

Ejercicio 1.

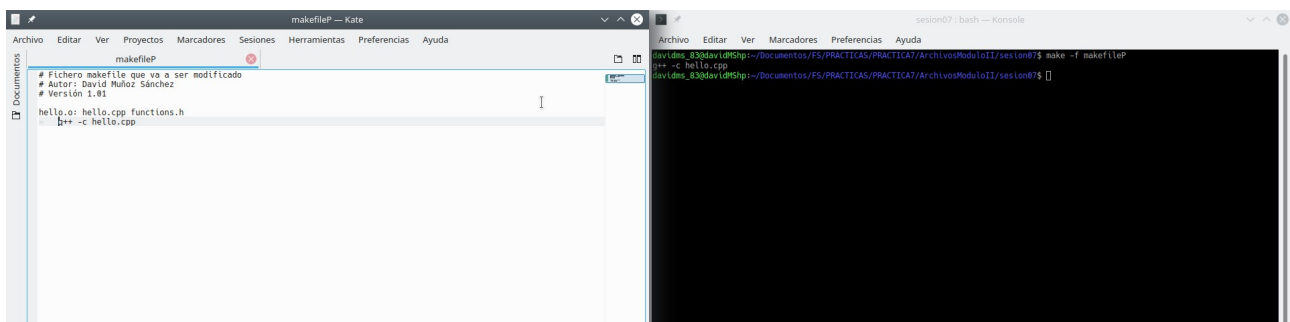
Considerando los ficheros proporcionados (main2.cpp, factorial.cpp, hello.cpp, sin.cpp, cos.cpp y tan.cpp), realiza las siguientes operaciones por pasos (no te saltes ninguno y ve mostrando cada uno de los resultados requeridos):

1. Escribe en un archivo makefile las órdenes necesarias para que dé la posibilidad de crear el archivo compilado hello.o.
 - a) Mediante el makefile, haz que se cree dicho fichero.
2. Modifica el mismo archivo para que se puedan compilar los archivos tan.cpp, cos.cpp y sin.cpp y obtener sus correspondientes archivos objetos.
 - a) Mediante el makefile, haz que se creen cada uno de los ficheros independientemente.
 - b) Mediante el makefile, haz que se creen cada uno de los ficheros de una sola vez.
3. Seguidamente, modifica el makefile para que genere una librería llamada libmath.
 - a) Ejecuta el makefile y comprueba qué resultado a proporcionado.
4. Añade las órdenes necesarias para obtener el programa principal compilado y un fichero ejecutable denominado Practica7.

Para el apartado 1, he introducido la siguiente línea en un archivo makefile, al que he llamado makefileP:

```
hello.o: hello.cpp functions.h  
g++ -c hello.cpp
```

Cabe destacar que como los archivos de cabecera están en el mismo directorio que los archivos fuentes, los puedo poner como una dependencia más. Este es el resultado que he obtenido:



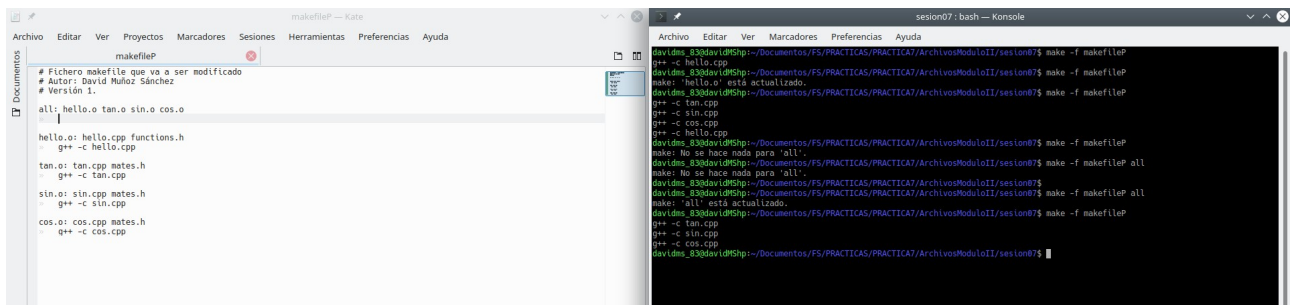
Para el apartado a) del segundo ejercicio, he creado un objetivo principal llamado "all" que tenía como archivos de cabecera todos los ejecutables que pretendía crear, porque de no haberla creado, se hubiera ejecutado solo el primer comando, considerado en ese caso principal y he añadido la siguientes líneas:

```
all: hello.o tan.o sin.o cos.o
```

```
...
```

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

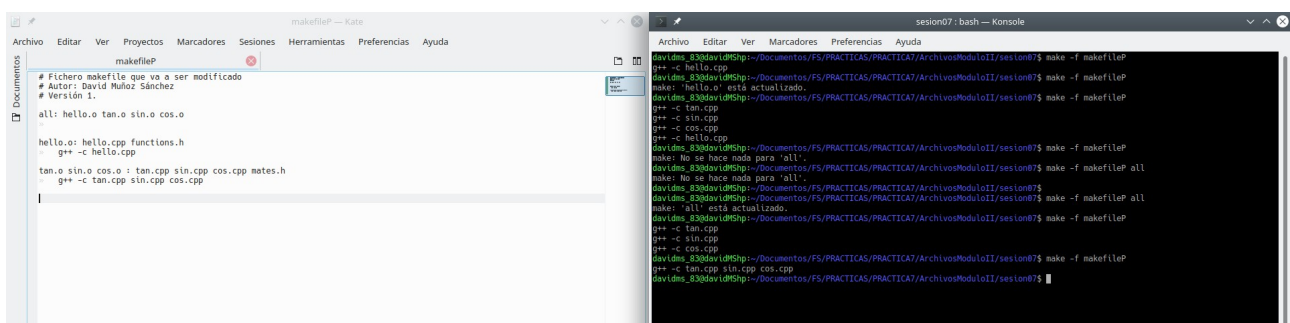
Este es el resultado que he obtenido:



En el b) he hecho una simple modificación para poner la creación de "sin", "cos" y "tan" en una sola línea.

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

Adjunto resultado:



Para el apartado 3, he creado una biblioteca. Para ello he modificado las siguientes líneas del makefileP.

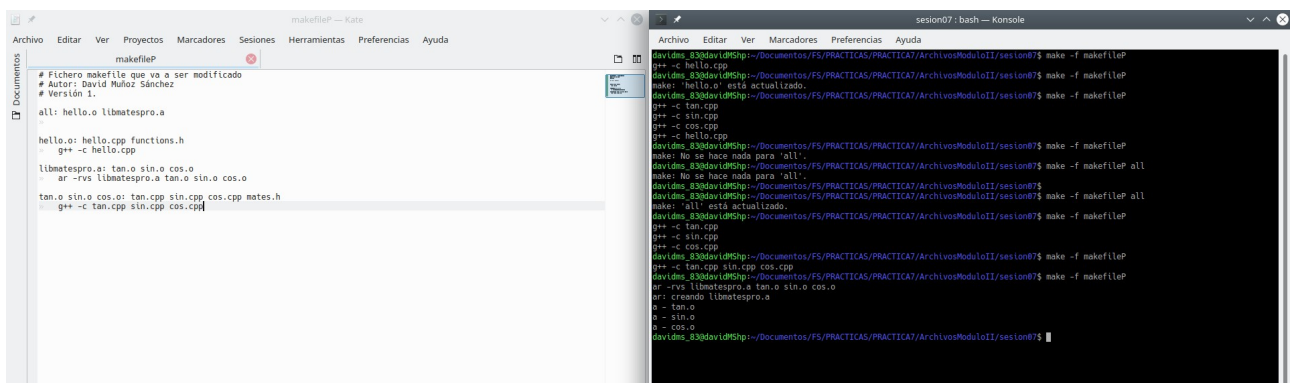
```
all: hello.o libmatespro.a
```

```
...
```

```
libmatespro.a: tan.o sin.o cos.o
ar -rvs libmatespro.a tan.o sin.o cos.o
```

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

He obtenido el siguiente resultado:



En el apartado 4 ya no es necesario mantener “all”, puesto que el programa, si lo asignamos como tarea principal (es decir, la ponemos al principio), necesita de todos los ejecutables que hemos hecho anteriormente.

```
Practica7: main2.o hello.o factorial.o libmatespro.a
g++ -o Practica7 main2.o hello.o factorial.o libmatespro.a
```

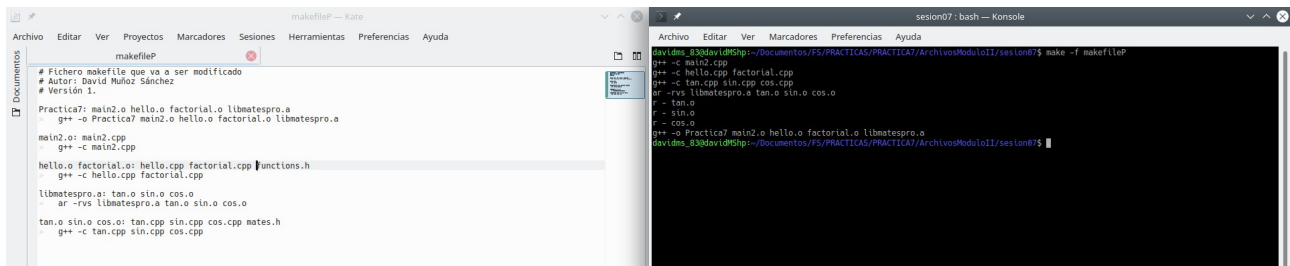
```
main2.o: main2.cpp
g++-c main2.cpp
```

```
hello.o factorial.o: hello.cpp factorial.cpp functions.h
g++-c hello.cpp factorial.cpp
```

```
libmatespro.a: tan.o sin.o cos.o
ar -rvs libmatespro.a tan.o sin.o cos.o
```

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

Este es el resultado:



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.

Adjunto directamente el código del makefile ya que ilustra todas las modificaciones que me pedía el ejercicio, además de una captura que muestra su correcto funcionamiento.

```
INCLU = ./
COM = g++
OBJ = $@
DEPEN = $^
```

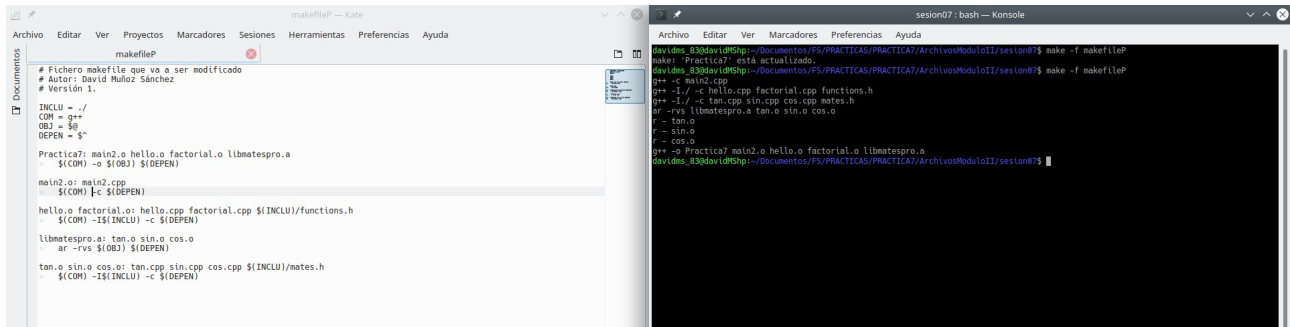
```
Practica7: main2.o hello.o factorial.o libmatespro.a
$(COM) -o $(OBJ) $(DEPEN)
```

```
main2.o: main2.cpp
$(COM) -c $(DEPEN)
```

```
hello.o factorial.o: hello.cpp factorial.cpp
$(INCLU)/functions.h
$(COM) -I$(INCLU) -c $(DEPEN)
```

```
libmatespro.a: tan.o sin.o cos.o
ar -rvs $(OBJ) $(DEPEN)
```

```
tan.o sin.o cos.o: tan.cpp sin.cpp cos.cpp $(INCLU)/mates.h
$(COM) -I$(INCLU) -c $(DEPEN)
```



The image shows a code editor window titled 'makefileP - Kate' and a terminal window titled 'sesion07: bash - Konsole'.

The code editor displays a Makefile with the following content:

```
# Fichero makefile que va a ser modificado
# Autor: David Muñoz Sánchez
# Versión 1.

INCLU = ./
COM = g++
OBJ = %.o
DEPEN = %*

Practica7: main2.o hello.o factorial.o libmatespro.a
    $(COM) -o $(OBJ) $(DEPEN)

main2.o: main2.cpp
    $(COM) -c $(DEPEN)

hello.o factorial.o: hello.cpp factorial.cpp $(INCLU)/functions.h
    $(COM) -c $(INCLU) -c $(DEPEN)

libmatespro.a: tan.o sin.o cos.o
    ar -rsv $(OBJ) $(DEPEN)

tan.o sin.o cos.o: tan.cpp sin.cpp cos.cpp $(INCLU)/mates.h
    $(COM) -c $(INCLU) -c $(DEPEN)
```

The terminal window shows the following commands and output:

```
dauidms_838davidm@hp:~/Documentos/FS/PRACTICAS/PRACTICA7/ArchivosModuloII/sesion07$ make -f makefileP
make: 'Practica7' está actualizado.
dauidms_838davidm@hp:~/Documentos/FS/PRACTICAS/PRACTICA7/ArchivosModuloII/sesion07$ make -f makefileP
g++ -c main2.cpp
g++ -I./ -c hello.cpp factorial.cpp functions.h
g++ -I./ -c tan.cpp sin.cpp cos.cpp mates.h
ar -rsv libmatespro.a tan.o sin.o cos.o
r - tan.o
r - sin.o
r - cos.o
g++ -o Practica7 main2.o hello.o factorial.o libmatespro.a
dauidms_838davidm@hp:~/Documentos/FS/PRACTICAS/PRACTICA7/ArchivosModuloII/sesion07$
```