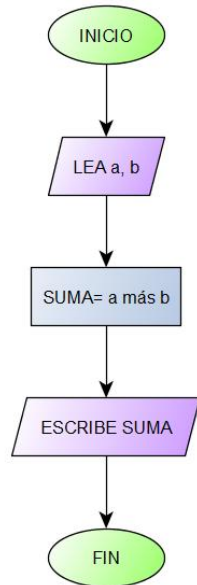


DIAGRAMAS DE FLUJO

1_ Realiza un diagrama que lea dos números, los sume y muestre el resultado.



ALGORITMO SUMA. Las variables son a, b y suma, siendo números enteros.

Inicio.

Escriba (“indique el primer número”);

Lea(a);

Escriba (“indique el segundo número”);

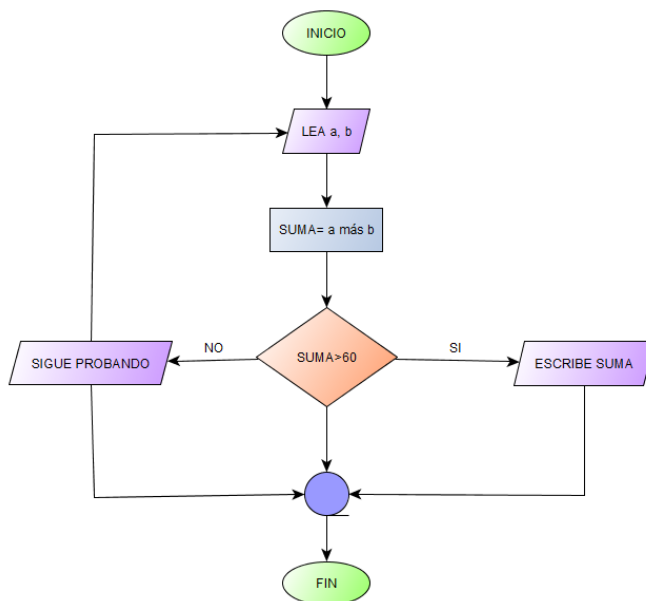
Lea(b);

Suma= a+b;

Escriba (“El resultado es”: suma);

Fin.

2_ Realizar un diagrama que lea dos números, los sume, y si el resultado es mayor que 60, que escriba el resultado, en caso contrario, que escriba la frase “SIGUE PROBANDO”.



ALGORITMO. Las variables son a, b y suma, siendo números enteros.

Inicio.

Escriba (“indique el primer número”);

Lea(a);

Escriba (“indique el segundo número”);

Lea(b);

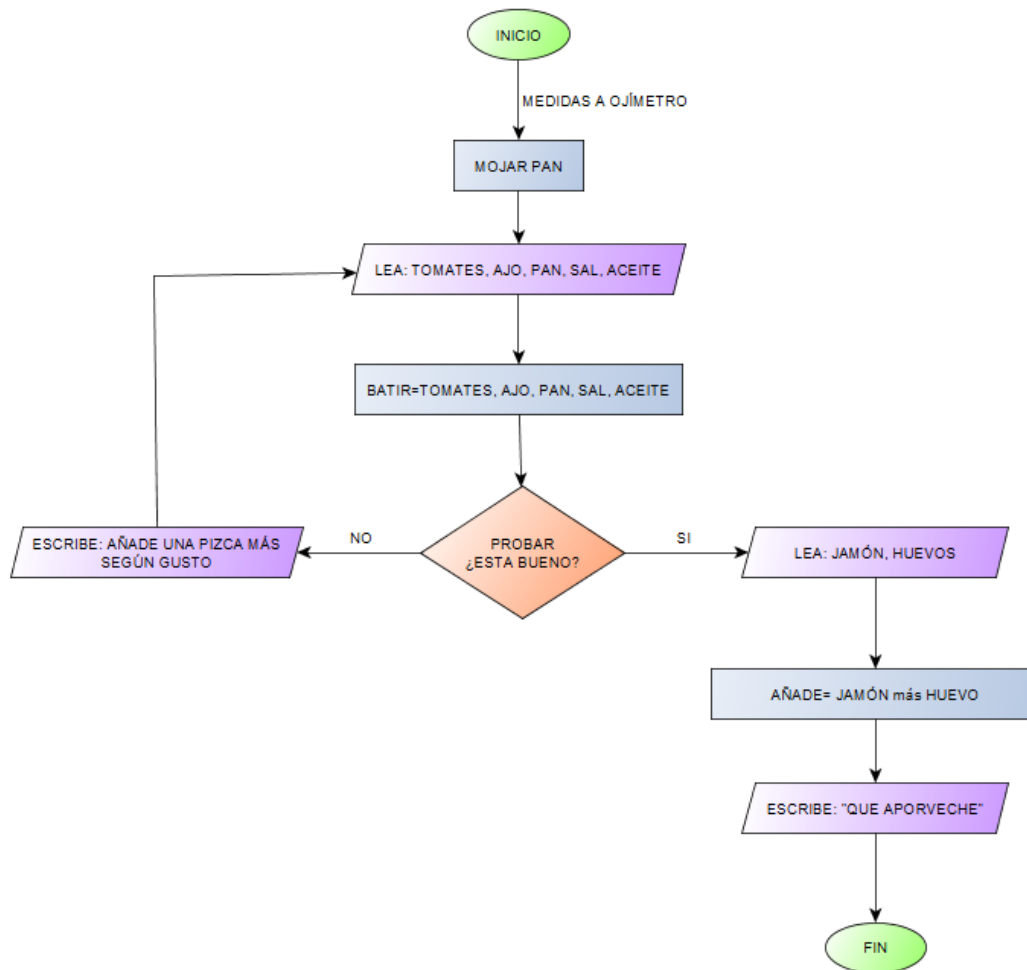
Suma= a+b;

Escriba (“El resultado es”: suma);

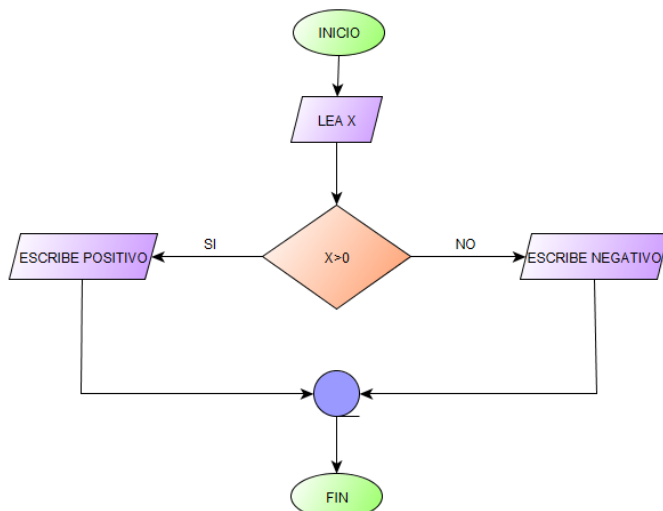
Fin.

3_ **RETO_3_F1** Realiza un diagrama de flujo para cocinar nuestra receta preferida.

La receta representada es de *Porra antequerana*.



4_RETO_3_F2_Realiza un diagrama de flujo para determinar si un número es positivo o negativo.



ALGORITMO. Variable X

INICIO

Lea(X)

Escriba=X

Si (X>0)

Escriba= POSITIVO

SI (X<0)

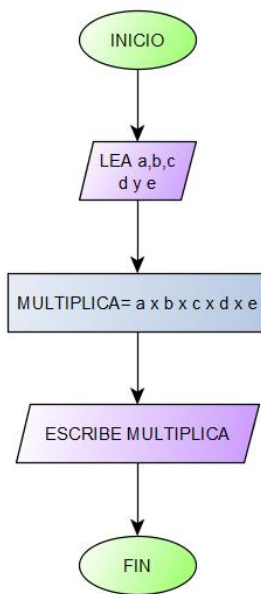
Escriba=NEGATICO

FIN

5_RETO_3_p1. Algoritmo que multiplique cinco números introducidos por el usuario.

SCRIPT: `multiplica5numeros<-function(a,b,c,d,e){
a*b*c*d*e}
multiplica5numeros(a=5,b=6,c=1,d=10,e=4)`

DIAGRAMA DE FLUJO Y ALGORITMO.



ALGORITMO:

Variables a, b, c, d, e, multiplica: entero

INICIO

Escriba ("indique el primer número")

Lea a

Escriba ("indique el segundo número")

Lea b

Escriba ("indique el tercer número")

Lea c

Escriba ("indique el cuarto número")

Lea d

Escriba ("indique el quinto número")

Lea e

Multiplica= a*b*c*d*e

Escriba ("El resultado es:", multiplica)

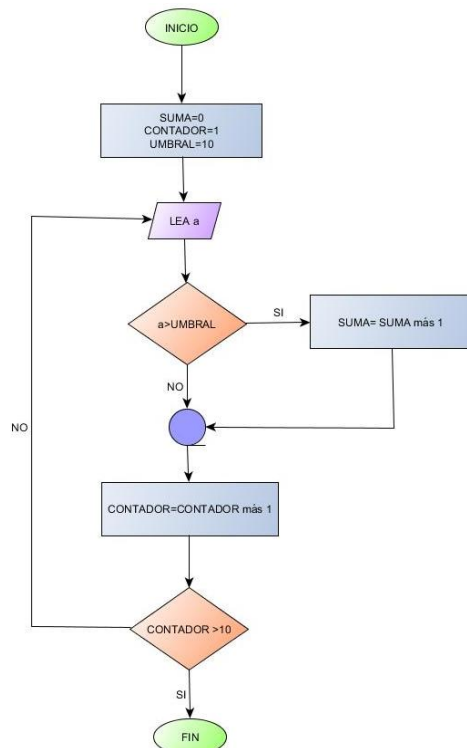
FIN

6_RETO_3_P2_.R. Algoritmo que dado un umbral por el usuario, dados 10 números por el usuario, cuente cuántos de esos números supera el umbral indicado.

```

SCRIPT: numeros<-scan(n=10)
      suma<- (suma=0)
      umbral<-40
      for (valor in numeros [1:10]){
        if (valor>umbral){
          suma<-suma+1}
      }
      print(suma)
  
```

DIAGRAMA DE FLUJO Y ALGORITMO.



ALGORITMO

INICIO

Variables, SUMA, CONTADOR, UMBRAL, a

SUMA=0

CONTADOR=1

UMBRAL=10

LEA a

Mientras (a> UMBRAL)

SUMA= SUMA+1

FIN_Mientras

CONTADOR= CONTADOR+1

Para (CONTADOR=1 hasta 10)

LEA a

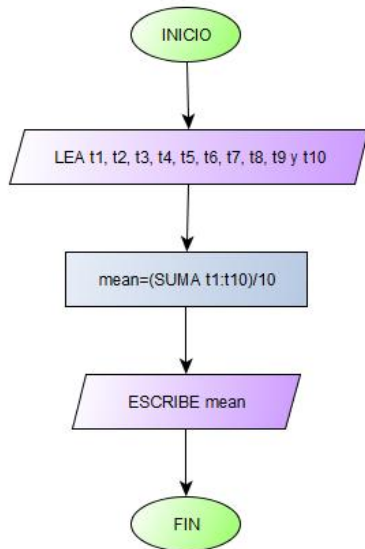
FIN_Para

} BUCLE

7_RETO_3_P3_.R Algoritmo que haga la media de 10 temperaturas indicadas por el usuario.

```
SCRIPT_ temperatura<-scan(n=10)
      mean(temperatura)
```

DIAGRAMA DE FLUJO Y ALGORITMO.



ALGORITMO

Variables Temperatura, t1, t2,...,t10.
 INICIO
 Escriba ("indique el primer número")
 Lea t1
 Escriba ("indique el segundo número")
 Lea t2
 ...
 Escriba ("indique el decimo número")
 Lea t10
 Media (mean)= ((SUMA t1:t10)/10)
 Escribe mean
 FIN

7_RETO_3_P4_.R. Queremos hacer un script que muestre la evolución del NDVI medio para las horas del día.

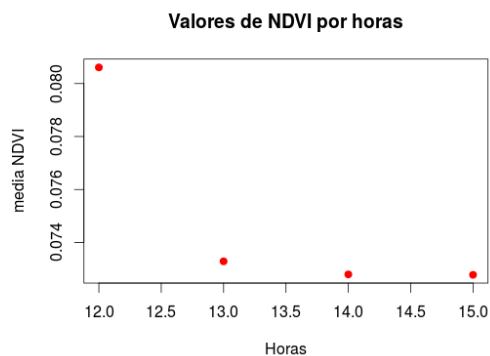
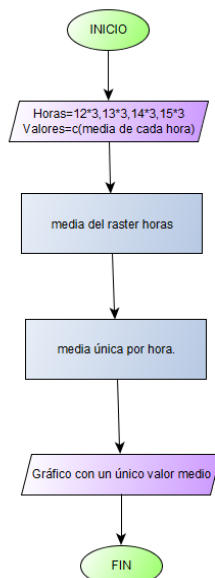
SCRIPT_ en la plataforma Reto_3_p4.r

DIAGRAMA DE FLUJO Y ALGORITMO.

ALGORITMO

Variables: 12 raster con el NDVI, 3 para cada horas.
 Lea todos los raster de cada una de las horas.
 Media = de los 3 raster por hora.
 Media = única para cada raster.
 Escriba=gráfico medias.
 Fin.

} BUCLE



PROGRAMACIÓN EN R_ ASOCIAR COMENTARIOS Y UTILIDAD DE ALGUNAS FUNCIONES.

ASOCIAR COMENTARIOS.

```
## leemos el csv de internet
temperaturas <- read.table("http://www.iecolab.es/ecoinfo/temperatura.csv", header = TRUE,
                           sep= ",")
temperaturas
names(temperaturas) ##ESTE COMANDO ME DEVUELVE EL NOMBRE DE LAS COLUMNAS DE MI
DATAFRAME
```

```
## calculamos la temperatura máxima de las máximas registradas
temperatura_max <- max(temperaturas$tmax) ##el simbolo $= accedo en mi data.frame a la columna que
quiera, solo tengo que poner el nombre.
temperatura_max
```

```
## mostramos la temperatura máxima
print(temperatura_max)
```

```
## calculamos la temperatura mínima de las máximas
temperatura_min <- min(temperaturas$tmin)
temperatura_min
```

```
## mostramos la temperatura mínima
print(paste("La temperatura minima de las temperaturas máximas es:",temperatura_min)) #Lo escrito entre
comillas, aparece tal cual en el resultado
## mostramos un gráfico con la distribución de la temperatura mínima
plot(temperaturas$tmin) ##el comando original era: plot(x$tmin), he cambiado x por "temperaturas", ya que
el objeto x no existe.
```

UTILIDAD DE ALGUNAS FUNCIONES.

```
## Para introducir datos desde la ventana de comandos.
x<-scan(n=20)
```

```
##Para generar sucesiones numéricas, genera un vector que empieza en 1 y acaba en 10, pero de dos en dos.
## generando una secuencia impar.
b<-seq(1, 10, 2)
```

```
##Genera aleatoriamente una distribución normal,
#de n=1000 y cuya media es igual a 3 y desviación estándar = 2.
normal<-rnorm(1000, mean = 3, sd = 2)
```

```
##Crea una matriz con distribución normal, nrow y ncol devuelven el número de filas y de columnas de una
matriz.
f<-matrix(rnorm(1000), nrow = 10, ncol = 100)
```

```
##Ayuda con el paquete raster.
help(package = "raster")
```

##Este script me dice lo que hace varias funciones.

```
##Crear matrices: El 2 es el número de filas y el 5 son las columnas
x<-matrix(seq(1:10),2,5)
x #me devuelve una matriz de las características que le he dicho (2 filas y 5 columnas)
#[,1] [,2] [,3] [,4] [,5]
#[1,]  1  3  5  7  9
```

```
#[2,] 2 4 6 8 10
```

```
x[1,3]?
```

```
x[2, ]? #SUPUESTAMENTE ME DEVOLVERIA UN NÚMERO DE LA MATRIZ, PERO NO PASA  
NADA...¿PREGUNTAR?
```

```
x[, 3]?
```

mode(x) ## muestra el tipo de objeto

```
#[1] "numeric" (resultados de la consola)
```

str(x) ## obtenemos la estructura de un objeto

```
# int [1:2, 1:5] 1 2 3 4 5 6 7 8 9 10 (resultados de la consola)
```

length(x) ## devuelve la longitud de un vector

```
# [1] 10 (resultados de la consola)
```

mean(x) ## devuelve el valor medio

```
# [1] 5.5 (resultados de la consola)
```

max(x) ## devuelve el máximo

```
# [1] 10 (resultados de la consola)
```

##Te hace una lista de los documentos dentro del directorio.

list.files ()

##Une todos los vectores de caracteres que se le suministran y construye una sola, cadena de caracteres. Cada argumento se separa del siguiente con un espacio en blanco, se puede cambiar por "sep" que sustituye el espacio en blanco.

paste()

#Concatenar varios vectores en un vector único. Debe de tener alguno de estos argumentos, **x** siendo una lista o data.frame que se quiera concatenar, **select** que indica las variables que se seleccionan de la data.frame o

from.

stack()

#Construyen matrices uniendo unas con otras verticalmente (por fila). Sustituye filas por columnas.

rbind()