



ECOINFORMÁTICA

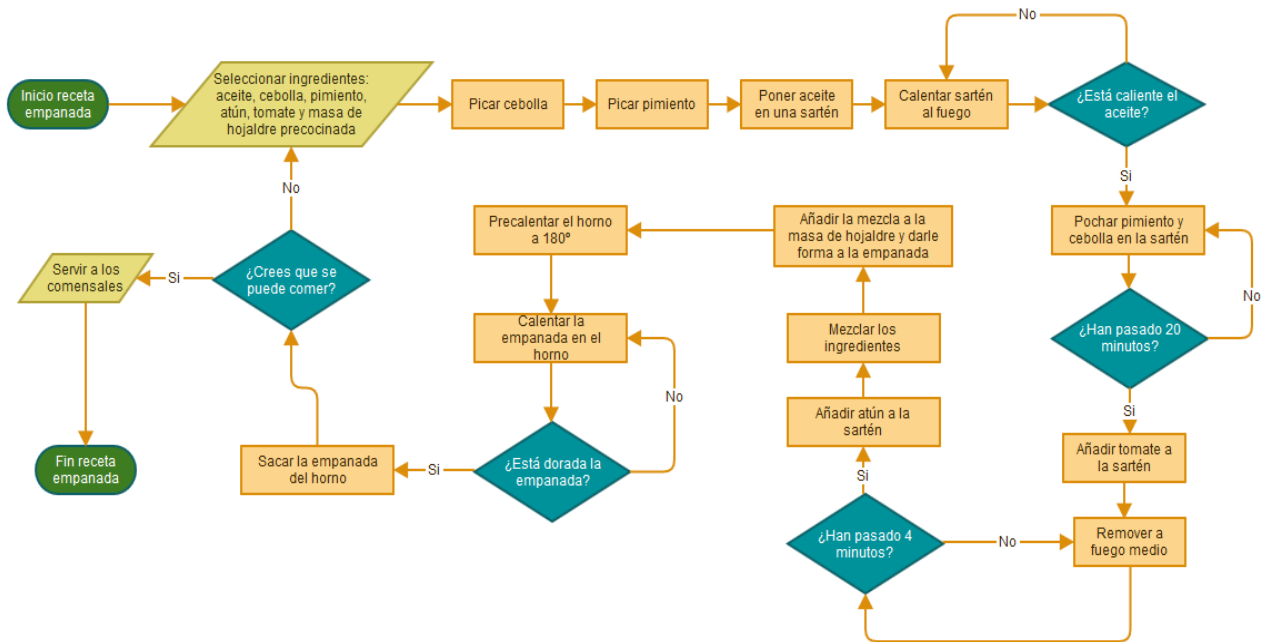
Reto 3: Algorítmica e Introducción a R

María de la Luz Tovar Salvador



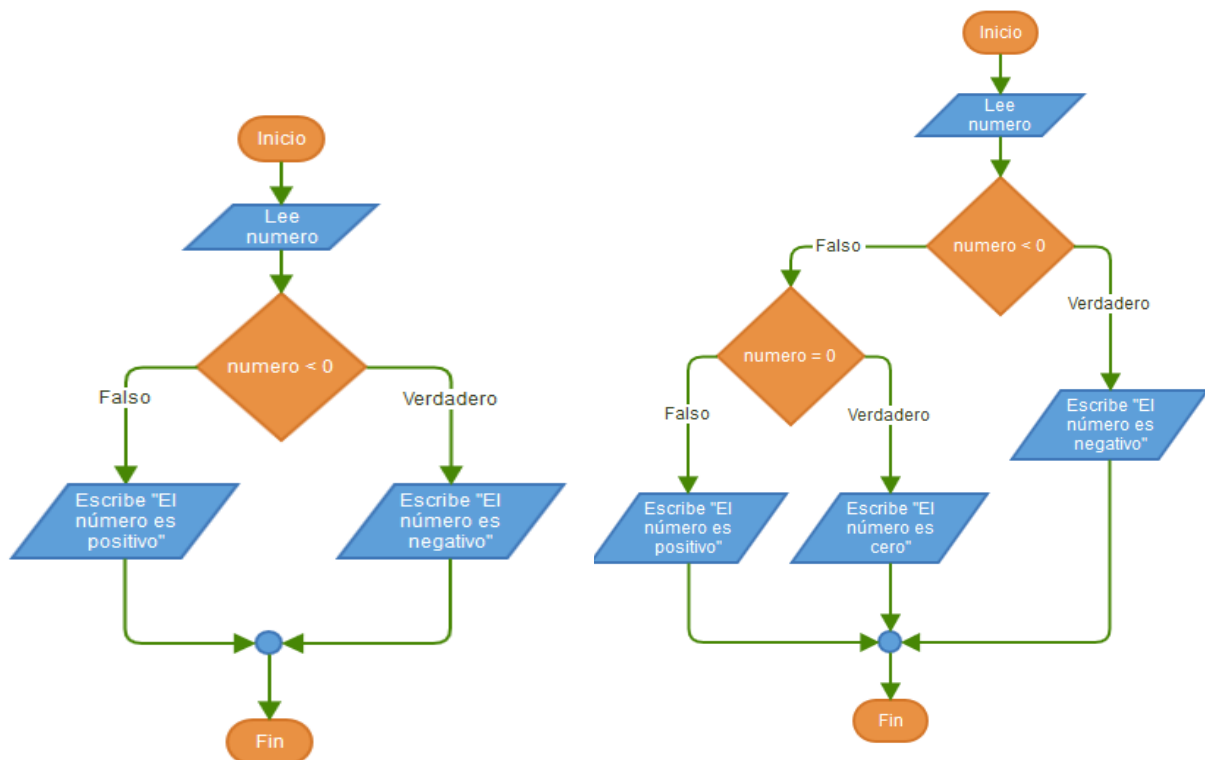
Reto_3_f1 Diagrama de flujo para cocinar vuestra receta preferida.

Receta de empanada de atún.



Reto_3_f2 Diagrama de flujo para determinar si un número es negativo o positivo.

En este caso, he hecho dos supuestos. El primero separa los números en negativos si son menores a cero y en positivos a partir de cero. El segundo diagrama diferencia números negativos, números positivos y el cero.



Reto_3_p1 Algoritmo que multiplique 5 números introducidos por el usuario.

```
multiplicacion<-1
```

```
##input <- scan(file="", n=5, sep= " ") ## con file="" indicamos que se va a leer de teclado,  
n=5 que va a leer 5 numeros como maximo y sep indica como se van a separar los valores  
introducidos
```

```
##input <- c(1,2,3,3,2) ##creamos un vector de datos directamente desde el script
```

```
for (indice in 1:5){
```

```
  input <- scan(file="", n=1)
```

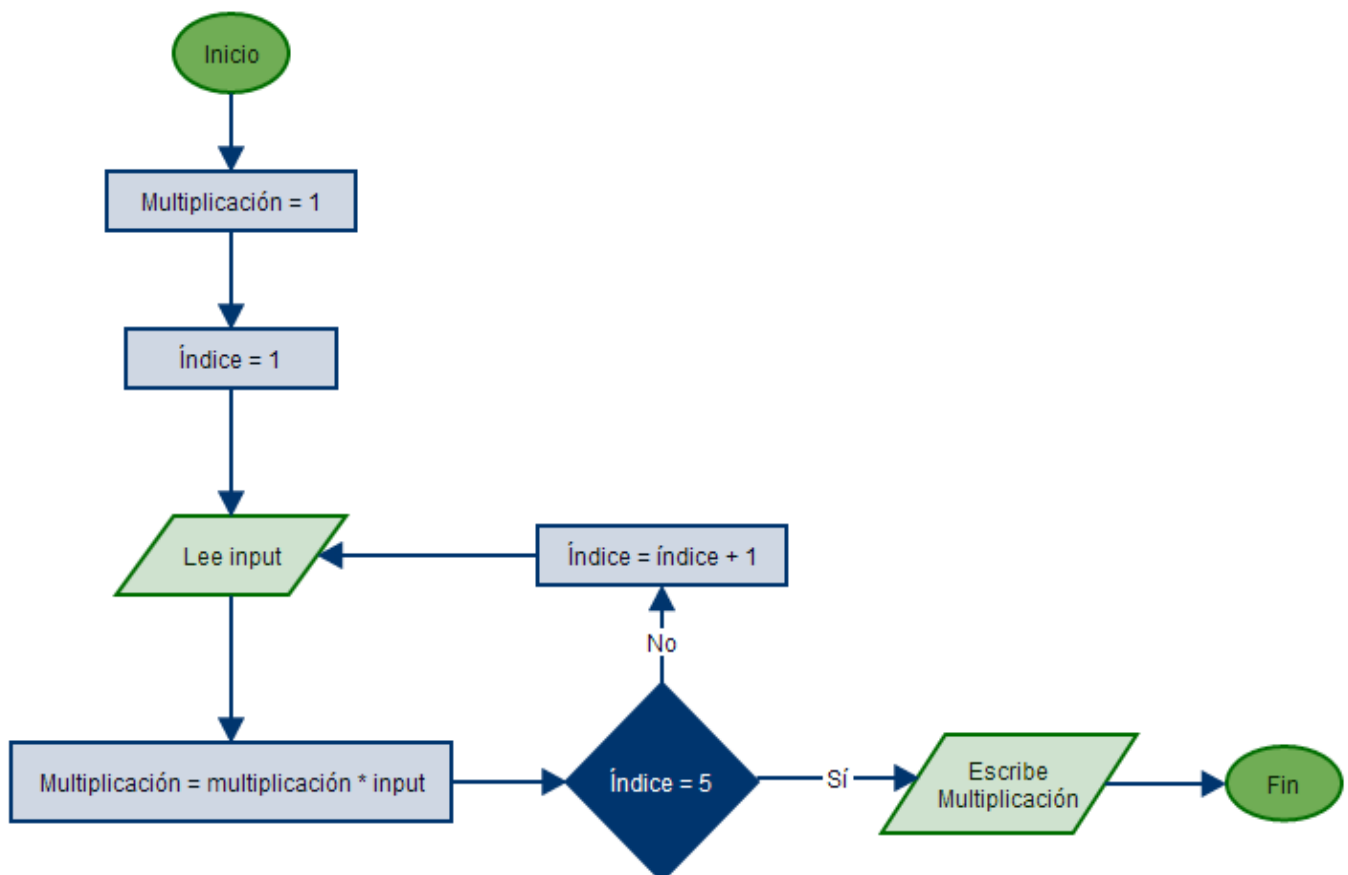
```
  multiplicacion <- multiplicacion * input
```

```
  ##multiplicacion <- multiplicacion * input[indice]
```

```
}
```

```
print(multiplicacion)
```

Diagrama que multiplique 5 números introducidos por el usuario.

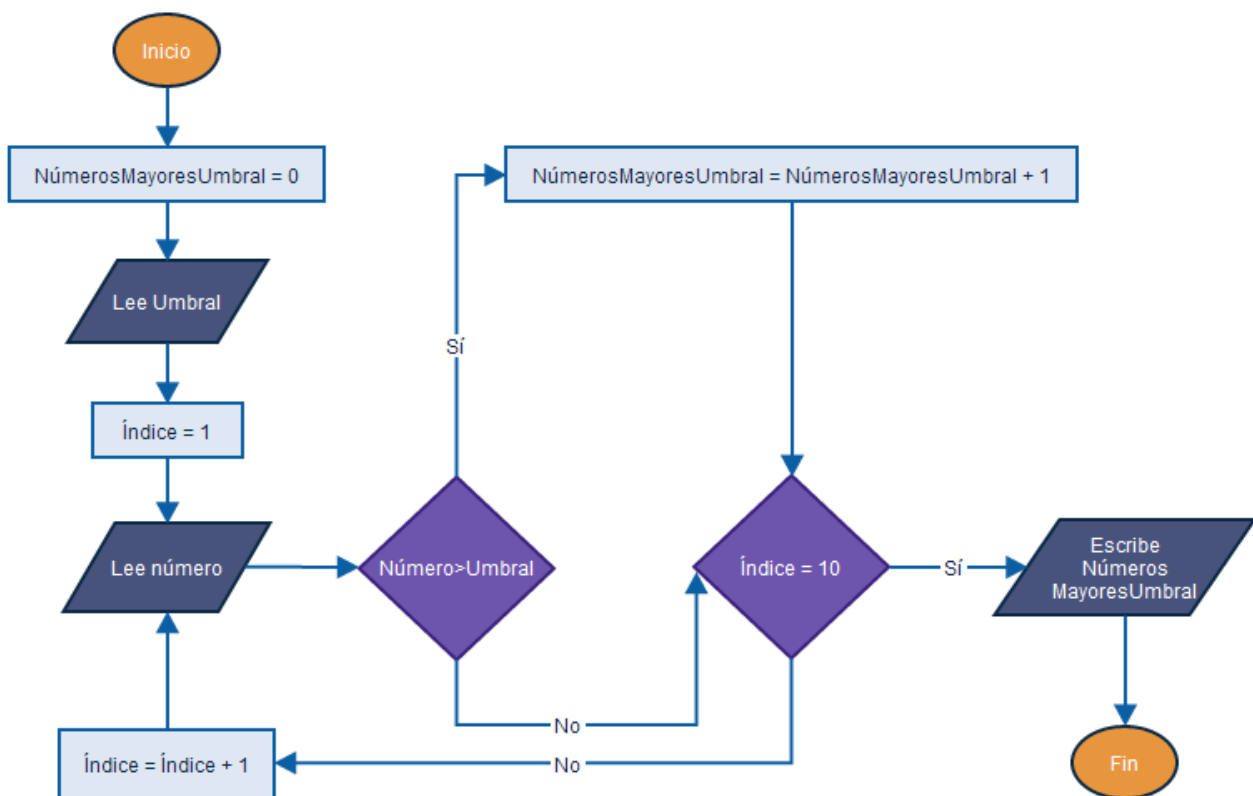


Reto_3_p2 (opcional)

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

```
print(paste("Introduzca el valor umbral: "))
umbral <- scan(file="",n=1) ##leemos el umbral de teclado
numerosMayoresUmbral = 0
print(paste("Introduzca 10 numeros por teclado: "))
for(indice in 1:10){
  numero <- scan(file="",n=1) #leemos un numero de teclado
  if(numero > umbral){
    numerosMayoresUmbral <- numerosMayoresUmbral + 1
  }
}
print(paste("El total de valores que han superado el umbral es: ",numerosMayoresUmbral))
```

Diagrama de flujo



Reto_3_p3 Algoritmo que haga la media de 10 temperaturas indicadas por el usuario.

VERSIÓN 1

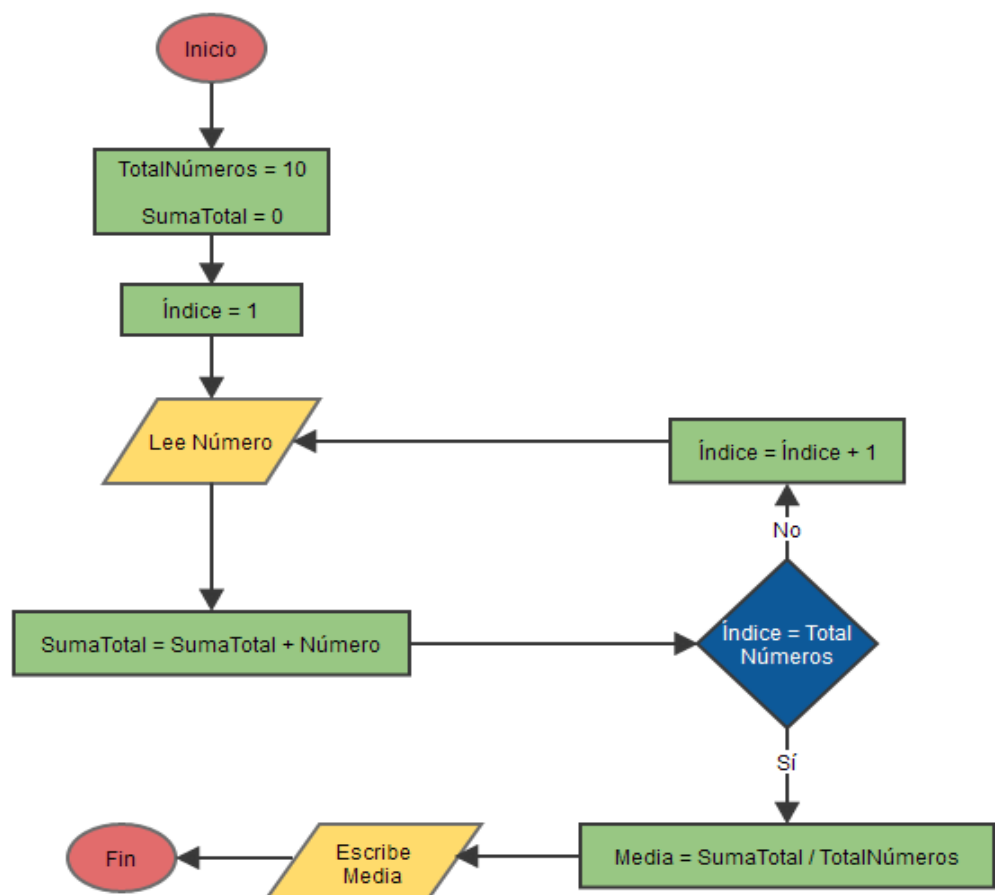
```
input <- scan(file="", n=10, sep= " ") ## con file="" indicamos que se va a leer de teclado, n=5
que va a leer 5 numeros como maximo y sep indica como se van a separar los valores
introducidos
##input <- c(15,24,30,23,20,14,10,19,26,25) ##creamos un vector de datos directamente desde
el script
media <- mean(input)

print(media)
```

VERSIÓN 2

```
totalNumeros <- 10
sumaTotal <- 0
for(indice in 1:totalNumeros){
  numero <- scan(file="",n = 1)
  sumaTotal <- sumaTotal + numero
}
media <- sumaTotal / totalNumeros
print(paste("La temperatura media es: ",media))
```

Diagrama de flujo:



Reto_3_p4 (reto estrella)

Dado el conjunto de datos ndvi.zip, queremos hacer un script que muestre la evolución del NDVI medio para las horas del día.

```
library(raster)
#install.packages(c("rgdal"))
library(rgdal)

setwd("~/Ejemplo/RetosMluz")

rutaImagenes <- "~/ecoinformatica_2014_2015/RetosDesarrollo"

horas <- c("12","13","14","15")

## vector vacio para ir acumulando los vectores
valores <- c()

for(indice in horas){
  ##imagenes <- list.files(path="../Ejemplo/RetosDesarrollo/", full.names = TRUE,
  pattern=paste("_",horas[indice],"..\\.jpg\\.asc$",sep=""))
  imagenes <- list.files(path=rutaImagenes, full.names = T,
  pattern=paste("_",indice,"..\\.jpg\\.asc$",sep=""))
  print(imagenes) ##lo hago para ver las imagenes que ha leído hasta ahora
  ## hacemos el stack de imagenes
  stack <- stack(imagenes)
  ## calculamos las media con el mean
  media <- mean(stack)
  print(media) ##para ver la media que llevamos para las imagenes que hemos leído
  ## "1" calculamos la media de todas las celdas (cellStats)
  mediaCeldas <- cellStats(media,stat='mean')
  print(mediaCeldas) ##vemos la media total de todas las celdas
  ## usamos el rbind para concatenar el vector con el valor calculado en "1"
  valores <- rbind(valores,mediaCeldas)
}

# pintar el vector valores
print(valores) ##miro los resultados totales que he calculado antes de pintarlos en la grafica
plot(valores)
```

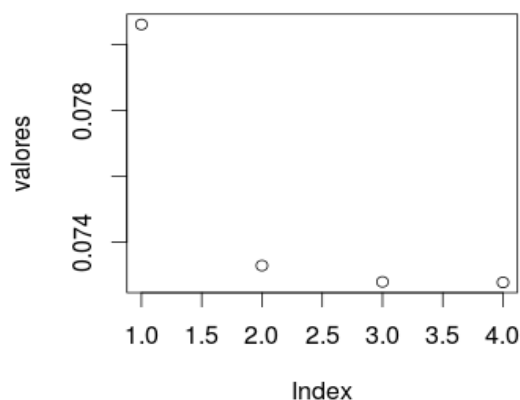


Diagrama de flujo

