

# HERRAMIENTAS DE DESARROLLO

Enlace a GitHub:

[https://github.com/claralcala/ED\\_CARRILLO\\_22/tree/main/HerramientasDeDesarrollo](https://github.com/claralcala/ED_CARRILLO_22/tree/main/HerramientasDeDesarrollo)

Ejercicios finales  
primer trimestre ED

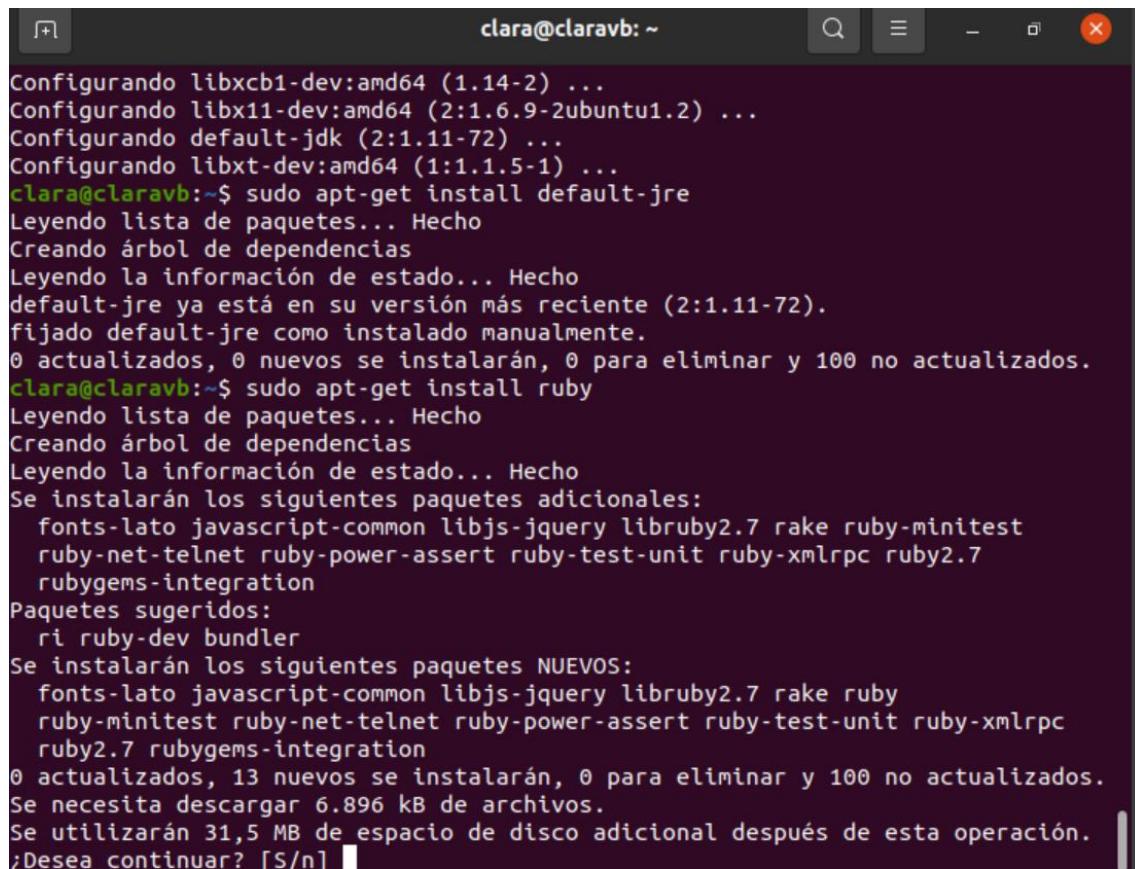
## **ÍNDICE**

Ejercicio 1 & Ejercicio 5.....	pg. 2
Ejercicio 2.....	pg. 9
Ejercicio 3.....	pg. 10
Ejercicio 4.....	pg. 10
Ejercicio 6.....	pg. 11
Ejercicio 7.....	pg. 11
Ejercicio 8.....	pg. 12
Ejercicio 9.....	pg. 13
Ejercicio 10.....	pg. 15
Ejercicio 11.....	pg. 15
Ejercicio 12.....	pg. 15
Ejercicio 13.....	pg. 16
Ejercicio 14.....	pg. 18
Ejercicio 15.....	pg. 21
Ejercicio 16.....	pg. 21
Ejercicio 17.....	pg. 21
Ejercicio 18.....	pg. 24
Ejercicio 19.....	pg. 27
Ejercicio 20.....	pg. 34
Ejercicio 21.....	pg. 38

## Ejercicio 1 y Ejercicio 5

Para empezar, instalamos los paquetes de los lenguajes de programación en Ubuntu. El proceso es el mismo para todos: en la terminal, con “sudo apt-get install” y el nombre del paquete. Por ejemplo “sudo apt-get install python”. Cuando nos pregunte si queremos continuar, pulsamos la S en el teclado.

En la captura vemos cómo se instala Ruby, por ejemplo.



A screenshot of a terminal window titled "clara@claravb: ~". The window shows the output of several "sudo apt-get install" commands. It starts with configurations for libxcb1-dev and libx11-dev. Then, it installs "default-jre" and checks for updates, finding none. Next, it installs "ruby" and lists additional packages to be installed: fonts-lato, javascript-common, libjs-jquery, libruby2.7, rake, ruby-minitest, ruby-net-telnet, ruby-power-assert, ruby-test-unit, ruby-xmlrpc, and ruby2.7. It then lists suggested packages: ri, ruby-dev, and bundler. Finally, it lists new packages: fonts-lato, javascript-common, libjs-jquery, libruby2.7, rake, ruby, ruby-minitest, ruby-net-telnet, ruby-power-assert, ruby-test-unit, ruby-xmlrpc, and ruby2.7. It concludes by stating there are 13 new packages to install, 0 to remove, and 100 not updated. It asks if the user wants to continue, with the cursor at the end of the question.

```
Configurando libxcb1-dev:amd64 (1.14-2) ...
Configurando libx11-dev:amd64 (2:1.6.9-2ubuntu1.2) ...
Configurando default-jdk (2:1.11-72) ...
Configurando libxt-dev:amd64 (1:1.1.5-1) ...
clara@claravb:~$ sudo apt-get install default-jre
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
default-jre ya está en su versión más reciente (2:1.11-72).
fijado default-jre como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 100 no actualizados.
clara@claravb:~$ sudo apt-get install ruby
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  fonts-lato javascript-common libjs-jquery libruby2.7 rake ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby-xmlrpc ruby2.7
  rubygems-integration
Paquetes sugeridos:
  ri ruby-dev bundler
Se instalarán los siguientes paquetes NUEVOS:
  fonts-lato javascript-common libjs-jquery libruby2.7 rake ruby
  ruby-minitest ruby-net-telnet ruby-power-assert ruby-test-unit ruby-xmlrpc
  ruby2.7 rubygems-integration
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 100 no actualizados.
Se necesita descargar 6.896 kB de archivos.
Se utilizarán 31,5 MB de espacio de disco adicional después de esta operación.
;Desea continuar? [S/n] 
```

## Bash

En la terminal escribimos gedit ./hello.sh. En el archivo que nos sale escribimos lo de la captura de abajo, guardamos y cerramos. Nos vamos de vuelta a la terminal.

```
1 #!/bin/bash
2 echo Hello World
```

sh ▾ Anchura del tabulador: 8 ▾ Ln 2, Col 17 ▾ INS

En la terminal, escribimos ./hello.sh. Lo normal es que nos diga que no tenemos permiso, para ello debemos escribir “chmod 764./hello.sh”. Volvemos a introducir “./hello.sh” y ahora sí nos devuelve nuestro Hello World.

```
clara@claravb:~$ gedit ./hello.sh
clara@claravb:~$ ./hello.sh
bash: ./hello.sh: Permiso denegado
clara@claravb:~$ ./hello.sh
bash: ./hello.sh: Permiso denegado
clara@claravb:~$ chmod 764 ./hello.sh
clara@claravb:~$ ./hello.sh
Hello World
clara@claravb:~$ S
```

## Python

En primer lugar, ejecutamos el intérprete escribiendo “python” en la terminal.

```
clara@claravb:~$ python
Python 2.7.18 (default, Jul 1 2022, 12:27:04)
[GCC 9.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Escribimos y ejecutamos lo que queremos que nos devuelva por pantalla. Ya sabemos que el intérprete funciona correctamente. Salimos con control+D

```
>>> print("Hola mundo")
Hola mundo
>>>
```

Creamos el documento hola.py (nosotros accedemos al editor de texto a través de la terminal con gedit).

```
1 #!/usr/bin/env python
2 print "Hola mundo"
```

Damos permisos para ejecutar y ejecutamos con "./hola.py"

```
clara@claravb:~$ chmod +x hola.py
clara@claravb:~$ ./hola.py
Hola mundo
clara@claravb:~$ 
```

## PHP

Comprobamos que el intérprete funciona

```
php > echo "hola mundo";
hola mundo
php >
```

Creamos un archivo con gedit y lo llamamos hola.php

```
1 #!/usr/bin/env php
2 <?php
3 echo "Hola mundo\n"
4 ?>
```

Damos permisos y ejecutamos

```
clara@claravb:~$ chmod +x hola.php
clara@claravb:~$ ./hola.php
Hola mundo
clara@claravb:~$
```

### Javascript

Comprobamos que el intérprete funciona:

```
clara@claravb:~$ node
> console.log('hola mundo');
hola mundo
```

Creamos y editamos el archivo hola.js.

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

Damos permiso y ejecutamos:

```
clara@claravb:~$ chmod +x hola.js
clara@claravb:~$ ./hola.js
Hola mundo
clara@claravb:~$
```

### C

Creamos y editamos el archivo hola.c

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

Compilamos, enlazamos y ejecutamos

```
clara@claravb:~$ gcc -o hola hola.c
clara@claravb:~$ ./hola
Hola mundo
clara@claravb:~$
```

## C++

Creamos y editamos el archivo hola.cpp:

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

Compilamos, enlazamos y ejecutamos:

```
clara@claravb:~$ g++ -o hola hola.cpp
clara@claravb:~$ ./hola
Hola mundo
clara@claravb:~$ █
```

## Java

Creamos y editamos el archivo Hola.java

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

Compilamos, interpretamos y ejecutamos:

```
clara@claravb:~$ javac Hola.java
clara@claravb:~$ java Hola
Hola mundo
clara@claravb:~$ █
```

Ejecutamos:

```
clara@claravb:~$ java -jar hola.jar
Hola mundo
clara@claravb:~$ █
```

### Ensamblador (nasm)

Creamos y editamos el archivo hola.asm

```
1 section .data
2 msg      db "Hola mundo", 0Ah
3 len      equ     $ - msg
4
5 section .text
6
7 global _start
8
9 _start:
10        mov      eax, 04h
11        mov      ebx, 01h
12        mov      ecx, msg
13        mov      edx, len
14        int      80h
15        mov      eax, 01h
16        mov      ebx, 00h
17        int      80h
```

Ensamblamos y enlazamos

```
clara@claravb:~$ nasm -f elf64 hola.asm
clara@claravb:~$ ld hola.o -o hola
```

Ejecutamos:

```
clara@claravb:~$ ./hola
Hola mundo
```

### Ruby

Comprobamos que el intérprete funciona:

```
clara@claravb:~$ ruby
puts "Hola mundo"
puts "hola mundo"

Hola mundo
hola mundo
```

Creamos y editamos el archivo hola.rb

---

```
1 #!/usr/bin/env ruby
2 puts "Hola mundo"
```

Damos permisos y ejecutamos:

```
clara@claravb:~$ chmod +x hola.rb
clara@claravb:~$ ./hola.rb
Hola mundo
clara@claravb:~$
```

Go

Creamos y editamos el archivo hola.go

---

```
1 package main
2 import "fmt"
3 func main() {
4     fmt.Println("Holaaa mundo")
5 }
```

Compilamos, enlazamos y ejecutamos

```
clara@claravb:~$ go build hola.go
clara@claravb:~$ ./hola
Holaaa mundo
clara@claravb:~$
```

Rust

Creamos y editamos el archivo hola.rs:

---

```
1 fn main() {
2     println!("Hola, mundoo!");
3 }
```

Compilamos, enlazamos y ejecutamos:

```
clara@claravb:~$ rustc hola.rs
clara@claravb:~$ ./hola
Hola, mundoo!
clara@claravb:~$
```

## Lisp

Comprobamos que el intérprete funciona:

```
clara@claravb:~$ clisp
 i i i i i i      00000    0      0000000    00000    00000
 I I I I I I I     8      8    8      8      8    0    8    8
 I \ `+' / I       8      8    8      8      8    8    8    8
 \ `--+' /         8      8    8      8      00000   80000
   -+|-
   |           8      8    8      8      8    8    8
-----+----- 00000    8000000  0008000   00000    8
Bienvenido a GNU CLISP 2.49.92 (2018-02-18) <http://clisp.org/>
Copyright (c) Bruno Haible, Michael Stoll 1992-1993
Copyright (c) Bruno Haible, Marcus Daniels 1994-1997
Copyright (c) Bruno Haible, Pierpaolo Bernardi, Sam Steingold 1998
Copyright (c) Bruno Haible, Sam Steingold 1999-2000
Copyright (c) Sam Steingold, Bruno Haible 2001-2018
Teclee :h y pulse Intro para ayuda contextual.

[1]> (format t "Hola mundo")
Hola mundo
```

Creamos y editamos el archivo hola.lisp:

---

```
1 #!/usr/bin/env clisp
2 (format t "Hola mundoo!")
3
```

Damos permisos y ejecutamos:

```
clara@claravb:~$ chmod +x hola.lisp
clara@claravb:~$ ./hola.lisp
Hola mundoo!
clara@claravb:~$
```

## Ejercicio 2

Bash: interpretado

Python: interpretado

PHP: interpretado

Javascript: interpretado

C: compilado

C++: compilado

Java: compilado e interpretado

Ensamblador (nasm): ¿?

Ruby: interpretado

Go: compilado

Rust: compilado

Lisp: compilado

### **Ejercicio 3**

Bash: bash

Python: python

PHP: php -a

Javascript: node

C: gcc

C++: g++

Java: java / java -jar

Ensamblador (nasm): nasm

Ruby: ruby

Go: go

Rust: rustc

Lisp: clisp

### **Ejercicio 4**

Bash: .sh

Python: .py

PHP: .php

Javascript: .js

C: .c

C++: .cpp

Java: .java

Ensamblador (nasm): .asm

Ruby: .rb

Go: .go

Rust: .rs

Lisp: .lisp

### Ejercicio 5

Hecho ya en el ejercicio 1 en las capturas

### Ejercicio 6

.obj / .o

### Ejercicio 7

Creamos el archivo datos.c y metemos el código:

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

Hacemos lo mismo con main.c y suma.c:

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

```
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 }
```

Ejecutamos los comandos en la terminal:

```
clara@claravb:~/entornos$ gcc -c main.c  
clara@claravb:~/entornos$ gcc -c datos.c  
clara@claravb:~/entornos$ gcc -c suma.c
```

Listo, en la carpeta ya tenemos los tres archivos .o:



datos.c



datos.o



main.c



main.o



suma.c



suma.o

### Ejercicio 8

Ponemos los comandos en la terminal:

```
clara@claravb:~/entornos$ gcc -o programa main.o datos.o suma.o  
clara@claravb:~/entornos$
```

Obtenemos el archivo de nombre programa:



## programa

Damos permisos, ejecutamos y funciona:

```
clara@claravb:~/entornos$ chmod +x programa.bin
clara@claravb:~/entornos$ ./programa.bin
Hola a todos y todas
18
clara@claravb:~/entornos$ █
```

### Ejercicio 9

Igual que antes, creamos los archivos con los nombres correspondientes y metemos el código:

```
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;
```

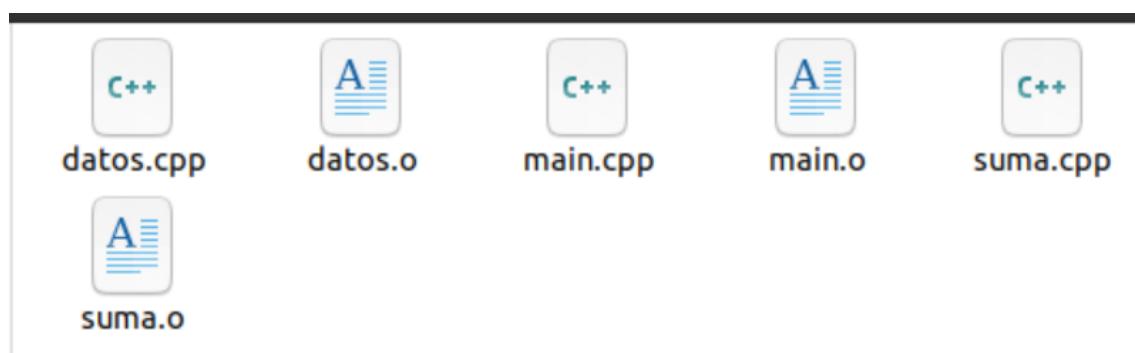
```
1 //-----  
2 // main.cpp  
3 //-----  
4 #include <iostream>  
5  
6 using namespace std;  
7  
8 int suma (int a, int b);  
9  
10 extern string mensaje;  
11 extern int num1, num2;  
12  
13 int main(){  
14     cout << mensaje << endl;  
15     cout << suma (num1, num2) << endl;  
16     return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

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

Nos vamos a la terminal:

```
clara@claravb:~/entornos2$ g++ -c main.cpp datos.cpp suma.cpp  
clara@claravb:~/entornos2$
```

Obtenemos los tres archivos .o:

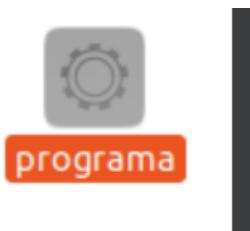


### Ejercicio 10

Metemos el comando en la terminal:

```
clara@claravb:~/entornos2$ g++ -o programa main.o datos.o suma.o  
clara@claravb:~/entornos2$
```

Obtenemos el archivo “programa”



Damos permisos y ejecutamos para comprobar que funciona

```
clara@claravb:~/entornos2$ chmod +x programa.bin  
clara@claravb:~/entornos2$ ./programa.bin  
Hola a todos y todas  
18  
clara@claravb:~/entornos2$
```

### Ejercicio 11

Las librerías son conjuntos de archivos de código que se utilizan para desarrollar software. Su objetivo es facilitar la programación, al proporcionar funcionalidades comunes que ya han sido resueltas previamente por otros programadores.

Las librerías son caseras o externas, dependiendo de si las creamos nosotros mismos u otros programadores, aunque la principal distinción se realiza entre librerías estáticas y librerías dinámicas. Veremos más sobre estos dos tipos en un ejercicio posterior.

### Ejercicio 12

En principio se usan más las dinámicas porque no se copian en el programa al compilarlas. Vamos a poner algunos ejemplos de librerías más usadas para varios lenguajes:

- Javascript: Jquery, Mootools, Moment.js, anime.js, ramda, D3.js, Chart.js, MathJS, Hammer.js, Glimmer.js, etc.

- C++: iostream, cmath, cstring,ctime, algorithm, etc.
- Python: matplotlib, seaborn, bokeh, Numpy, Scipy, Pandas, Numba, gensim, etc.

### Ejercicio 13

Creamos el archivo en C:

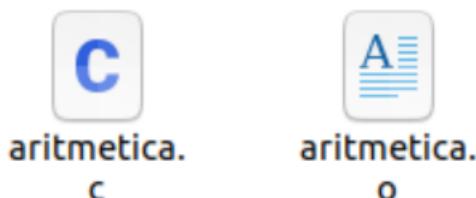
```

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

Compilamos y se nos genera el .o:

```

clara@claravb:~/bibliotecas$ gcc -c -fPIC aritmetica.c
clara@claravb:~/bibliotecas$
```



Empaquetamos:

```
clara@claravb:~/bibliotecas$ gcc -shared -fPIC -o libaritmetica.so aritmetica.o  
clara@claravb:~/bibliotecas$
```

Por último, instalamos la biblioteca en el sistema:

Para ello, necesitamos acceder a los permisos de root con sudo -s

```
clara@claravb:~/bibliotecas$ sudo -s  
[sudo] contraseña para clara:  
root@claravb:/home/clara/bibliotecas# cp libaritmetica.so /lib  
root@claravb:/home/clara/bibliotecas#
```

Ahora creamos y guardamos el archivo main.c:

```
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 }
```

Compilamos y enlazamos

```
clara@claravb:~/bibliotecas$ gcc -o main main.c libaritmetica.so  
clara@claravb:~/bibliotecas$
```

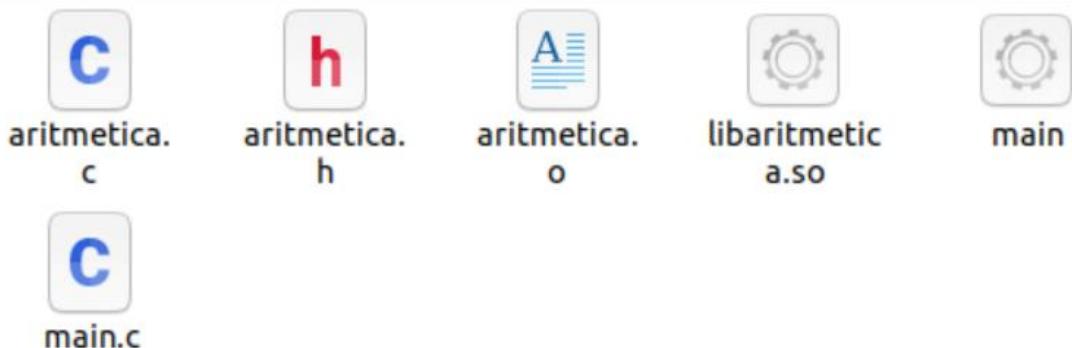
Comprobamos que la biblioteca nos sale como vínculo dinámico

```
clara@claravb:~/bibliotecas$ ldd main  
    linux-vdso.so.1 (0x00007fff1713a000)  
    libaritmetica.so => /lib/libaritmetica.so (0x00007f067d419000)  
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f067d227000)  
    /lib64/ld-linux-x86-64.so.2 (0x00007f067d436000)  
clara@claravb:~/bibliotecas$
```

Distribución →

```
clara@claravb:~/bibliotecas$ gcc -L. -Wl,-rpath=. -Wall -o main main.c -laritme  
tica  
clara@claravb:~/bibliotecas$
```

Comprobamos que tenemos todos los archivos en la carpeta bibliotecas



Damos permisos con chmod +x como siempre y luego ejecutamos para comprobar que el programa funciona bien:

```
clara@claravb:~/bibliotecas$ ./main.bin
Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.500000
clara@claravb:~/bibliotecas$
```

#### Ejercicio 14

Creamos el directorio aritmética:

```
clara@claravb:~$ mkdir aritmetica
clara@claravb:~$
```

Creamos y guardamos el archivo Aritmetica.java

```

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 }

```

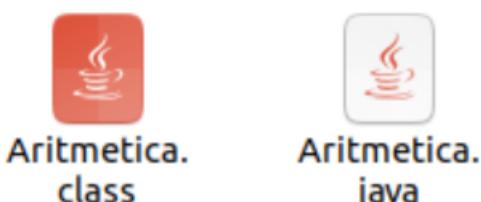
Compilamos

```

clara@claravb:~/aritmetica$ javac Aritmetica.java
clara@claravb:~/aritmetica$ 

```

A parte del .java que creamos antes, ya se nos ha generado un .class con el bytecode



Creamos el paquete jar

```

clara@claravb:~$ jar cvf aritmetica.jar aritmetica/*.class

```

Instalamos la biblioteca en el sistema. Al igual que nos pasó en el ejercicio anterior, no tenemos permisos, así que entramos en modo root

```

clara@claravb:~$ mv aritmetica.jar /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/aritm.jar
mv: no se puede mover 'aritmetica.jar' a '/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/aritm.jar': Permiso denegado
clara@claravb:~$ sudo -s
[sudo] contraseña para clara:
root@claravb:/home/clara# mv aritmetica.jar /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/aritm.jar
root@claravb:/home/clara#

```

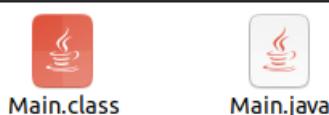
Creamos y guardamos el archivo Main.java:

```
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:

```
clara@clara-vb:~/aritmetica$ javac Main.java
clara@clara-vb:~/aritmetica$
```

Efectivamente, se nos ha generado el Main.java



Ahora nos volvemos a la terminal y ejecutamos:

```
clara@clara-vb:~/aritmetica$ 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
clara@clara-vb:~/aritmetica$
```

Vemos que busca la biblioteca que hemos creado y que nos devuelve los resultados de las operaciones correctos.

### Ejercicio 15

Las principales ventajas de las bibliotecas estáticas son que no es necesario proporcionar las bibliotecas correspondientes al publicar el programa y que, además, son de carga rápida.

Las principales desventajas son que las bibliotecas están empaquetadas en la aplicación, lo que significa que acabamos teniendo una biblioteca muy grande. Por otro lado, si la biblioteca cambia es necesario volver a compilar el programa.

### Ejercicio 16

Las bibliotecas dinámicas tienen varias ventajas, como que son idóneas para el desarrollo de aplicaciones a gran escala. También se ahorra más memoria. Es difícil definir desventajas para las librerías dinámicas.

### Ejercicio 17

Nos descargamos el código fuente:

```
clara@claravb:~/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 | 1.12 MiB/s, listo.
Resolviendo deltas: 100% (173/173), listo.
clara@claravb:~/Proyectos$
```

Entramos en este directorio, creamos otro directorio de build y nos metemos en él:

```
clara@claravb:~/Proyectos$ cd DAW1-ED-Bibliotecas/cpp
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp$ mkdir build && cd build
```

Creamos el archivo Makefile. Si nos dice que no podemos crearlo, simplemente instalamos cmake con sudo apt install

```

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ cmake ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done

Resumen de construcción (build) para la aplicación.
-----
* Prefijo de instalación      : /usr/local
* Directorio del ejecutable   : /usr/local/bin
* Directorio de la biblioteca : /usr/local/lib
* Tipo de construcción (build): Release
* Plataforma                  : Linux-5.15.0-53-genericx86_64

-- Configuring done
-- Generating done
-- Build files have been written to: /home/clara/Proyectos/DAW1-ED-Bibliotecas/
cpp/build
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ █

```

Ahí lo tenemos:



Construimos con el comando make:

```

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

```

Instalamos con sudo make install:

```

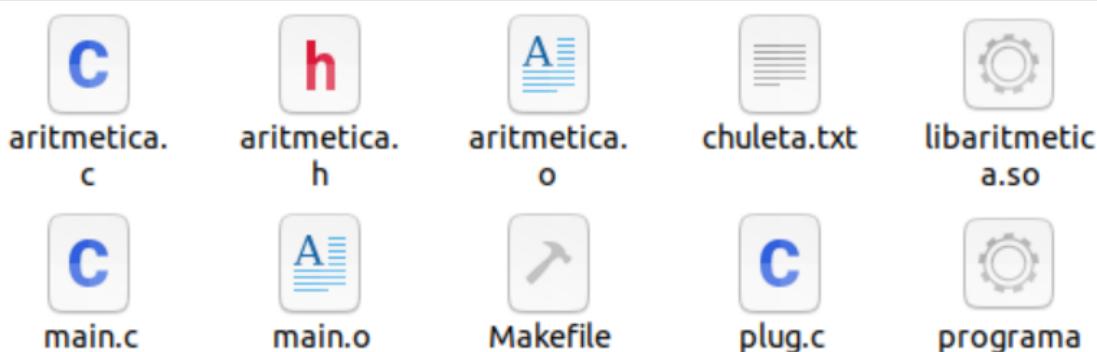
cmake_check_build_system:
    ${CMAKE_COMMAND} -S${CMAKE_SOURCE_DIR} -B${CMAKE_BINARY_DIR} --check-build-system CMakeFiles/Makefile.cmake 0
.PHONY : cmake_check_build_system

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ make
Scanning dependencies of target MyLibrary2
[ 16%] Building CXX object CMakeFiles/MyLibrary2.dir/MyLib2/potencias.cpp.o
[ 33%] Linking CXX static library lib/libMyLibrary2.a
[ 33%] Built target MyLibrary2
Scanning dependencies of target MyLibrary1
[ 50%] Building CXX object CMakeFiles/MyLibrary1.dir/MyLib1/aritmetica.cpp.o
[ 66%] Linking CXX shared library lib/libMyLibrary1.so
[ 66%] Built target MyLibrary1
Scanning dependencies of target MyApplication
[ 83%] Building CXX object CMakeFiles/MyApplication.dir/MyApp/main.cpp.o
[100%] Linking CXX executable bin/MyApplication
[100%] Built target MyApplication
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ sudo make install
[ 33%] Built target MyLibrary2
[ 66%] Built target MyLibrary1
[100%] Built target MyApplication
Install the project...
-- Install configuration: "Release"
-- Installing: /usr/local/bin/MyApplication
-- Set runtime path of "/usr/local/bin/MyApplication" to "/usr/local/lib"
-- Installing: /usr/local/lib/libMyLibrary1.so
-- Installing: /usr/local/lib/libMyLibrary2.a
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$
```

Nos vamos a la carpeta dedicada a C, ejecutamos make y se nos genera todo, incluido el ejecutable “programa”

```

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/c$ make
gcc -O -c main.c
gcc -O -c -fPIC aritmetica.c
gcc -O -shared -fPIC -o libaritmetica.so aritmetica.o
gcc -O -Wl,-rpath=/usr/local/lib main.o libaritmetica.so -o programa
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/c$
```



Ejecutamos programa (tras previamente haberle cambiado el nombre a programa.bin para que nos detecte la extensión del archivo) vemos que se ejecuta y nos devuelve las operaciones:

```

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/c$ ./programa.bin
Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.500000
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/c$ █

```

## Ejercicio 18

Creamos y guardamos el archivo build.xml

```

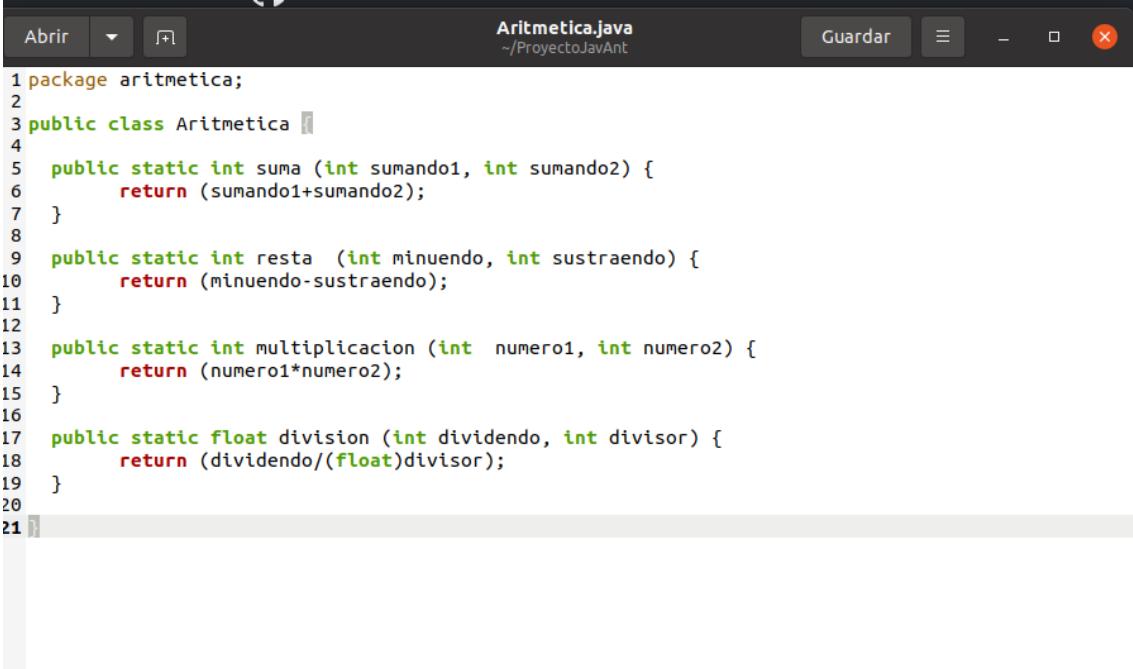
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project name="programa" default="jar" basedir=".">
3   <description>Programa que usa biblioteca aritmetica.</description>
4   <!-- Ejemplo de archivo de construcción (buildfile)
5     Para crear este archivo se ha consultado:
6       https://ant.apache.org/manual/tutorial-HelloworldWithAnt.html
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="aritmetica" />
26  </path>
27
28
29  <!-- Compilamos -->
30  <target name="compile" depends="init" >
31    <javac srcdir="${src.dir}" destdir="${classes.dir}" includeantruntime="false"
32      debug="true" >
33      <classpath refid="compile.classpath"/>
34    </javac>
35  </target>
36
37  <!-- Creamos archivo .jar -->

```

Creamos el 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 " + Aritmetica.multiplicacion(NUM1, NUM2) );
14         System.out.println ("La división es " + Aritmetica.division(NUM1, NUM2) );
15     }
16
17
18 }
```

Así como Aritmetica.java



A screenshot of a Java code editor window titled "Aritmetica.java". The window has a dark theme with white text. The code is as follows:

```
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 }
```

Introducimos el comando ant para crear nuestro proyecto. Si no tenemos instalado ant, la propia terminal de Ubuntu nos avisará y podemos instalarlo desde la misma terminal con sudo apt install ant

```

clara@clara-vb:~/ProyectoJavAnt$ ant
Buildfile: /home/clara/ProyectoJavAnt/build.xml

init:
    [mkdir] Created dir: /home/clara/ProyectoJavAnt/build/classes
    [mkdir] Created dir: /home/clara/ProyectoJavAnt/build/jar

compile:
    [javac] Compiling 2 source files to /home/clara/ProyectoJavAnt/build/classes

jar:
    [jar] Building jar: /home/clara/ProyectoJavAnt/build/jar/programa.jar

BUILD SUCCESSFUL
Total time: 2 seconds
clara@clara-vb:~/ProyectoJavAnt$ 

```

Para que la build no nos falle, tenemos que tener el directorio del proyecto tal que así (ignorando la carpeta build, pues esa es la que se genera al ejecutar ant desde la terminal):



Si ejecutamos el programa con el comando ant run, vemos que se ejecuta perfectamente:

```

clara@clara-vb:~/ProyectoJavAnt$ ant run
Buildfile: /home/clara/ProyectoJavAnt/build.xml

init:
compile:
jar:
run:
    [java] Dados los números 5 y 2
    [java] La suma es 7
    [java] La resta es 3
    [java] La multiplicación es 10
    [java] La división es 2.5

BUILD SUCCESSFUL
Total time: 1 second
clara@clara-vb:~/ProyectoJavAnt$ 

```

## Ejercicio 19

Creamos nueva carpeta llamada ProyectoJavaMaven, la podemos crear desde la terminal o manualmente. Nosotros la creamos manualmente.



Intentamos crear el proyecto con el arquetipo quickstart pero obviamente no nos deja porque no tenemos instalado Maven. No es un problema porque la propia terminal nos da las instrucciones para instalarlo con sudo apt install Maven

```
clara@clara-vb:~/ProyectoJavaMaven$ mvn archetype:generate -DgroupId=com.miempresta.app -DartifactId=mi-app -Dversion=1.0.0 \ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
No se ha encontrado la orden «mvn», pero se puede instalar con:
sudo apt install maven
clara@clara-vb:~/ProyectoJavaMaven$
```

Ahora sí, creamos nuestro proyecto:

```
Choose a number or apply filter (format: [groupId]:[artifactId], case sensitive contains): 1992: DarchetypeArtifFactId=maven-archetype-quickstart
Choose archetype:
  Your filter doesn't match any archetype (hint: enter to return to initial list)
Choose a number or apply filter (format: [groupId]:[artifactId], case sensitive contains): : maven-archetype-quickstart
Choose archetype:
1: remote -> com.github.ywching:maven-archetype-quickstart (Provide up-to-date java quickstart archetype)
2: remote -> org.apache.maven.archetypes:maven-archetype-quickstart (A simple Maven archetype)
3: remote -> org.apache.maven.archetypes:maven-archetype-quickstart (An archetype which contains a sample Maven project.)
Choose a number or apply filter (format: [groupId]:[artifactId], case sensitive contains): 3: 3
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8: 8
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.pom (1.6 kB at 3.3 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom (4.5 kB at 6.3 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 5.4 kB/s)
[INFO] Using property: groupId = com.miempresta.app
[INFO] Using property: artifactId = mi-app
[INFO] Using property: version = 1.0.0
[INFO] Using property: package = com.miempresta.app
Confirm properties configuration:
groupId: com.miempresta.app
artifactId: mi-app
version: 1.0.0
package: com.miempresta.app
Y: :
```

```
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: com.miempresta.app
[INFO] Parameter: artifactId, Value: mi-app
[INFO] Parameter: version, Value: 1.0.0
[INFO] Parameter: package, Value: com.miempresta.app
[INFO] Parameter: packageInPathFormat, Value: com/miempresa/app
[INFO] Parameter: package, Value: com.miempresta.app
[INFO] Parameter: version, Value: 1.0.0
[INFO] Parameter: groupId, Value: com.miempresta.app
[INFO] Parameter: artifactId, Value: mi-app
[INFO] Project created from Archetype in dir: /home/clara/ProyectoJavaMaven/mi-app
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 06:17 min
[INFO] Finished at: 2022-12-13T10:13:10+01:00
[INFO] -----
clara@clara-vb:~/ProyectoJavaMaven$
```

En la carpeta mi-app se nos ha generado un pom con la configuración básica:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5
6     <groupId>com.miempresa.app</groupId>
7     <artifactId>mi-app</artifactId>
8     <version>1.0.0</version>
9
10    <name>mi-app</name>
11    <!-- FIXME change it to the project's website -->
12    <url>http://www.example.com</url>
13
14    <properties>
15      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
16      <maven.compiler.source>1.7</maven.compiler.source>
17      <maven.compiler.target>1.7</maven.compiler.target>
18    </properties>
19
20    <dependencies>
21      <dependency>
22        <groupId>junit</groupId>
23        <artifactId>junit</artifactId>
24        <version>4.11</version>
25        <scope>test</scope>
26      </dependency>
27    </dependencies>
28
29    <build>
30      <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
31      <plugins>
32        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_Lifecycle -->
33        <plugin>
34          <artifactId>maven-clean-plugin</artifactId>
35          <version>3.1.0</version>
36        </plugin>
37        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin_bindings_for_jar_packaging -->
38        <plugin>
39          <artifactId>maven-resources-plugin</artifactId>
40          <version>3.0.2</version>
41        </plugin>
42        <plugin>
43          <artifactId>maven-compiler-plugin</artifactId>
44          <version>3.8.0</version>
45        </plugin>
46        <plugin>
47          <artifactId>maven-surefire-plugin</artifactId>
48        </plugin>
49      </plugins>
50    </pluginManagement>
51  </build>
52
53  <dependency>
54    <scope>test</scope>
55  </dependency>
56 </dependencies>
57
58 <build>
59   <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
60   <plugins>
61     <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_Lifecycle -->
62     <plugin>
63       <artifactId>maven-clean-plugin</artifactId>
64       <version>3.1.0</version>
65     </plugin>
66     <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin_bindings_for_jar_packaging -->
67     <plugin>
68       <artifactId>maven-resources-plugin</artifactId>
69       <version>3.0.2</version>
70     </plugin>
71     <plugin>
72       <artifactId>maven-compiler-plugin</artifactId>
73       <version>3.8.0</version>
74     </plugin>
75     <plugin>
76       <artifactId>maven-surefire-plugin</artifactId>
77       <version>2.22.1</version>
78     </plugin>
79     <plugin>
80       <artifactId>maven-jar-plugin</artifactId>
81       <version>3.0.2</version>
82     </plugin>
83     <plugin>
84       <artifactId>maven-install-plugin</artifactId>
85       <version>2.5.2</version>
86     </plugin>
87     <plugin>
88       <artifactId>maven-deploy-plugin</artifactId>
89       <version>2.8.2</version>
90     </plugin>
91     <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site_Lifecycle -->
92     <plugin>
93       <artifactId>maven-site-plugin</artifactId>
94       <version>3.7.1</version>
95     </plugin>
96     <plugin>
97       <artifactId>maven-project-info-reports-plugin</artifactId>
98       <version>3.0.0</version>
99     </plugin>
100   </plugins>
101 </pluginManagement>
102 </build>
103 </project>

```

Ahora nos vamos a la ruta src>main>java>com>miempresa>app y ahí vamos a editar el siguiente archivo App.java:

Por defecto nos viene un Hello world, pero nosotros lo cambiamos:

```
1 package com.miempresa.app;
2
3 /**
4  * Hello world!
5  *
6 */
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12    }
13 }
```

Este es el archivo modificado:

```
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 ("Datos 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 " + Aritmetica.multiplicacion(NUM1, NUM2) );
14         System.out.println ("La división es " + Aritmetica.division(NUM1, NUM2) );
15     }
16 }
```

Dentro de ese mismo directorio vamos a crear otro archivo, esta vez será Aritmetica.java

```
1 package com.miempresa.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 }
```

Ahora nos vamos a la ruta src>test>java>com>miempresa>app y vamos a crear otras dos clases, AppTest.java y AritmeticaTest.java:

```
1 package com.miempresa.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 }
```

```
1 package com.miempresa.app;
2
3 import org.junit.Test;
4 import static org.junit.Assert.*;
5
6 public class AritmeticaTest {
7
8     @Test
9     public void testSuma() {
10         assertEquals("Suma (2,3) = 5", 5, Aritmetica.suma(2,3));
11     }
12 }
```

Editamos el pom básico que teníamos:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project>
3
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>com.miempresta.app</groupId>
6   <artifactId>mi-app</artifactId>
7   <version>1.0.0</version>
8   <name>mi-app</name>
9
10  <build>
11    <plugins>
12      <plugin>
13          <!-- Para construir un JAR ejecutable -->
14          <groupId>org.apache.maven.plugins</groupId>
15          <artifactId>maven-jar-plugin</artifactId>
16          <version>3.0.2</version>
17          <configuration>
18              <archive>
19                  <manifest>
20                      <addClasspath>true</addClasspath>
21                      <classpathPrefix>./</classpathPrefix>
22                      <mainClass>com.miempresta.app.App</mainClass>
23                  </manifest>
24              </archive>
25          </configuration>
26      </plugin>
27
28      <plugin>
29          <!-- Para ejecutar el JAR creado -->
30          <groupId>org.codehaus.mojo</groupId>
31          <artifactId>exec-maven-plugin</artifactId>
32          <version>1.2.1</version>
33          <configuration>
34              <mainClass>com.miempresta.app.App</mainClass>
35          </configuration>
36      </plugin>
37  </plugins>
38 </build>
39
40 <dependencies>
41   <dependency>
42       <!-- Prueba de unidades -->
43       <groupId>junit</groupId>
44       <artifactId>junit</artifactId>
45       <version>4.12</version>
46       <scope>test</scope>
47   </dependency>
48 </dependencies>
```

```

<plugin>
    <!-- Para ejecutar el JAR creado -->
    <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>1.2.1</version>
        <configuration>
            <mainClass>com.miempresa.app.App</mainClass>
        </configuration>
    </plugin>
</plugins>
</build>

<dependencies>
    <dependency>
        <!-- Prueba de unidades -->
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
        <scope>test</scope>
    </dependency>
</dependencies>

</project>

```

Abrimos la terminal en la ruta del pom y compilamos con mvn compile:

Como es la primera vez que compilamos en Ubuntu, se nos descargan los recursos de la web de Apache Maven y luego la build nos sale correcta:

```

[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 2 source files to /home/clara/ProyectoJavaMaven/mi-app/target/classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  04:46 min
[INFO] Finished at: 2022-12-13T10:35:57+01:00
[INFO] -----
clara@clara-vb:~/ProyectoJavaMaven/mi-app$ 

```

Se nos ha generado una carpeta target y desde ahí ya podemos ejecutar el bytecode de java.

```

lara@clara-vb:~/ProyectoJavaMaven/mi-app$ cd target/classes && java com.miempresa.app.App && cd ../..
ados los números 5 y 2
la suma es 7
la resta es 3
la multiplicación es 10
la división es 2.5
lara@clara-vb:~/ProyectoJavaMaven/mi-app$ 

```

Pero continuamos, porque ahora nos tenemos que volver a la carpeta donde está el pom y ejecutar mvn package para generar el jar:

```

[INFO] Building jar: /home/clara/ProyectoJavaMaven/mi-app/target/mi-app-1.0.0.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  46.654 s
[INFO] Finished at: 2022-12-13T10:42:05+01:00
[INFO] -----
clara@clara-vb:~/ProyectoJavaMaven/mi-app$ 

```

Ahora ejecutamos con mvn exec:java

```

clara@clara-vb:~/ProyectoJavaMaven/mi-app$ mvn exec:java
[INFO] Scanning for projects...
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/no/execution-maven-plugin/1.2.1/execution-maven-plugin-1.2.1.pom (7.7 kB at 1.4 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/no/execution-maven-plugin/1.2.1/execution-maven-plugin-1.2.1.pom (7.7 kB at 1.4 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/no/no-parent/28/no-parent-28.pom (26 kB at 9.5 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/no/no-parent/28/no-parent-28.pom (26 kB at 9.5 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/no/no-parent/3/no-parent-3.pom (4.1 kB at 0.6 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/no/execution-maven-plugin/1.2.1/execution-maven-plugin-1.2.1.jar (38 kB at 18 kB/s)
[INFO]
[INFO] -----> com.mtempresa.app:mi-app >-----
[INFO] Building mi-app 1.0.0
[INFO] -----[ jar ]-----
[INFO] >>> exec-maven-plugin:1.2.1:java (default-cli) > validate @ mi-app >>>
[INFO] <<< exec-maven-plugin:1.2.1:java (default-cli) < validate @ mi-app <<<
[INFO]
[INFO]
[INFO] *** exec-maven-plugin:1.2.1:java (default-cli) @ mi-app ***
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0/maven-plugin-api-2.0.pom (601 B at 549 B/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0/maven-plugin-api-2.0.pom (601 B at 549 B/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0/maven-2.0.pom (8.8 kB at 3.2 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0/maven-2.0.pom (8.8 kB at 3.2 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons/commons-exec/1.1/commons-exec-1.1.pom (11 kB at 8.9 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons/commons-parent/17/commons-parent-17.pom (31 kB at 19 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons/commons-parent/17/commons-parent-17.pom (31 kB at 19 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.jar (53 kB at 16 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.jar (53 kB at 16 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9/plexus-container-default-1.0-alpha-9.jar (195 kB at 35 kB/s)
Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.5
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 26.528 s
[INFO] Finished at: 2022-12-13T10:43:20+01:00
[INFO] -----
clara@clara-vb:~/ProyectoJavaMaven/mi-app$ 

```

```

Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.5
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 26.528 s
[INFO] Finished at: 2022-12-13T10:43:20+01:00
[INFO] -----
clara@clara-vb:~/ProyectoJavaMaven/mi-app$ 

```

Al igual que en Windows, podemos ejecutar también con java -jar:

```

clara@clara-vb:~/ProyectoJavaMaven/mi-app$ java -jar target/mi-app-1.0.0.jar
Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.5
clara@clara-vb:~/ProyectoJavaMaven/mi-app$ 

```

Por último, realizamos las pruebas con mvn test

```

clara@clara-vb:~/ProyectoJavaMaven/mi-app$ mvn test
[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/clara/ProyectoJavaMaven/mi-app/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mi-app ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ 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/clara/ProyectoJavaMaven/mi-app/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ mi-app ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ mi-app ---
[INFO] Surefire report directory: /home/clara/ProyectoJavaMaven/mi-app/target/surefire-reports

-----
T E S T S
-----
Running com.miempresa.app.AritmeticaTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.146 sec
Running com.miempresa.app.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  4.728 s
[INFO] Finished at: 2022-12-13T10:45:09+01:00
[INFO] -----
clara@clara-vb:~/ProyectoJavaMaven/mi-app$ 

```

No obtenemos ningún error, solo un par de Warning, así que todo perfecto. Ya hemos completado nuestro proyecto con Maven.

## Ejercicio 20

Para este proyecto vamos a crear otra carpeta llamada ProyectoJavaGradle



Dentro de esta carpeta creamos otra llamada miapp. Esta vez la creamos desde la terminal, pero da lo mismo que crearla manualmente. Entramos en esa carpeta:

```

clara@clara-vb:~/ProyectoJavaGradle$ mkdir miapp && cd miapp
clara@clara-vb:~/ProyectoJavaGradle/miapp$ 

```

Intentamos generar el proyecto de Gradle y no nos deja porque no lo tenemos instalado. No es problema, lo instalamos con las instrucciones que nos da la terminal. Nosotros instalamos la versión 4.4.1-10 con sudo apt install gradle:

```
clara@clara-vb:~/ProyectoJavaGradle$ mkdir miapp && cd miapp
clara@clara-vb:~/ProyectoJavaGradle/miapp$ gradle init --type java-application

No se ha encontrado la orden «gradle», pero se puede instalar con:

sudo snap install gradle # version 7.2, or
sudo apt install gradle # version 4.4.1-10

Consulte «snap info gradle» para ver más versiones.

clara@clara-vb:~/ProyectoJavaGradle/miapp$
```

Ahora sí, generamos la estructura de carpetas del proyecto:

```
clara@clara-vb:~/ProyectoJavaGradle/miapp$ gradle init --type java-application
Starting a Gradle Daemon (subsequent builds will be faster)

BUILD SUCCESSFUL in 10s
2 actionable tasks: 2 executed
clara@clara-vb:~/ProyectoJavaGradle/miapp$
```

Borramos las clases que nos vienen por defecto.

```
clara@clara-vb:~/ProyectoJavaGradle/miapp$ rm src/main/java/App.java
clara@clara-vb:~/ProyectoJavaGradle/miapp$ rm src/test/java/AppTest.java
clara@clara-vb:~/ProyectoJavaGradle/miapp$
```

Creamos dos clases en src/main/java:

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 " + Aritmetica.multiplicacion(NUM1, NUM2) );
12        System.out.println ("La división es " + Aritmetica.division(NUM1, NUM2) );
13    }
14}
```

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 }
```

Ahora nos vamos a src>test>java y creamos otras dos clases:

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 }
```

AritmeticaTest.java:

```
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 el archivo build.gradle para que quede así:

```

9 // Apply the java plugin to add support for Java
10 apply plugin: 'java'
11 apply plugin: 'application'
12
13 repositories {
14     jcenter()
15 }
16
17 dependencies {
18     compile 'com.google.guava:guava:23.0'
19     testCompile 'junit:junit:4.12'
20 }
21
22 jar {
23     manifest {
24         attributes ('Main-Class': 'Main')
25     }
26 }
27
28 mainClassName = 'Main'

```

Compilamos con ./gradlew assemble

```

clara@clara-vb:~/ProyectoJavaGradle/miapp$ ./gradlew assemble
Downloading https://services.gradle.org/distributions/gradle-4.4-i-bin.zip
Unzipping /home/clara/.gradle/wrapper/dists/gradle-4.4-i-bin/46gopw3g8l1v3zqqx4q949t2x/gradle-4.4-i-bin.zip to /home/clara/.gradle/wrapper/dists/gradle-4.4-i-bin/46gopw3g8l1v3zqqx4q949t2x/gradle-4.4.1/bin/gradle
Set executable permissions for: /home/clara/.gradle/wrapper/dists/gradle-4.4.1-bin/46gopw3g8l1v3zqqx4q949t2x/gradle-4.4.1/bin/gradle
Download https://jcenter.bintray.com/com/google/guava/guava/23.0/guava-23.0.pom
Download https://jcenter.bintray.com/com/google/guava/guava-parent/23.0/guava-parent-23.0.pom
Download https://jcenter.bintray.com/com/sonatype/oss/oss-parent/7/oss-parent-7.pom
Download https://jcenter.bintray.com/com/google/code/findbugs/jsr305/1.9/jsr305-1.9.pom
Download https://jcenter.bintray.com/com/google/errorprone/error_prone_annotations/2.0.18/error_prone_annotations-2.0.18.pom
Download https://jcenter.bintray.com/org/codhaus/mojo/animal-sniffer-annotations/1.14/animal-sniffer-annotations-1.14.pom
Download https://jcenter.bintray.com/org/codhaus/mojo/animal-sniffer-parent/1.14/animal-sniffer-parent-1.14.pom
Download https://jcenter.bintray.com/org/codhaus/mojo/mojo-parent/34/mojo-parent-34.pom
Download https://jcenter.bintray.com/org/codhaus/codehaus-parent/4/codehaus-parent-4.pom
Download https://jcenter.bintray.com/com/google/errorprone/error_prone_annotations/2.0.18/error_prone_annotations-2.0.18.pom
Download https://jcenter.bintray.com/com/google/errorprone/error_prone_parent/2.0.18/error_prone_parent-2.0.18.pom
Download https://jcenter.bintray.com/com/google/guava/guava/23.0/guava-23.0.jar
Download https://jcenter.bintray.com/com/google/errorprone/error_prone_annotations/2.0.18/error_prone_annotations-2.0.18.jar
Download https://jcenter.bintray.com/com/google/j2objc/j2objc-annotations/1.1/j2objc-annotations-1.1.jar
Download https://jcenter.bintray.com/org/codhaus/mojo/animal-sniffer-annotations/1.14/animal-sniffer-annotations-1.14.jar
Download https://jcenter.bintray.com/com/google/code/findbugs/jsr305/1.3.9/jsr305-1.3.9.jar
BUILD SUCCESSFUL in 2s
5 actionable tasks: 5 executed
clara@clara-vb:~/ProyectoJavaGradle/miapp$ 

```

El jar se nos ha generado en el directorio build>libs. Lo ejecutamos con java -jar miapp.jar

```

clara@clara-vb:~/ProyectoJavaGradle/miapp/build/libs$ java -jar miapp.jar
Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.5
clara@clara-vb:~/ProyectoJavaGradle/miapp/build/libs$ 

```

Podemos hacerlo con ./gradlew run también

```

clara@clara-vb:~/ProyectoJavaGradle/miapp$ ./gradlew run

> Task :run
Dados los números 5 y 2
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.5

BUILD SUCCESSFUL in 2s
2 actionable tasks: 1 executed, 1 up-to-date
clara@clara-vb:~/ProyectoJavaGradle/miapp$ 

```

Ejecutamos los test con ./gradlew test

```
clara@clara-vb:~/ProyectoJavaGradle/miapp$ ./gradlew test

BUILD SUCCESSFUL in 10s
3 actionable tasks: 2 executed, 1 up-to-date
clara@clara-vb:~/ProyectoJavaGradle/miapp$
```

Abrimos el archivo marcado con el navegador de internet (Firefox en nuestro caso, pero da igual)



Ahí lo tenemos, sale todo 100% correcto, por lo que ya hemos completado nuestro proyecto en Gradle:

### Test Summary

2	0	0	0.059s		
tests	failures	ignored	duration		
100%			successful		
Generated by Gradle 4.4.1 at 13-dic-2022 11:14:30					
<a href="#">Packages</a> <a href="#">Classes</a>					
Package	Tests	Failures	Ignored	Duration	Success rate
<a href="#">default-package</a>	2	0	0	0.059s	100%

### Ejercicio 21

Los pasos previos los tenemos ya conseguidos de cuando hicimos el ejercicio de C, así que nos vamos a la carpeta de CPP y ejecutamos cmake para generar el Makefile

```

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ cmake ..
Resumen de construcción (build) para la aplicación.
-----
* Prefijo de instalación      : /usr/local
* Directorio del ejecutable  : /usr/local/bin
* Directorio de la biblioteca : /usr/local/lib
* Tipo de construcción (build): Release
* Plataforma                  : Linux-5.15.0-53-genericx86_64

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

```

Comprobamos que se nos ha generado:

```

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ cat Makefile
# CMAKE generated file: DO NOT EDIT!
# Generated by "Unix Makefiles" Generator, CMake Version 3.16

# 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:

```

Generamos con make e instalamos con sudo make install

```

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ make
[ 33%] Built target MyLibrary2
[ 66%] Built target MyLibrary1
[100%] Built target MyApplication
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ █

```

```

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ sudo make install
[sudo] contraseña para clara:
Lo siento, pruebe otra vez.
[sudo] contraseña para clara:
[ 33%] Built target MyLibrary2
[ 66%] Built target MyLibrary1
[100%] Built target MyApplication
Install the project...
-- Install configuration: "Release"
-- Up-to-date: /usr/local/bin/MyApplication
-- Up-to-date: /usr/local/lib/libMyLibrary1.so
-- Up-to-date: /usr/local/lib/libMyLibrary2.a
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build$ █

```

Si nos vamos a CPP>build>bin vemos que tenemos un archivo llamado MyApplication. Lo renombramos para que tenga la extensión .bin y lo podemos ejecutar desde la terminal:



MyApplication.bin

Abrimos la terminal en este mismo directorio, ejecutamos y observamos que nos devuelve los resultados de las operaciones correctamente:

```
clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build/bin$ ./MyApplication.bin
Dados los números 5 y 2
-----
La suma es 7
La resta es 3
La multiplicación es 10
La división es 2.5
-----
El cuadrado de 5 es 25
El cubo de 5 es 125

clara@claravb:~/Proyectos/DAW1-ED-Bibliotecas/cpp/build/bin$
```