Biotecnología Programación 2023-2024
1º curso Grupo M10

Dificultad: media-alta
Revisado por M2

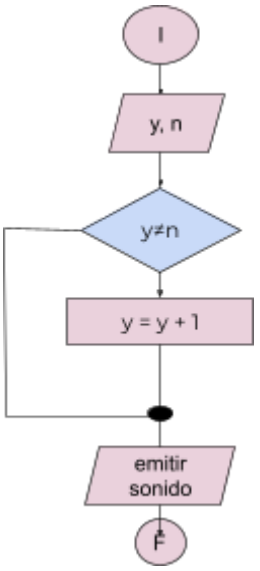
REPASO BUCLES Y OTRAS ESTRUCTURAS

En este documento aparecen las ideas básicas que debéis tener sobre bucles para las prácticas. Las estructuras más utilizadas son los bucles for, while y las estructuras condicionales if, por lo que son en los que nos vamos a centrar.

El **objetivo** de este documento es adquirir los conocimientos básicos que se necesitan saber para elaborar los ejercicios propuestos en las prácticas. Además, incluimos algunos consejos o aspectos a tener en cuenta a la hora de hacer algún ejercicio que contengan estas estructuras, por lo que sería conveniente leer el documento antes de realizar las prácticas.

BUCLES WHILE

Los bucles while sirven para hacer un proceso que se va a **repetir** sucesivamente hasta que deje de cumplir la condición. Sin embargo, las veces que se repite (número de vueltas) no está predeterminado, solo hay una condición que, mientras se cumpla, el bucle continuará. Si al terminar una vuelta deja de cumplirse dicha condición, el bucle finaliza.

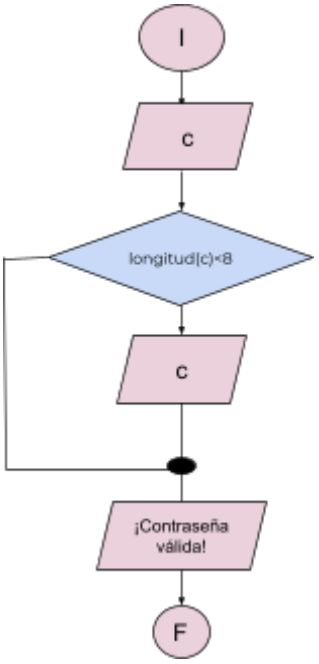


En este caso el valor de y comienza en 0, y se le irá sumando 1 siempre que "y" sea distinto de "n". Llegará un momento en el que la condición no se cumplirá (n será igual que y) por lo que el bucle terminará y en este caso, un aparato emitirá un sonido.

Es importante que de alguna manera la condición pueda cambiar dentro del bucle. Si no se hace, la condición nunca dejará de cumplirse y el bucle será infinito: no parará nunca.

¿Cómo se realiza en R?

```
while (Condición) {
Proceso
}
```



Una posible utilidad podría ser para la creación de condiciones de contraseñas. En este ejemplo, el programa no te dejará avanzar si no introduces una contraseña con 8 o más caracteres. Una vez los tiene, la condición deja de cumplirse y te deja avanzar.

Biotecnología Programación 2023-2024 Dificultad: media-alta
1º curso Grupo M10 Revisado por M2

BUCLES FOR:

Los bucles for se utilizan para operaciones sencillas, como puede ser sumas o multiplicaciones, de forma repetitiva.

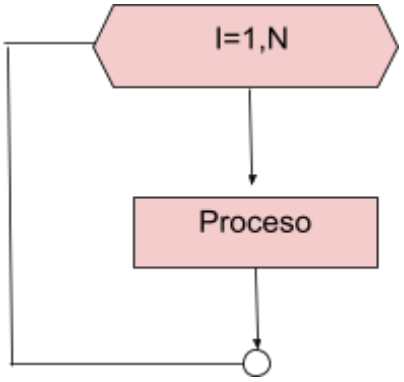
La forma que utilizamos para los bucles for es la siguiente:

Para la elaboración de algoritmos con bucles for hay que tener en cuenta que vamos a necesitar: una variable control, valor inicial de la variable, variable final de la variable, incremento con el que se pasa desde el valor inicial hasta valor final.

Lo que va ocurrir con los bucles for es que se va a ir incrementando el valor inicial que se va a comparar con la variable control hasta que esta sea igual al valor final.

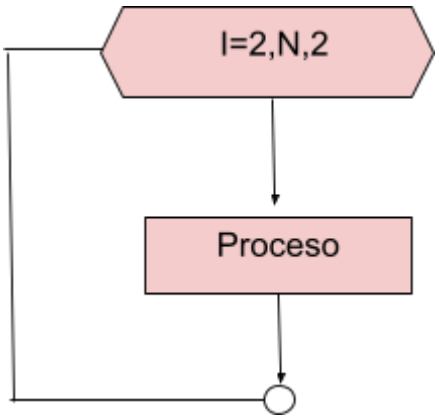
El bloque de instrucciones que se va a repetir es lo que se conoce como cuerpo del bucle o proceso y cada repetición se denomina interacción o secuencia.

Ejemplo:



Sin embargo, el inicio no tiene porqué ser necesariamente 1, ni el aumento tiene que ser de una única unidad.

Ejemplo:



En este caso, el inicio se daría en 2 y el aumento se produciría de 2 en 2.

¿Cómo se realiza en R?
De forma general lo escribiríamos de la siguiente forma:

```
for (I in 1:N){
Proceso
}
```

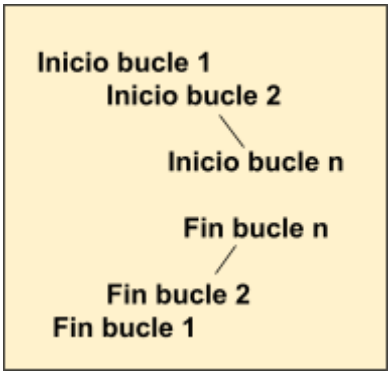
¿Cuando utilizo while y cuando utilizo if?

Generalmente, siempre que tengamos que usar if, nos impondrán una serie de **condiciones** y nos darán los "procesos" que ocurren si se cumplen o no. En cambio, normalmente cuando nos piden utilizar while en el enunciado nos dirán que tenemos que hacer algo **mientras que o hasta que** ocurra otra cosa.

BUCLES ANIDADOS

En caso de tener varios bucles for seguidos hablamos de bucles anidados.

Para la elaboración de bucles anidados es necesario que se cumpla la siguiente condición:

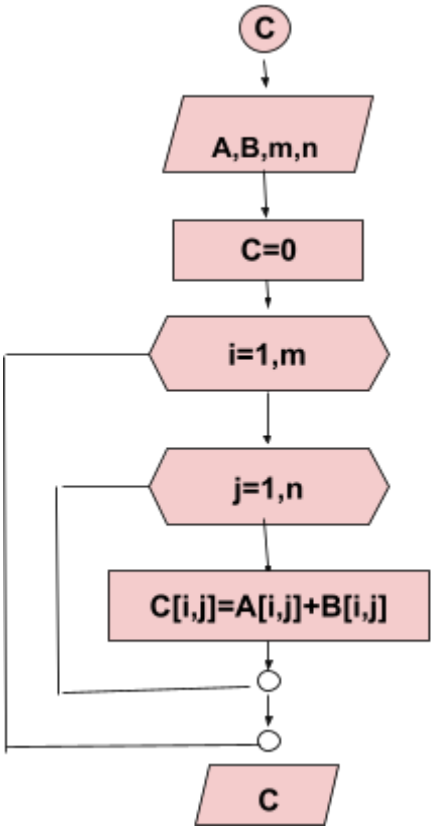


Biotecnología Programación 2023-2024
1º curso Grupo M10

Dificultad: media-alta
Revisado por M2

Ejemplo de bucle anidado:

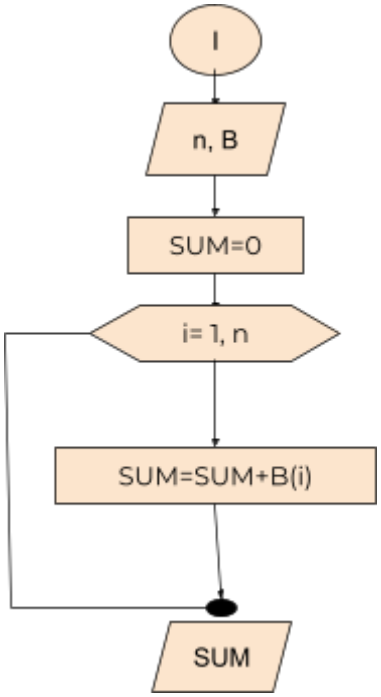
Para la elaboración de matrices es necesario realizar bucles anidados:



SUMATORIOS

Cuando veamos una fórmula de este estilo, la mejor manera de plasmarlo en código es con un bucle for. Como el sumatorio suma muchos términos cuya variación reside en una única variable (n, i, etc) el bucle for es ideal.

$$SUM=\sum_{i=1}^{n} B(i)$$



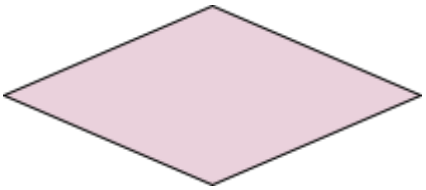
Para empezar, creamos la variable en la que almacenaremos el resultado con valor inicial 0 (el término neutro de la suma). Luego iniciamos el buble for, y por cada valor de la variable del bucle sumamos a la del resultado lo que indique la formula del sumatorio.

Esto es también aplicable a productorios, solo que el valor inicial de la variable del resultado tiene que ser 1 y en el bucle multiplicamos, no sumamos.

Biotecnología Programación 2023-2024 Dificultad: media-alta
1º curso Grupo M10 Revisado por M2

ESTRUCTURA CONDICIONAL IF

En primer lugar, debemos saber cómo es su estructura:

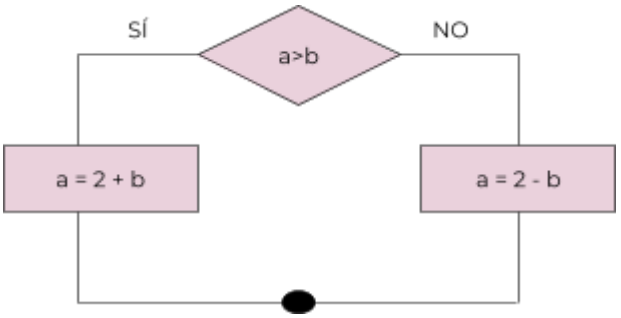


- Algunas **variables lógicas** que le pueden acompañar son:
  - **T=True** ⇨ verdadero
  - **F=False** ⇨ falso

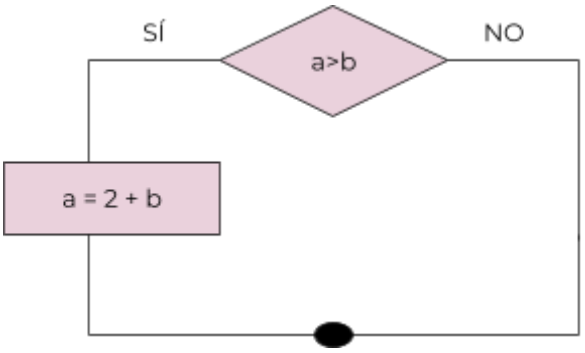
- Las **expresiones** que nos pueden ayudar a establecer **relación** son:
  - **>** ⇨ mayor que
  - **>=** ⇨ mayor o igual
  - **<** ⇨ menor
  - **<=** ⇨ menor o igual
  - **==** ⇨ igual

- Si queremos que **dentro de una misma condición** tengan que ocurrir 2 cosas diferentes, podemos utilizar
  - **AND** ⇨ si queremos que se cumplan esas dos cosas
  - **OR** ⇨ si se cumple alguna de las dos.

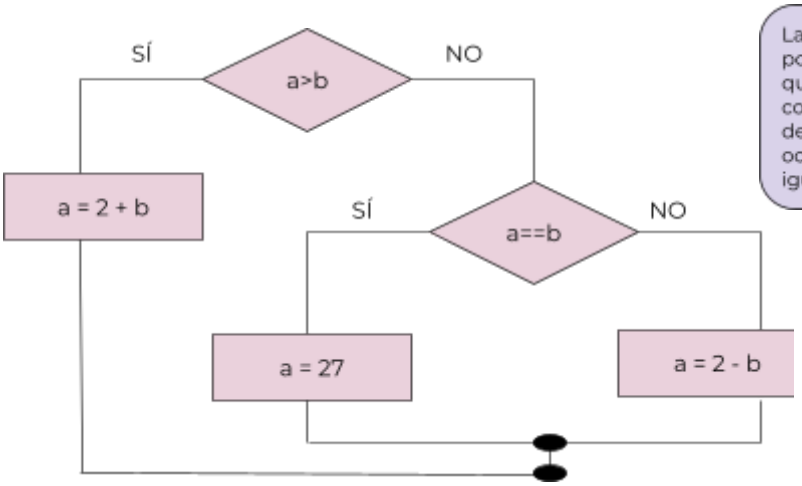
Ahora vamos a ver cómo se realizaría el algoritmo y cómo se expresaría en R.
A partir de la condición que vamos a establecer (dentro del rombo) va a surgir una bifurcación, es decir, dos ramas, dependiendo si se cumple esa condición o no.



También, estos bucles nos permiten solo poner la rama del **SÍ**, es decir, esta rama tiene que aparecer obligatoriamente, mientras que la rama del **NO** puede encontrarse vacía. En este caso, se expresaría así:



También podemos poner **varias condiciones if** relacionadas de forma que si no se cumple una condición, podamos volver a establecer otra estructura condicional if, de la siguiente forma.



Las ramas de "SÍ" y "NO", las podemos poder en el lado que queramos (siempre que concuerde, es decir, la rama del SÍ conduzca al proceso que ocurre si la respuesta es sí, e igual con la rama del NO)

¿Cómo se realiza en R?

Partiendo de la estructura anterior solo tenemos que añadir algunas cosas.

De forma general:

```
if (Condición 1){
 Proceso 1
}else if (condición 2) {
proceso 2
}else{
  proceso 3
}
```

En este caso:

```
if (a>b){
 a= 2 + b
}else if (a==b) {
a=27
}else{
  a= 2 - b
}
```

Incluimos cómo se realiza el organigrama tanto de forma visual, ya que creemos que así es más fácil de entender, cómo en R, ya que de cara a las prácticas va a ser muy útil, y además consideramos que ayuda a entenderlo mejor y a afianzar el concepto.

Sentencia if en sumatorios

Las sentencias if también pueden estar presentes en sumatorios si poseen alguna condición de **a≠b.**

Veamos un ejemplo:

$$SUM = \sum_{\substack{i=3 \\ i \neq 4}}^{n} B(i)$$