

## Makefile

1. Construir las reglas de dependencia y construcción de los programas realizados en el práctico anterior: `holamundo.o`, `prueba.o`, `max.o`, `promedio.o`, `holamundo`, `libmath.a`, `libmath.so` y `prueba` (compilado contra `libmath.so`)
2. Crear un objetivo artificial (`.PHONY`) denominado `all` que dependa de `holamundo`, `prueba`, `libmath.a` y `libmath.so`. Colocarlo en el primer lugar, de manera que funcione implícitamente cuando no se indica otra cosa en el comando `make`.
3. Borrar todos los productos (los `.o`, los `.so`, los `.a`, `prueba` y `holamundo`) y ejecutar `make`.
4. Crear reglas implícitas para generar los `.o` a partir de los `.c`, y para crear los ejecutables a partir de los `.o`. Eliminar todos los comandos de aquellas reglas que ya no lo necesiten.
5. Crear una variable `PREFIX` apuntando a `/usr/local`, y las variables `BINDIR`, `LIBDIR` e `INCLUDEDIR` apuntando a `bin`, `lib` e `include` respectivamente, dentro de `PREFIX`.
6. Crear la variable `BINTARGETS` que apunte a `prueba` y `holamundo`; `LIBTARGETS` que apunte a `libmath.so` y `libmath.a` e `INCLUDES` que apunte a `promedio.h` y `max.h`; `OBJECTS` que apunte a `holamundo.o`, `prueba.o`, `max.o` y `promedio.o` y `TARGETS` que incluya `BINTARGETS` y `LIBTARGETS`.
7. Generalizar las reglas que se puedan generalizar utilizando la variable `TARGETS`.
8. Crear un objetivo artificial `clean` que borre del directorio los archivos de backup (terminados en `'~'`), los objetos y los objetivos.
9. Crear un objetivo artificial `install` que instale los encabezados, las librerías y los binarios en sus respectivos directorios, y ejecute `ldconfig` a fin de actualizar el caché del vinculador dinámico. Utilizar el comando `install` para asegurarse de que los directorios de instalación existan, para asignar correctamente los permisos y para comprimir (*strip*) los binarios.
10. Crear un objetivo artificial `uninstall` que revierta los pasos del objetivo `install`, eliminando toda huella del proyecto de los directorios del sistema.
11. Crear una variable `SOURCES` que se forme de lo que contiene `OBJECTS`, cambiando `.o` por `.c`.  

```
SOURCES=$(OBJECTS:.o=.c)
```
12. Crear un objetivo artificial `dep` que dependa del archivo `.depend`.
13. Crear una regla que construya el archivo `.depend` a partir de los fuentes utilizando el comando `gcc -M`, eso creará las dependencias entre los `.o` y los `.h`.
14. Crear una regla `template` que a partir de los OBJETOS se cree la dependencia entre los `.o` y los `.c`.
15. Incluir el archivo `.depend` al final del `Makefile`. Utilizar la directiva `-include`.
16. Probar todo, leer el contenido del archivo `.depend`. Eliminar las reglas innecesarias.