

Dispositivos auxiliares para la práctica 1

Sistemas operativos

Curso 2014-2015

Makefile: ejemplo de la práctica 1

```
TARGET = mitar
CC = gcc
CFLAGS = -g -Wall
OBJS = mitar.o rut_mitar.o
SOURCES = $(addsuffix .c, $(basename $(OBJS)))
HEADERS = mitar.h
```

Definiciones de variables:
se sustituye `$(TARGET)`
por `mitar`

```
all: $(TARGET)

$(TARGET): $(OBJS)
    $(CC) $(CFLAGS) -o $(TARGET) $(OBJS)

.c.o:
    $(CC) $(CFLAGS) -c $< -o $@

$(OBJS): $(HEADERS)

clean:
    -rm -f *.o $(TARGET)
```

Makefiles: ejemplo de la práctica 1

```
TARGET = mitar
CC = gcc
CFLAGS = -g -Wall
OBJS = mitar.o rut_mitar.o
SOURCES = $(addsuffix .c, $(basename $(OBJS)))
HEADERS = mitar.h
```

```
all: $(TARGET)

$(TARGET): $(OBJS)
    $(CC) $(CFLAGS) -o $(TARGET) $(OBJS)

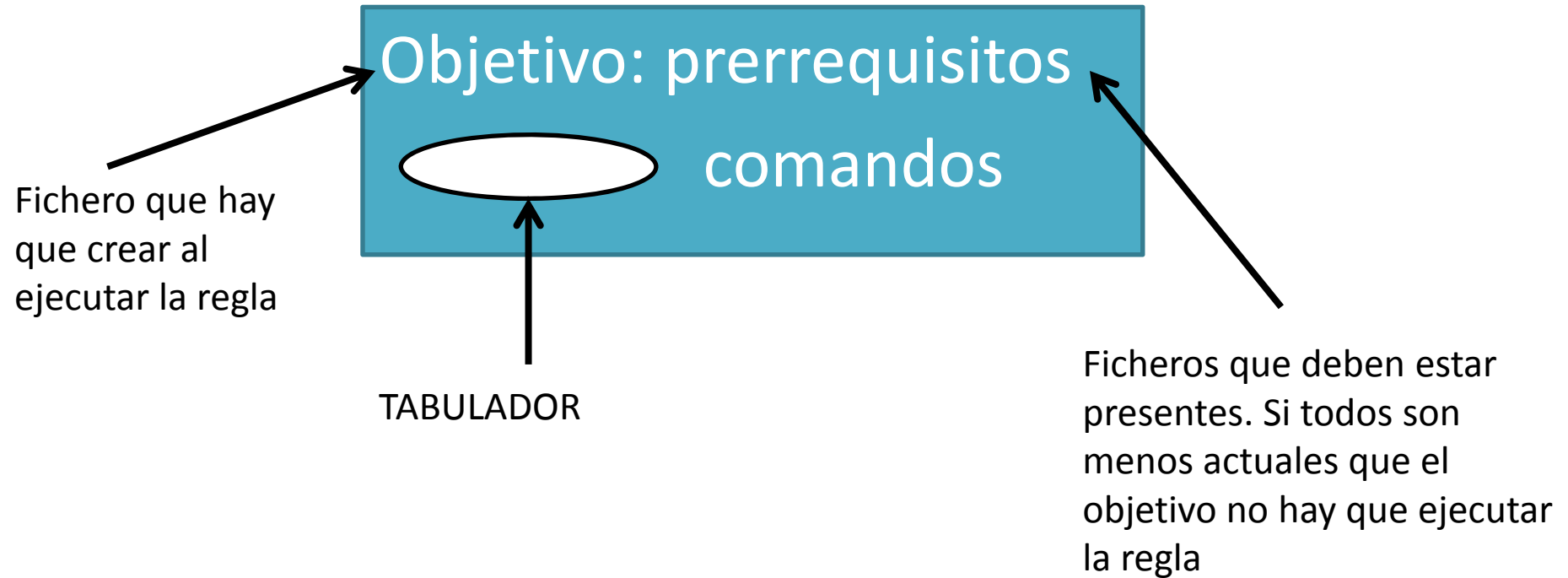
.c.o:
    $(CC) $(CFLAGS) -c $< -o $@

$(OBJS): $(HEADERS)

clean:
    -rm -f *.o $(TARGET)
```

Reglas: pasos a seguir.

Formato de las reglas



Makefiles: ejemplo de la práctica 1

```
TARGET = mitar
CC = gcc
CFLAGS = -g -Wall
OBJS = mitar.o rut_mitar.o
SOURCES = $(addsuffix .c, $(basename $(OBJS)))
HEADERS = mitar.h

all: $(TARGET)

$(TARGET): $(OBJS)
    $(CC) $(CFLAGS) -o $(TARGET) $(OBJS)

.c.o:
    $(CC) $(CFLAGS) -c $< -o $@

$(OBJS): $(HEADERS)

clean:
    -rm -f *.o $(TARGET)
```

\$ **make clean**

Se ejecuta la regla virtual clean.

No se genera ningún fichero:

borra todos los ficheros **.o** y **mitar**

Makefiles: ejemplo de la práctica 1

```
TARGET = mitar
CC = gcc
CFLAGS = -g -Wall
OBJS = mitar.o rut_mitar.o
SOURCES = $(addsuffix .c, $(basename $(OBJS)))
HEADERS = mitar.h
```

```
all: $(TARGET)
```

\$ make

Se ejecuta la regla principal

```
$(TARGET): $(OBJS)
    $(CC) $(CFLAGS) -o $(TARGET) $(OBJS)
```

```
.c.o:
    $(CC) $(CFLAGS) -c $< -o $@
```

```
$(OBJS): $(HEADERS)
```

```
clean:
    -rm -f *.o $(TARGET)
```

Regla implícita: solo se indican las dependencias. El prerequisite es [mitar](#).
⇒ Se procesa la regla para construir el prerequisite.

Makefiles: ejemplo de la práctica 1

```
TARGET = mitar
CC = gcc
CFLAGS = -g -Wall
OBJS = mitar.o rut_mitar.o
SOURCES = $(addsuffix .c, $(basename $(OBJS)))
HEADERS = mitar.h
```

```
all: $(TARGET)
```

```
$(TARGET): $(OBJS)
    $(CC) $(CFLAGS) -o $(TARGET) $(OBJS)
```

```
.O: .c
```

```
    $(CC) $(CFLAGS) -c $< -o $@
```

```
$(OBJS): $(HEADERS)
```

```
clean:
```

```
    -rm -f *.o $(TARGET)
```

Dependencia de la regla

```
gcc -g -Wall -c mitar.c -o mitar.o
gcc -g -Wall -c rut_mitar.c -o rut_mitar.o
```

Objetivo de la regla

Makefiles: ejemplo de la práctica 1

```
TARGET = mitar
CC = gcc
CFLAGS = -g -Wall
OBJS = mitar.o rut_mitar.o
SOURCES = $(addsuffix .c, $(basename $(OBJS)))
HEADERS = mitar.h
```

```
all: $(TARGET)
```

```
$(TARGET): $(OBJS)
    $(CC) $(CFLAGS) -o $(TARGET) $(OBJS)
```

```
gcc -g -Wall -o mitar mitar.o rut_mitar.o
```

```
.c.o:
    $(CC) $(CFLAGS) -c $< -o $@
```

```
$(OBJS): $(HEADERS)
```

```
clean:
    -rm -f *.o $(TARGET)
```