

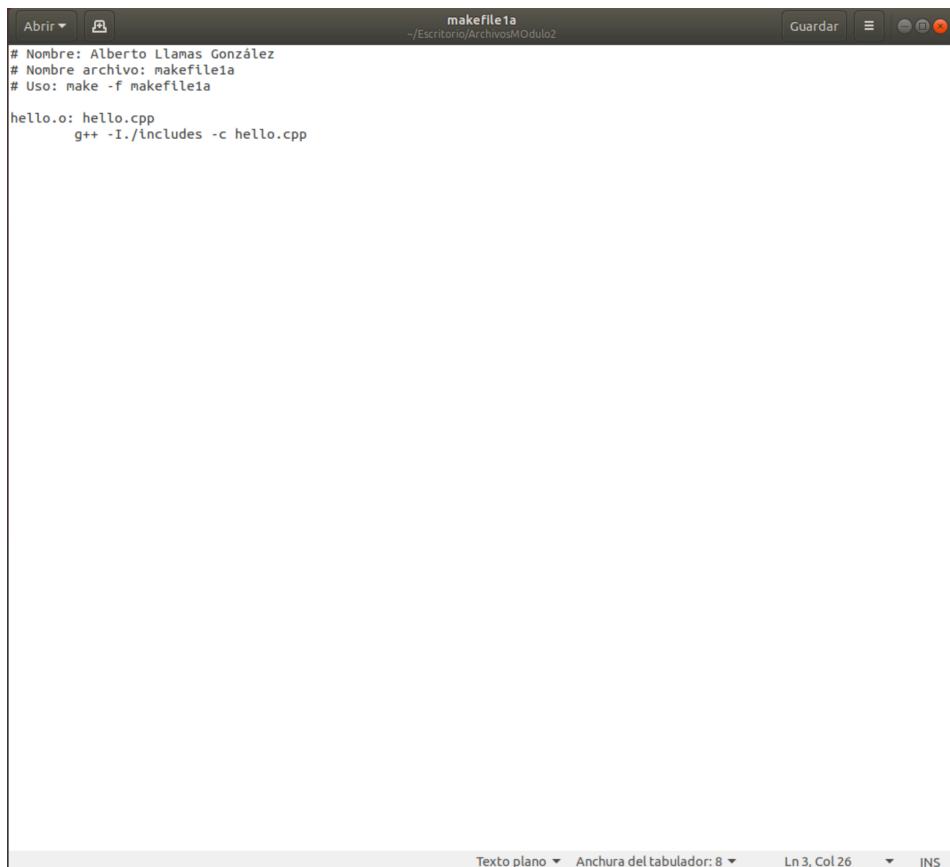
Entrega ejercicios Práctica 7

OBSERVACIÓN: En cada apartado se muestra una captura de pantalla del archivo makefile correspondiente a cada uno. Al final del pdf, se muestra una ejecución de cada archivo makefile de forma seguida en la terminal, mostrando después el directorio (en la terminal). Para realizar este proyecto, se ha utilizado el PDF proporcionado, luego ambos ejercicios se han realizado siguiendo la idea de él. Además, para cada apartado del ejercicio 1, se ha creado un archivo makefile distinto con el objetivo de entender mejor su uso, sin embargo, se ha subido a PRADO únicamente el archivo final 'makefile4'.

Ejercicio 1

Considerando los ficheros proporcionados (main2.cpp, factorial.cpp, hello.cpp, sin.cpp, cos.cpp y tan.cpp), realiza las siguientes operaciones por pasos (no te saltes ninguno y ve mostrando cada uno de los resultados requeridos):

1. Escribe en un archivo makefile las órdenes necesarias para que dé la posibilidad de crear el archivo compilado hello.o.



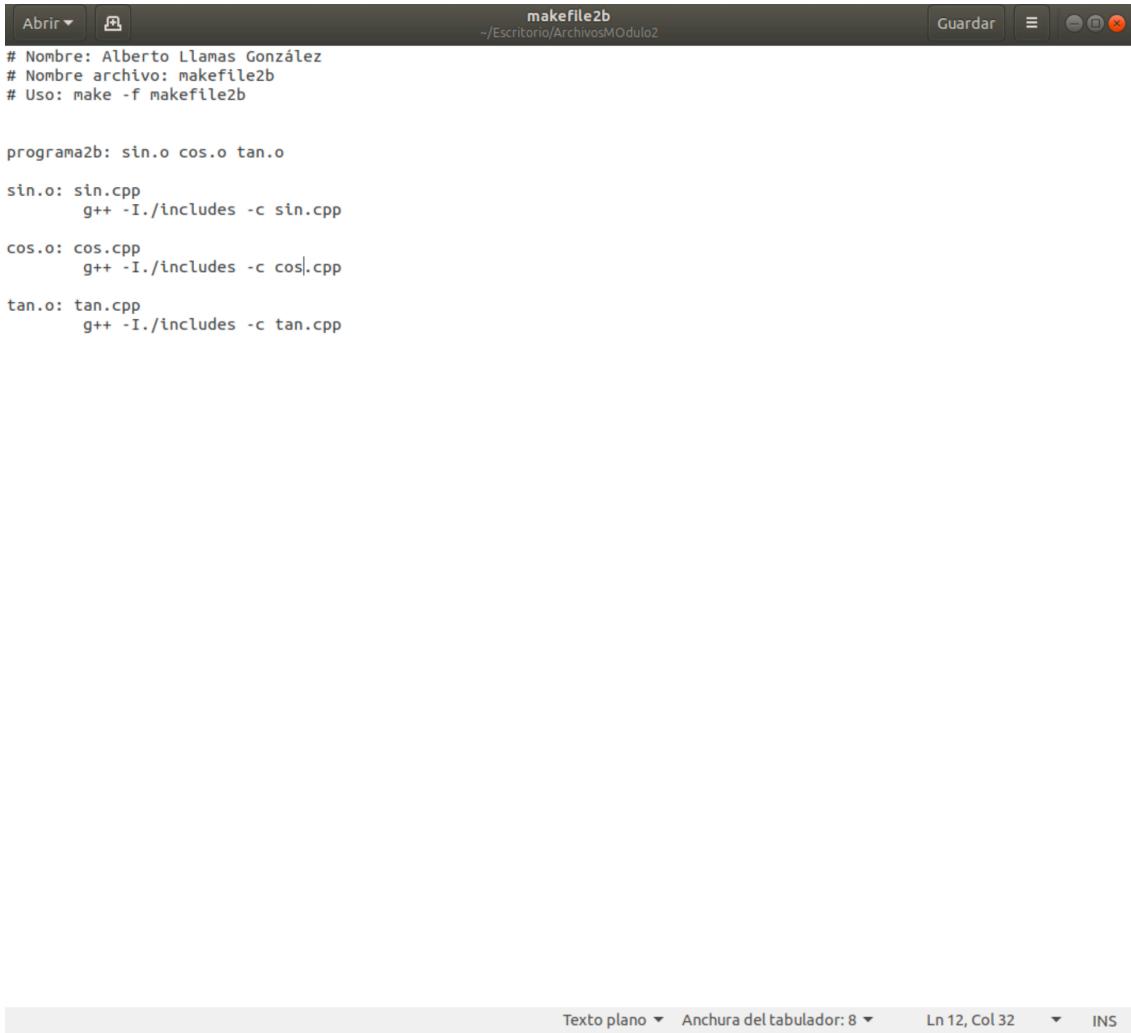
```
Abrir ▾ Guardar makefile1a ~/Escritorio/ArchivosModulo2
# Nombre: Alberto Llamas González
# Nombre archivo: makefile1a
# Uso: make -f makefile1a

hello.o: hello.cpp
    g++ -I./Includes -c hello.cpp
```

- a. Mediante el makefile, haz que se cree dicho fichero. Para crear dicho fichero simplemente hay que utilizar la orden g++ -c fichero.cpp

2. Modifica el mismo archivo para se puedan compilar los archivos tan.cpp, cos.cpp y sin.cpp y obtener sus correspondientes archivos objeto.

a. Mediante el makefile, haz que se creen cada uno de los ficheros independientemente.



The screenshot shows a terminal window titled "makefile2b" with the path "-/Escritorio/ArchivosModulo2". The window contains the following content:

```
# Nombre: Alberto Llamas González
# Nombre archivo: makefile2b
# Uso: make -f makefile2b

programa2b: sin.o cos.o tan.o

sin.o: sin.cpp
    g++ -I./includes -c sin.cpp

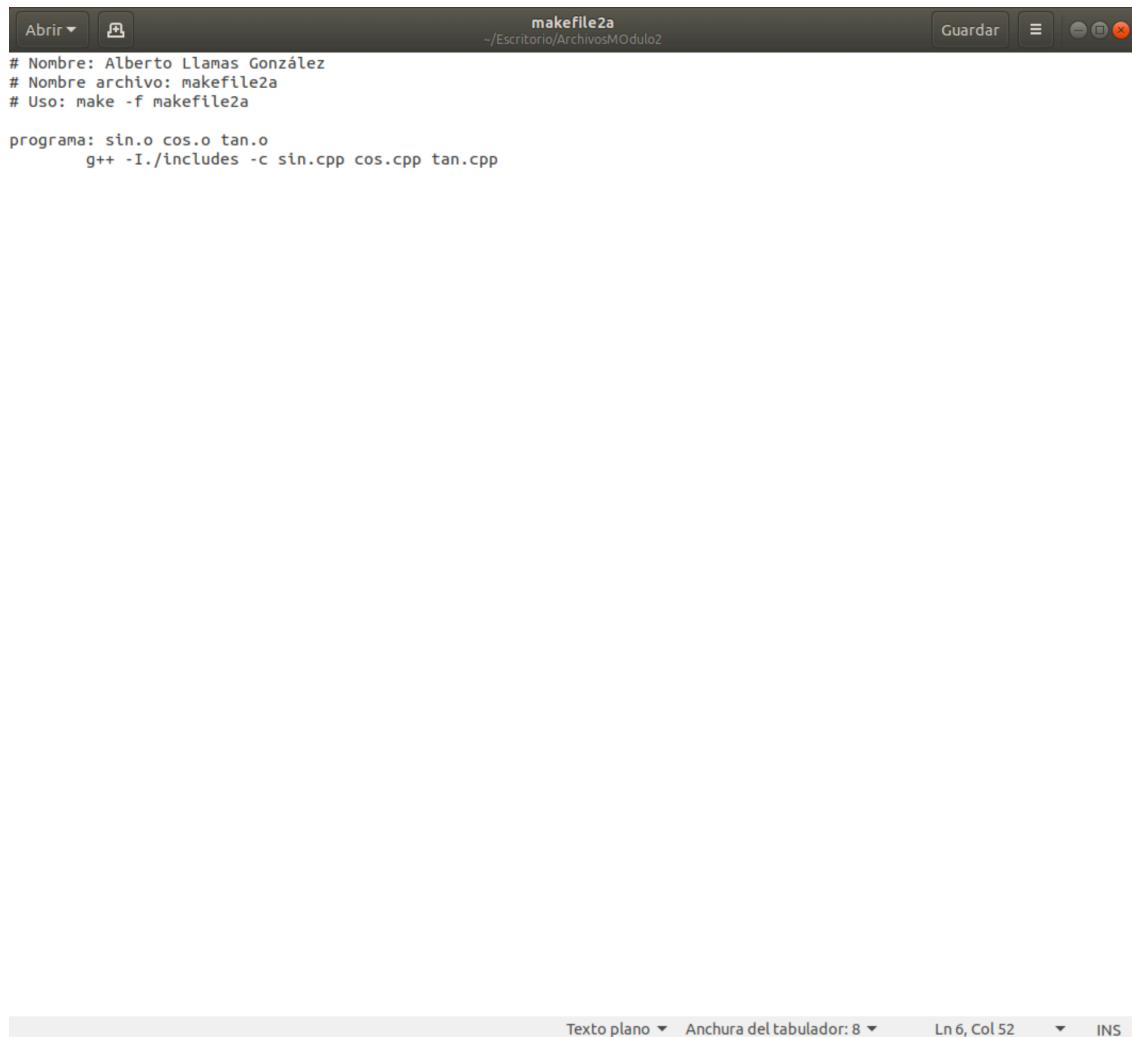
cos.o: cos.cpp
    g++ -I./includes -c cos.cpp

tan.o: tan.cpp
    g++ -I./includes -c tan.cpp
```

At the bottom of the terminal window, there are status indicators: "Texto plano ▾ Anchura del tabulador: 8 ▾ Ln 12, Col 32 ▾ INS".

Obtenemos de la siguiente forma cada fichero independientemente

b. Mediante el makefile, haz que se creen cada uno de los ficheros de 1 sola vez.



The screenshot shows a text editor window titled "makefile2a" located at "/Escritorio/ArchivosModulo2". The file contains the following content:

```
# Nombre: Alberto Llamas González
# Nombre archivo: makefile2a
# Uso: make -f makefile2a

programa: sin.o cos.o tan.o
    g++ -I./includes -c sin.cpp cos.cpp tan.cpp
```

At the bottom of the editor, there are status bars for "Texto plano", "Anchura del tabulador: 8", "Ln 6, Col 52", and "INS".

Se obtienen de una vez de la siguiente forma:

3. Seguidamente, modifica el makefile para que genere una librería llamada libmath.

a. Ejecuta el makefile y comprueba qué resultado ha proporcionado.

```
# Nombre: Alberto Llamas González
# Nombre archivo: makefile3
# Uso: make -f makefile3

programa3: libmath.a

libmath.a: sin.o cos.o tan.o
    ar -rvs libmath.a sin.o cos.o tan.o

sin.o cos.o tan.o: sin.cpp cos.cpp tan.cpp mates.h
    g++ -c sin.cpp cos.cpp tan.cpp
```

Como podemos ver al final del documento, este makefile genera la librería libmath.a

4. Añade las órdenes necesarias para obtener el programa principal compilado y un fichero ejecutable denominado Practica7.



The screenshot shows a terminal window with the following content:

```
# Nombre: Alberto Llamas González
# Nombre archivo: makefile4
# Uso: make -f makefile4

Practica7: main2.o hello.o factorial.o libmath.a
g++ -o Practica7 main2.o hello.o factorial.o libmath.a

main2.o: main2.cpp
g++ -I./includes -c main2.cpp

factorial.o hello.o: factorial.cpp hello.cpp functions.h
g++ -I./includes -c factorial.cpp hello.cpp

libmath.a: sin.o cos.o tan.o
ar -rvs libmath.a sin.o cos.o tan.o

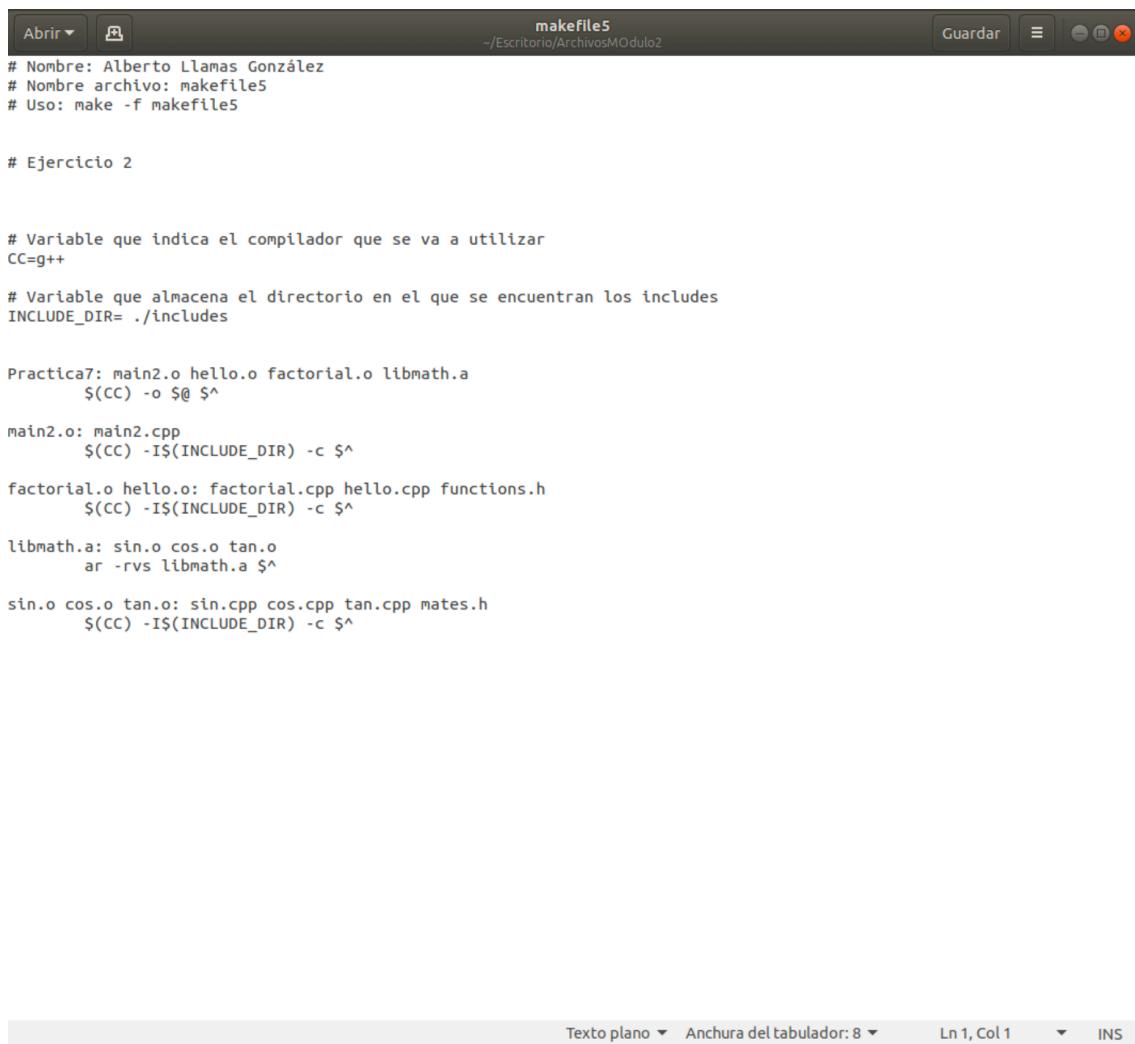
sin.o cos.o tan.o: sin.cpp cos.cpp tan.cpp mates.h
g++ -c sin.cpp cos.cpp tan.cpp
```

At the bottom of the terminal window, there are status indicators: "Texto plano ▾" (Text plain), "Anchura del tabulador: 8 ▾" (Tab width: 8), "Ln 3, Col 25 ▾" (Line 3, Column 25), and "INS".

Ejercicio 2.

Haciendo uso del archivo anterior, genera un nuevo makefile añadiendo al código las siguientes variables:

1. Una variable que almacene dónde están los ficheros include.
2. Una variable que almacene el compilador.
3. Utiliza la variable en la que se guarda el nombre del objetivo de la regla actual para no tener que repetir siempre dicha cadena.
4. Modifica las reglas que puedas utilizando la variable donde se almacenan las dependencias de la regla actual.



The screenshot shows a text editor window with the following content:

```
# Nombre: Alberto Llamas González
# Nombre archivo: makefile5
# Uso: make -f makefile5

# Ejercicio 2

# Variable que indica el compilador que se va a utilizar
CC=g++

# Variable que almacena el directorio en el que se encuentran los includes
INCLUDE_DIR= ./includes

Practica7: main2.o hello.o factorial.o libmath.a
           $(CC) -o $@ $^

main2.o: main2.cpp
         $(CC) -I$(INCLUDE_DIR) -c $^

factorial.o hello.o: factorial.cpp hello.cpp functions.h
                      $(CC) -I$(INCLUDE_DIR) -c $^

libmath.a: sin.o cos.o tan.o
          ar -rvs libmath.a $^

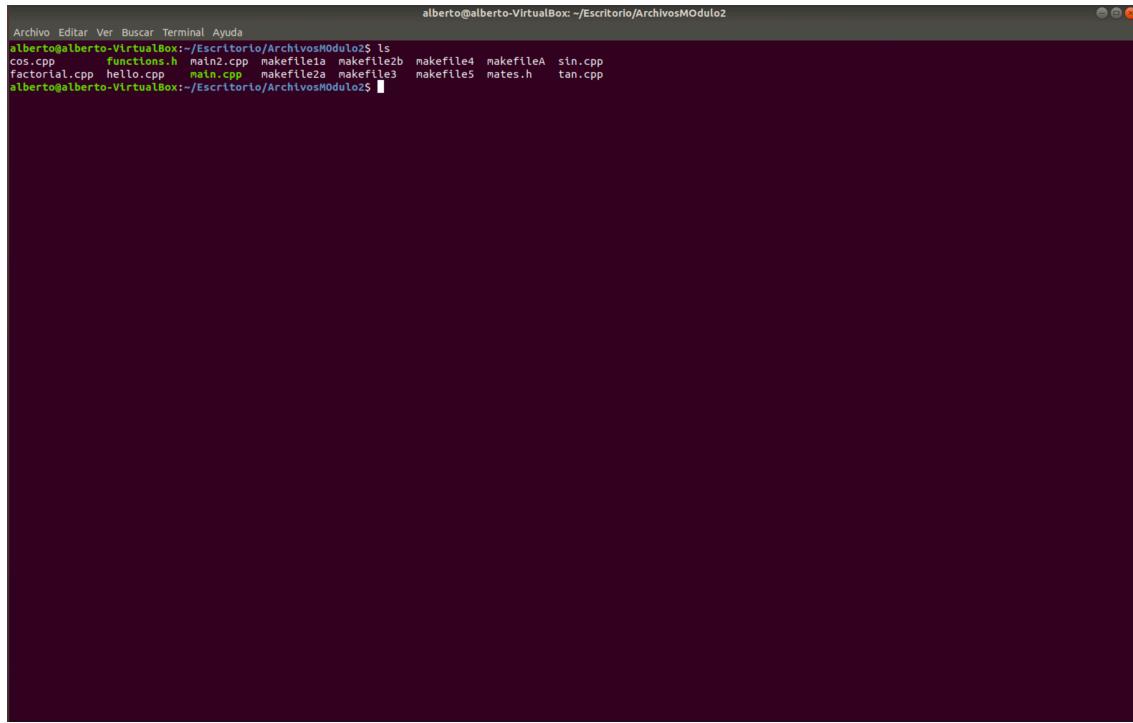
sin.o cos.o tan.o: sin.cpp cos.cpp tan.cpp mates.h
                     $(CC) -I$(INCLUDE_DIR) -c $^
```

At the bottom of the editor window, there are status bars with the following information:

- Texto plano ▾ Anchura del tabulador: 8 ▾
- Ln 1, Col 1 ▾
- INS

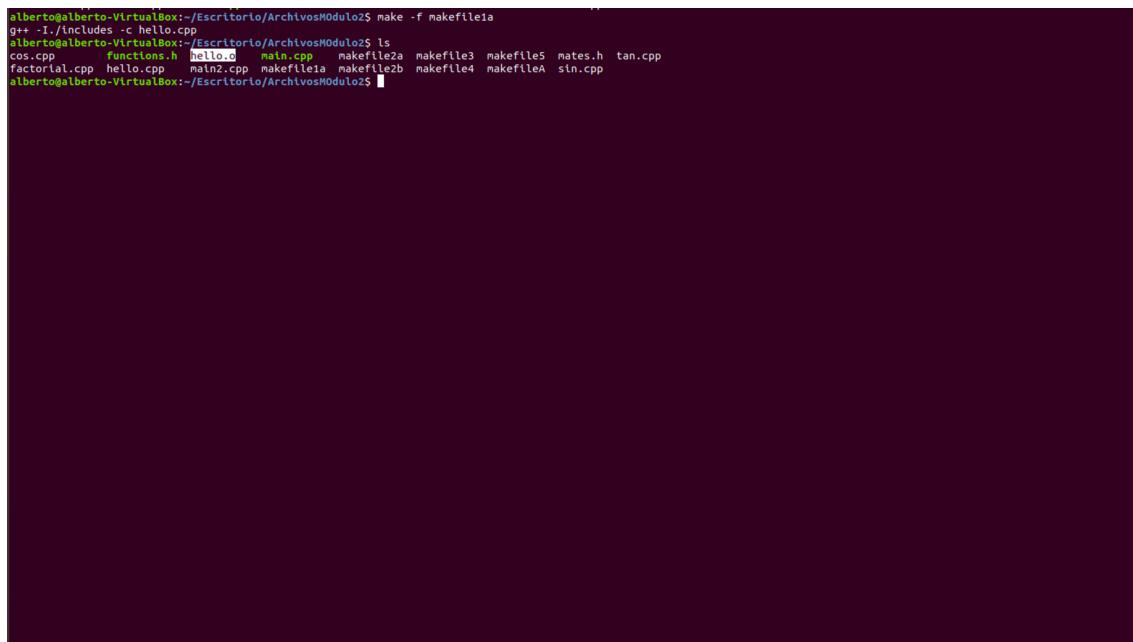
Ejecución completa de ambos ejercicios en la terminal:

Muestro por terminal la carpeta en la que se encuentran los archivos



```
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ ls
cos.cpp      functions.h  main2.cpp  makefile1a  makefile2b  makefile4  makefileA  sin.cpp
factorial.cpp hello.cpp  main.cpp   makefile2a  makefile3  makefile5  mates.h   tan.cpp
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$
```

Ejecutamos el primer makefile y como vemos aparece el .o deseado.



```
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile1a
g++ -I./includes -c hello.cpp
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ ls
cos.cpp      functions.h  hello.o   main.cpp  makefile2a  makefile3  makefile5  mates.h   tan.cpp
factorial.cpp hello.cpp  main2.cpp  makefile1a  makefile2b  makefile4  makefileA  sin.cpp
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$
```

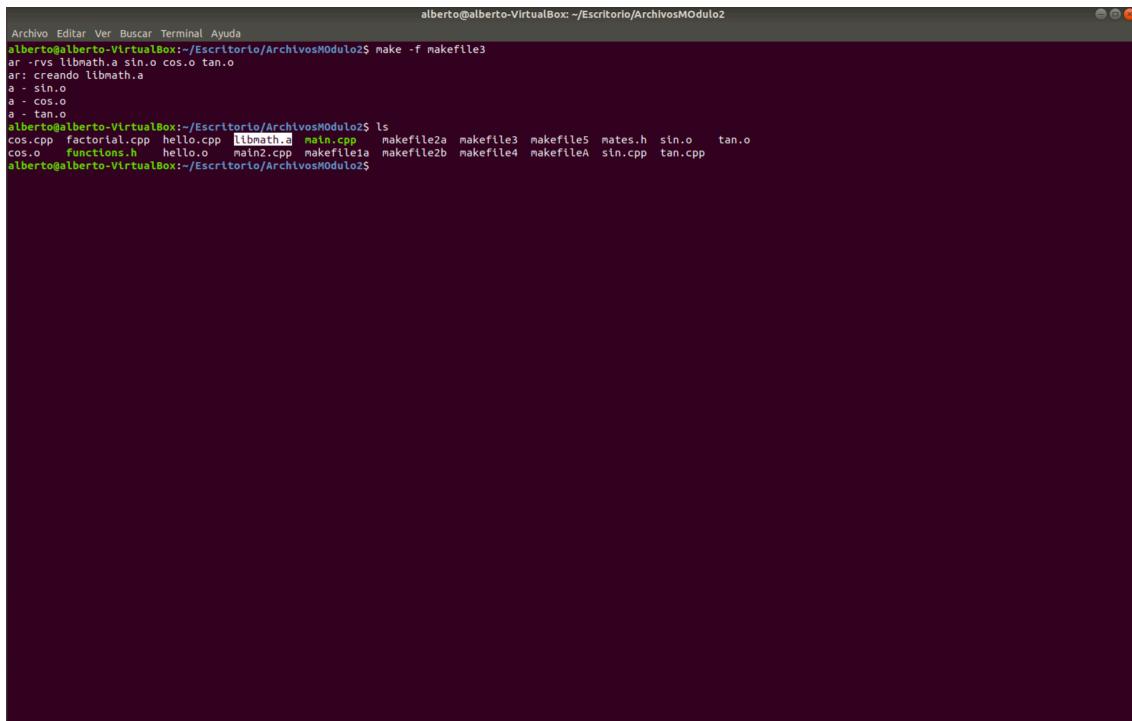
Ejecutamos el segundo (2a) makefile y observamos como debajo de sin.cpp, tan.cpp y cos.cpp aparecen sus respectivos archivos .o.

```
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile2a
g++ -c -o sin.o sin.cpp
g++ -c -o cos.o cos.cpp
g++ -c -o tan.o tan.cpp
g++ -I./includes -c sin.cpp cos.cpp tan.cpp
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ ls
cos.cpp factorial.cpp hello.cpp main2.cpp makefile2a makefile2b makefile4 makefileA sin.cpp tan.cpp
cos.o   functions.h hello.o   main.o   makefile2a makefile3 makefile5 mates.h sin.o   tan.o
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$
```

Ejecutamos el segundo makefile del apartado 2b del ejercicio 1 (he borrado los archivos .o para que se vea cómo crea el makefile los archivos)

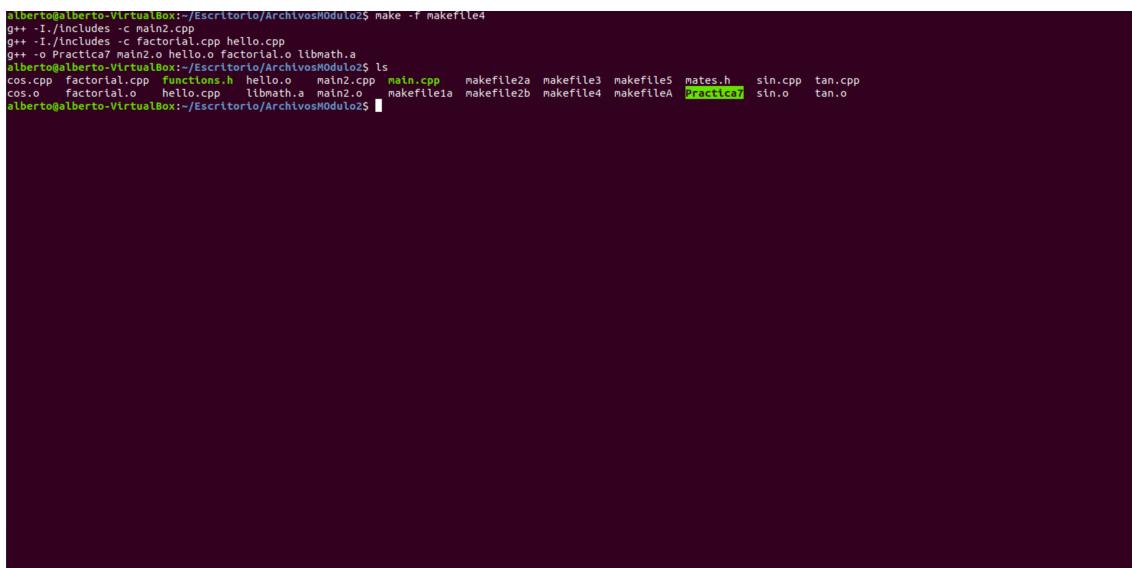
```
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile2b
make: No se hace nada para 'programa2b'.
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile2b
g++ -I./includes -c sin.cpp
g++ -I./includes -c cos.cpp
g++ -I./includes -c tan.cpp
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ ls
cos.cpp factorial.cpp hello.cpp main2.cpp makefile2b makefile4 makefileA sin.cpp tan.cpp
cos.o   functions.h hello.o   main.o   makefile2a makefile3 makefile5 mates.h sin.o   tan.o
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$
```

Creamos la librería libmath:



```
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile3
ar -rvs libmath.a sin.o cos.o tan.o
ar: creando libmath.a
a - sin.o
a - cos.o
a - tan.o
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ ls
cos.cpp factorial.cpp hello.cpp libmath.a main.cpp makefile2a makefile3 makefile5 mates.h sin.o tan.o
cos.o functions.h hello.o libmath2.cpp Makefile1a Makefile2b Makefile4 MakefileA sin.cpp tan.cpp
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$
```

Realizamos el último apartado del ejercicio 1:



```
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile4
g++ -I./Includes -c main2.cpp
g++ -I./Includes -c factorial.cpp hello.cpp
g++ -o Practica7 main2.o hello.o factorial.o libmath.a
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ ls
cos.cpp factorial.cpp functions.h hello.o main2.cpp main.cpp makefile2a makefile3 makefile5 mates.h sin.cpp tan.cpp
cos.o factorial.o hello.cpp libmath.a main2.o makefile1a makefile2b makefile4 makefileA Practica7 sin.o tan.o
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$
```

Como se puede ver en la imagen, se ha mostrado el resultado de ejecutar la orden con el archivo practica7 ya creado, y sin estar creado:

```
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile5
make: 'Practica7' está actualizado.
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ make -f makefile5
g++ -o Practica7 main2.o hello.o factorial.o libmath.a
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$ ls
cos.cpp factorial.cpp functions.h hello.o main2.cpp main.o makefile2a makefile3 makefile5 mates.h sin.cpp tan.cpp
cos.o factorial.o hello.cpp libmath.a main2.o makefile1a makefile2b makefile4 makefileA sin.o tan.o
alberto@alberto-VirtualBox:~/Escritorio/ArchivosModulo2$
```

El directorio quedaría de la siguiente forma:

