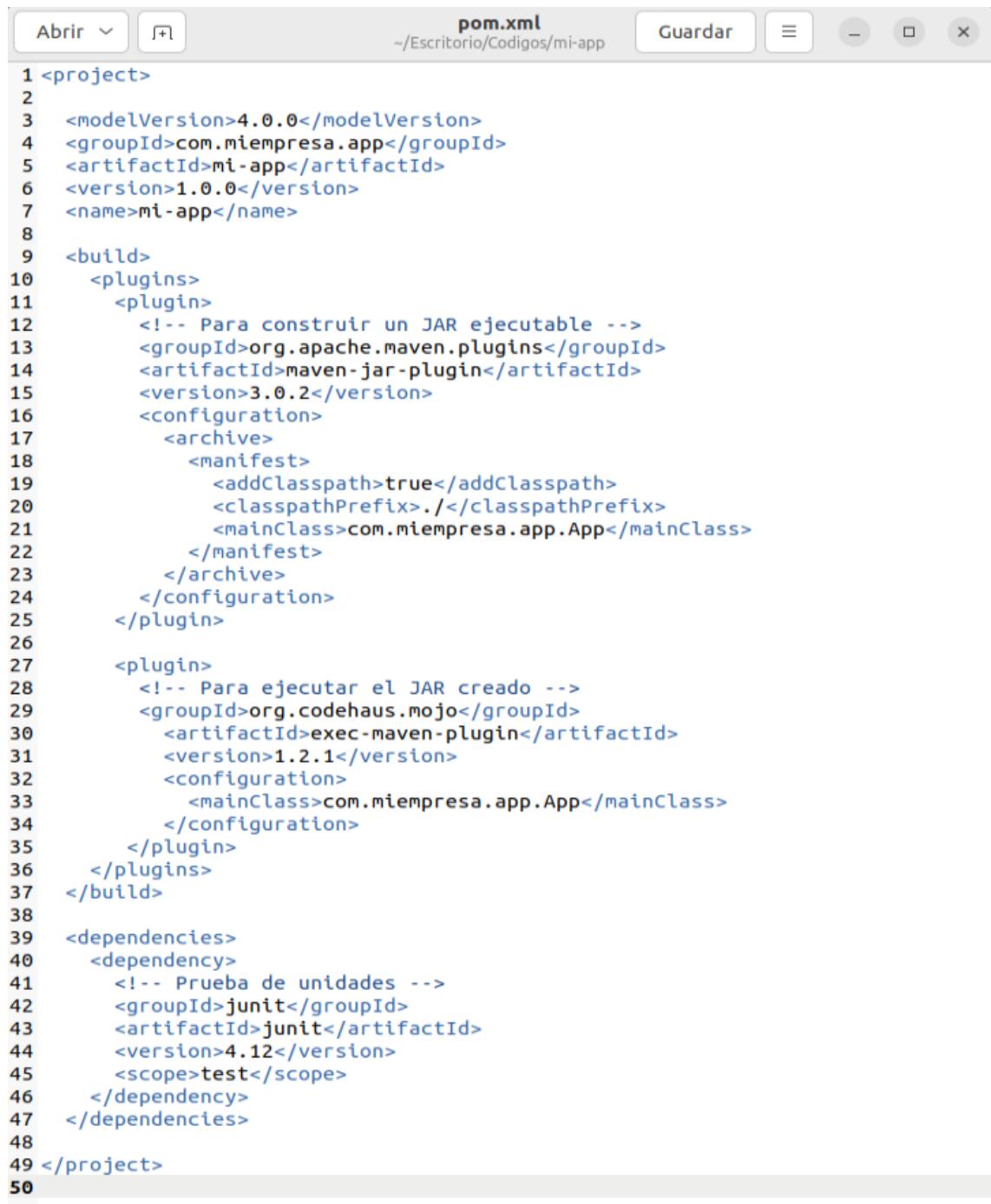


```
1 package com.miempresta.app;
2
3 public class Aritmetica {
4
5     public static int suma (int sumando1, int sumando2) {
6         return (sumando1+sumando2);
7     }
8
9     public static int resta (int minuendo, int sustraendo) {
10        return (minuendo-sustraendo);
11    }
12
13    public static int multiplicacion (int numero1, int numero2) {
14        return (numero1*numero2);
15    }
16
17    public static float division (int dividendo, int divisor) {
18        return (dividendo/(float)divisor);
19    }
20
21 }
```



```
1 package com.miempresta.app;
2
3 import org.junit.Test;
4 import static org.junit.Assert.*;
5
6 public class AppTest {
7
8     @Test
9     public void testMain() {
10         // Prueba vacía
11     }
12
13 }
```

⊕ Modificamos el pom



The screenshot shows a code editor window with the file `pom.xml` open. The title bar includes buttons for 'Abrir' (Open), '+', 'Guardar' (Save), and standard window controls. The code itself is a Maven project descriptor with the following structure:

```
1 <project>
2
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.miempreza.app</groupId>
5   <artifactId>mi-app</artifactId>
6   <version>1.0.0</version>
7   <name>mi-app</name>
8
9   <build>
10    <plugins>
11      <plugin>
12          <!-- Para construir un JAR ejecutable -->
13          <groupId>org.apache.maven.plugins</groupId>
14          <artifactId>maven-jar-plugin</artifactId>
15          <version>3.0.2</version>
16          <configuration>
17              <archive>
18                  <manifest>
19                      <addClasspath>true</addClasspath>
20                      <classpathPrefix>./</classpathPrefix>
21                      <mainClass>com.miempreza.app.App</mainClass>
22                  </manifest>
23              </archive>
24          </configuration>
25      </plugin>
26
27      <plugin>
28          <!-- Para ejecutar el JAR creado -->
29          <groupId>org.codehaus.mojo</groupId>
30          <artifactId>exec-maven-plugin</artifactId>
31          <version>1.2.1</version>
32          <configuration>
33              <mainClass>com.miempreza.app.App</mainClass>
34          </configuration>
35      </plugin>
36  </plugins>
37 </build>
38
39 <dependencies>
40     <dependency>
41         <!-- Prueba de unidades -->
42         <groupId>junit</groupId>
43         <artifactId>junit</artifactId>
44         <version>4.12</version>
45         <scope>test</scope>
46     </dependency>
47 </dependencies>
48
49 </project>
50
```

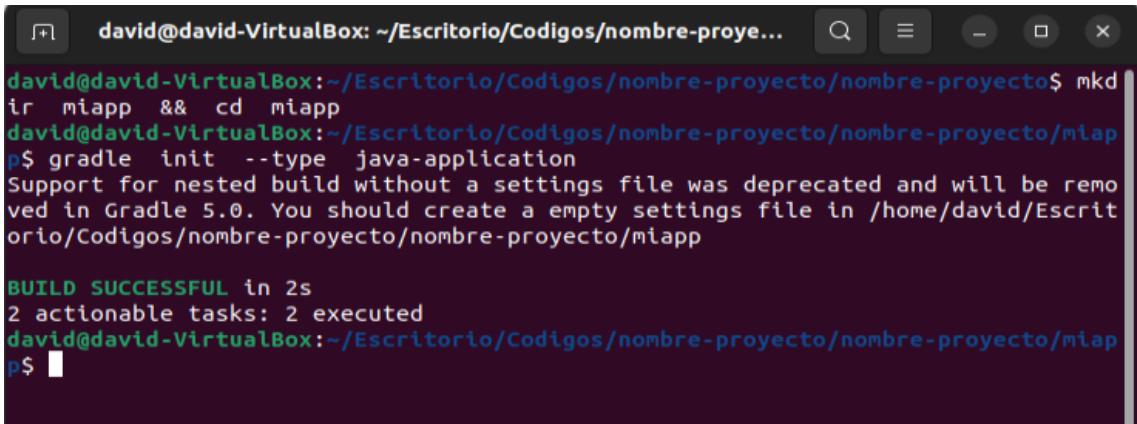
 Y compilamos.

```
david@david-VirtualBox:~/Escritorio/Codigos/mi-app$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ----- < com.miempresa.app:mi-app > -----
[INFO] Building mi-app 1.0.0
[INFO] ----- [ jar ] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mi-app ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i
.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/david/Escritorio/Codigos/mi-app
/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mi-app ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. bu
ild is platform dependent!
[INFO] Compiling 1 source file to /home/david/Escritorio/Codigos/mi-app/target/c
lasses
```

20. Build. Automatiza el proceso de compilación de ejecutable y biblioteca, su enlazado y la generación del archivo jar para código fuente en Java con Gradle. Haz uso de un buildfile.

- ⊕ Creamos proyecto.

```
david@david-VirtualBox:~/Escritorio/Codigos/nombre-proyecto$ mkdir nombre-proye
cto && cd nombre-proyecto
gradle init --type java-application
Starting a Gradle Daemon (subsequent builds will be faster)
<-----> 0% EXECUTING [4s]
> :wrapper
```



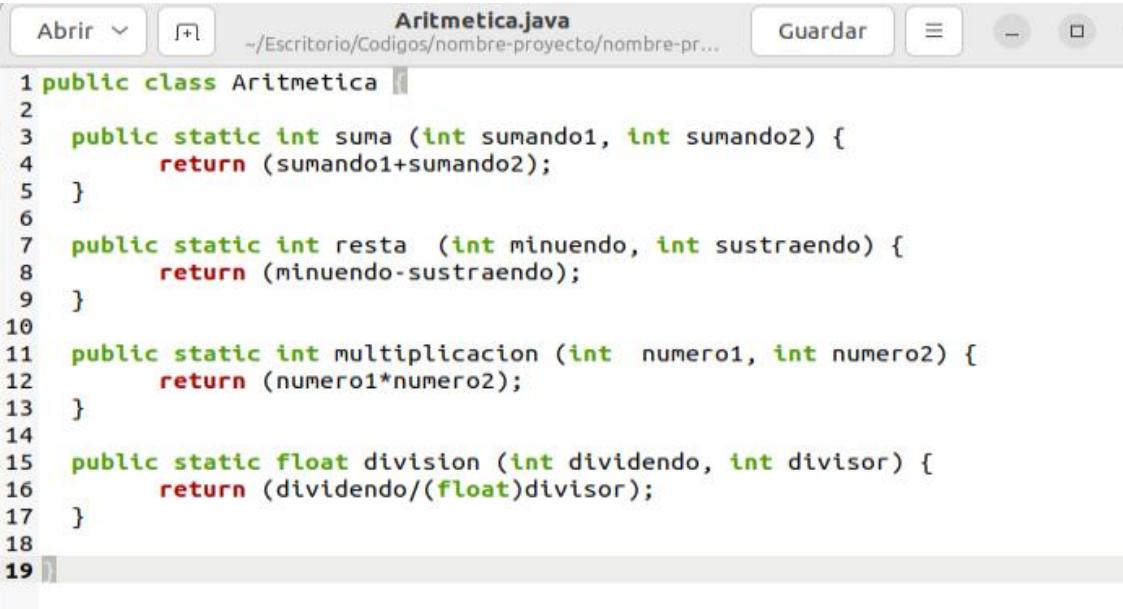
The screenshot shows a terminal window with the following command history:

```
david@david-VirtualBox: ~/Escritorio/Codigos/nombre-proyecto...$ mkdir nombre-proye
cto && cd nombre-proyecto
david@david-VirtualBox:~/Escritorio/Codigos/nombre-proyecto$ gradle init --type java-application
Support for nested build without a settings file was deprecated and will be removed in Gradle 5.0. You should create an empty settings file in /home/david/Escritorio/Codigos/nombre-proyecto/nombre-proyecto/gradle/wrapper
BUILD SUCCESSFUL in 2s
2 actionable tasks: 2 executed
```

- ⊕ Despues de borrar app.java y app.test creamos dentro de la ruta main/java lo siguiente:



```
Main.java
1 public class Main {
2
3     private static final int NUM1 = 5;
4     private static final int NUM2 = 2;
5
6
7     public static void main (String[] args) {
8         System.out.println ("Dados los números " + NUM1 + " y " + NUM2 );
9         System.out.println ("La suma es " + Aritmetica.suma(NUM1, NUM2) );
10        System.out.println ("La resta es " + Aritmetica.resta(NUM1, NUM2) );
11        System.out.println ("La multiplicación es " +
12            Aritmetica.multiplicacion(NUM1, NUM2) );
13        System.out.println ("La división es " + Aritmetica.division(NUM1,
14            NUM2) );
14    }
15 }
```

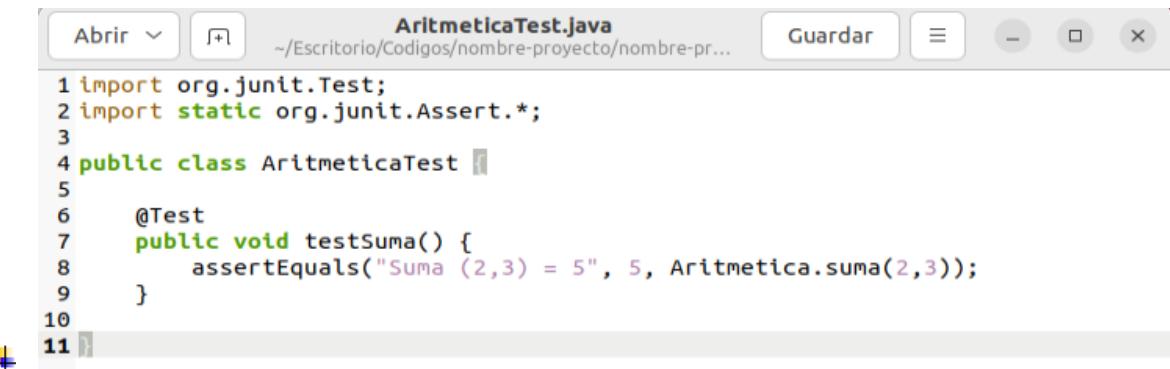


```
Aritmetica.java
1 public class Aritmetica {
2
3     public static int suma (int sumando1, int sumando2) {
4         return (sumando1+sumando2);
5     }
6
7     public static int resta (int minuendo, int sustraendo) {
8         return (minuendo-sustraendo);
9     }
10
11    public static int multiplicacion (int numero1, int numero2) {
12        return (numero1*numero2);
13    }
14
15    public static float division (int dividendo, int divisor) {
16        return (dividendo/(float)divisor);
17    }
18
19 }
```

⊕ Creamos dentro de la ruta test/java lo siguiente:



```
MainTest.java
1 import org.junit.Test;
2 import static org.junit.Assert.*;
3
4 public class MainTest {
5
6     @Test
7     public void testMain() {
8         // Prueba vacía
9     }
10 }
```



```
1 import org.junit.Test;
2 import static org.junit.Assert.*;
3
4 public class AritmeticaTest {
5
6     @Test
7     public void testSuma() {
8         assertEquals("Suma (2,3) = 5", 5, Aritmetica.suma(2,3));
9     }
10
11 }
```

✚ Editamos build.gradle



```
1 apply plugin: 'java'
2 apply plugin: 'application'
3
4 repositories {
5     jcenter()
6 }
7
8 dependencies {
9     compile 'com.google.guava:guava:23.0'
10    testCompile 'junit:junit:4.12'
11 }
12
13 jar {
14     manifest {
15         attributes ('Main-Class': 'Main')
16     }
17 }
18
19 mainClassName = 'Main'
```

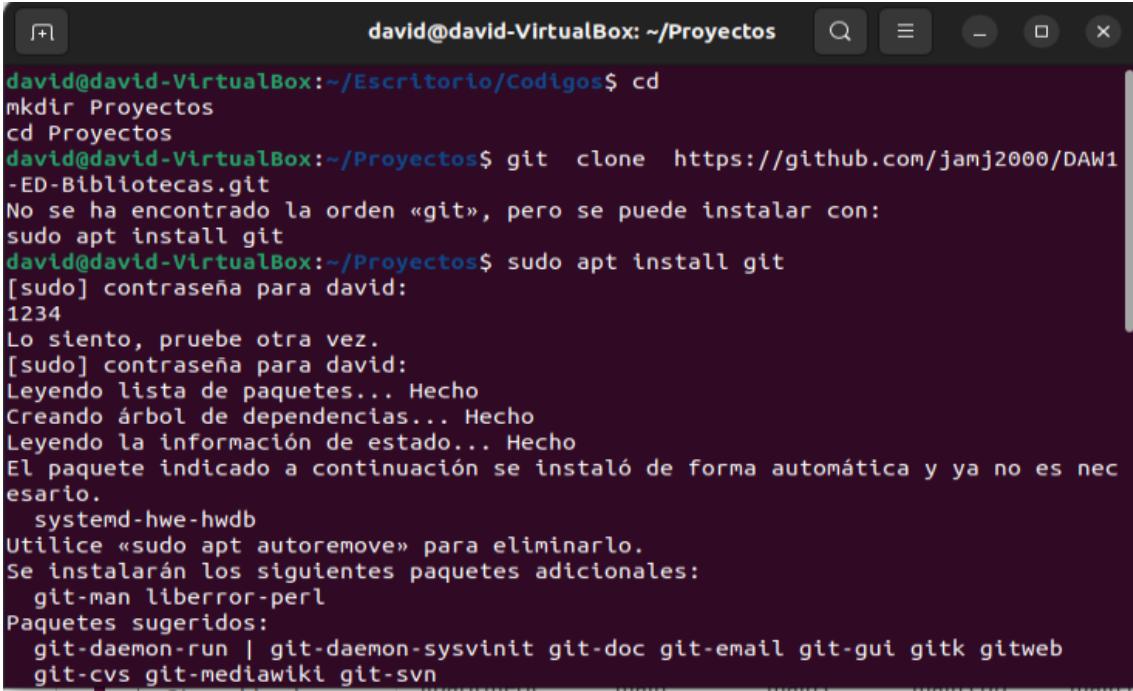
✚ Compilamos



```
david@david-VirtualBox: ~/Escritorio/Codigos/nombre-proy...$ ./gradlew assemble
Downloading https://services.gradle.org/distributions/gradle-4.4.1-bin.zip
Unzipping /home/david/.gradle/wrapper/dists/gradle-4.4.1-bin/46gopw3g8i1v3zqqx4q949t2x/gradle-4.4.1-bin.zip to /home/david/.gradle/wrapper/dists/gradle-4.4.1-bin/46gopw3g8i1v3zqqx4q949t2x
Set executable permissions for: /home/david/.gradle/wrapper/dists/gradle-4.4.1-bin/46gopw3g8i1v3zqqx4q949t2x/gradle-4.4.1/bin/gradle
```

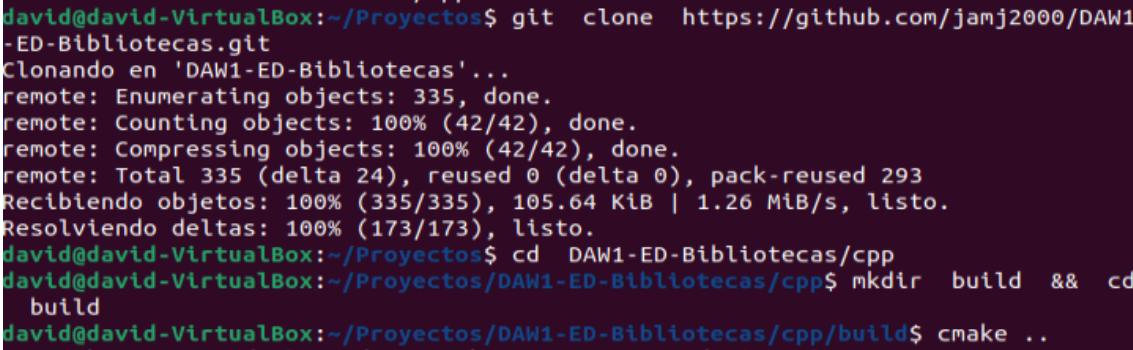
21. CMake. Automatiza el proceso de compilación de ejecutable y bibliotecas, su enlazado y la generación del archivo ejecutable para código fuente en C++. Crea un buildfile con CMake.

Creamos proyecto



```
david@david-VirtualBox:~/Escritorio/Codigos$ cd  
mkdir Proyectos  
cd Proyectos  
david@david-VirtualBox:~/Proyectos$ git clone https://github.com/jamj2000/DAW1-ED-Bibliotecas.git  
No se ha encontrado la orden «git», pero se puede instalar con:  
sudo apt install git  
david@david-VirtualBox:~/Proyectos$ sudo apt install git  
[sudo] contraseña para david:  
1234  
Lo siento, pruebe otra vez.  
[sudo] contraseña para david:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.  
    systemd-hwe-hwdb  
Utilice «sudo apt autoremove» para eliminarlo.  
Se instalarán los siguientes paquetes adicionales:  
    git-man liberror-perl  
Paquetes sugeridos:  
    git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb  
    git-cvs git-mediawiki git-svn
```

3. Descarga el código fuente y entra en el directorio DAW1-ED-Bibliotecas/cpp. Crea un directorio de construcción y entra en él.



```
david@david-VirtualBox:~/Proyectos$ git clone https://github.com/jamj2000/DAW1-ED-Bibliotecas.git  
Clonando en 'DAW1-ED-Bibliotecas'...  
remote: Enumerating objects: 335, done.  
remote: Counting objects: 100% (42/42), done.  
remote: Compressing objects: 100% (42/42), done.  
remote: Total 335 (delta 24), reused 0 (delta 0), pack-reused 293  
Recibiendo objetos: 100% (335/335), 105.64 KiB | 1.26 MiB/s, listo.  
Resolviendo deltas: 100% (173/173), listo.  
david@david-VirtualBox:~/Proyectos$ cd DAW1-ED-Bibliotecas/cpp  
david@david-VirtualBox:~/Proyectos/DAW1-ED-Bibliotecas/cpp$ mkdir build && cd  
build  
david@david-VirtualBox:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ cmake ..
```

```
CMake Deprecation Warning at CMakeLists.txt:1 (CMAKE_MINIMUM_REQUIRED):
  Compatibility with CMake < 2.8.12 will be removed from a future version of
  CMake.

  Update the VERSION argument <min> value or use a ...<max> suffix to tell
  CMake that the project does not need compatibility with older versions.

-- The C compiler identification is GNU 11.3.0
-- The CXX compiler identification is GNU 11.3.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
```

💡 Ejecutamos

```
david@david-VirtualBox:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ cmake .. -DCMAKE_INSTALL_PREFIX=/usr
CMake Deprecation Warning at CMakeLists.txt:1 (CMAKE_MINIMUM_REQUIRED):
  Compatibility with CMake < 2.8.12 will be removed from a future version of
  CMake.

  Update the VERSION argument <min> value or use a ...<max> suffix to tell
  CMake that the project does not need compatibility with older versions.

Resumen de construcción (build) para la aplicación.
-----
* Prefijo de instalación      : /usr
* Directorio del ejecutable   : /usr/bin
* Directorio de la biblioteca : /usr/lib
* Tipo de construcción (build): Release
* Plataforma                  : Linux-5.15.0-56-genericx86_64

-- Configuring done
-- Generating done
-- Build files have been written to: /home/david/Proyectos/DAW1-ED-Bibliotecas/cpp/build
```

💡 Comprobamos que se ha generado un archivo Makefile

```
david@david-VirtualBox:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ ls
cat Makefile
bin CMakeCache.txt CMakeFiles cmake_install.cmake lib Makefile
# CMAKE generated file: DO NOT EDIT!
# Generated by "Unix Makefiles" Generator, CMake Version 3.22

# Default target executed when no arguments are given to make.
default_target: all
.PHONY : default_target

# Allow only one "make -f Makefile2" at a time, but pass parallelism.
.NOTPARALLEL:

#=====
# Special targets provided by cmake.

# Disable implicit rules so canonical targets will work.
.SUFFIXES:
```

- ✚ Ahora podemos realizar el proceso de construcción con el comando make

```
david@david-VirtualBox:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ make
[ 16%] Building CXX object CMakeFiles/MyLibrary2.dir/MyLib2/potencias.cpp.o
[ 33%] Linking CXX static library lib/libMyLibrary2.a
[ 33%] Built target MyLibrary2
[ 50%] Building CXX object CMakeFiles/MyLibrary1.dir/MyLib1/aritmetica.cpp.o
[ 66%] Linking CXX shared library lib/libMyLibrary1.so
[ 66%] Built target MyLibrary1
[ 83%] Building CXX object CMakeFiles/MyApplication.dir/MyApp/main.cpp.o
[100%] Linking CXX executable bin/MyApplication
[100%] Built target MyApplication
```

- ✚ Instalamos archivos generados y listo.

```
david@david-VirtualBox:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ sudo make install
Consolidate compiler generated dependencies of target MyLibrary2
[ 33%] Built target MyLibrary2
Consolidate compiler generated dependencies of target MyLibrary1
[ 66%] Built target MyLibrary1
Consolidate compiler generated dependencies of target MyApplication
[100%] Built target MyApplication
Install the project...
-- Install configuration: "Release"
-- Installing: /usr/bin/MyApplication
-- Set runtime path of "/usr/bin/MyApplication" to "/usr/lib"
-- Installing: /usr/lib/libMyLibrary1.so
-- Installing: /usr/lib/libMyLibrary2.a
david@david-VirtualBox:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$
```