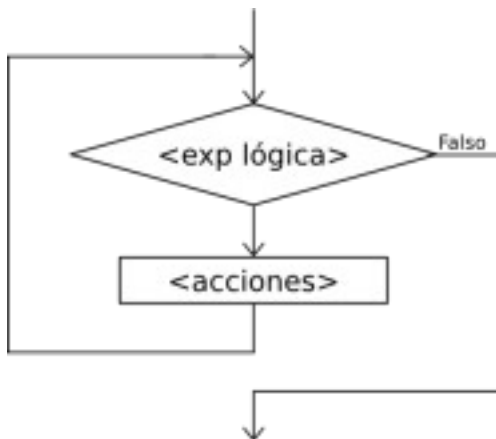


Estructuras repetitivas: Mientras



La instrucción **Mientras** ejecuta una secuencia de instrucciones mientras una condición sea verdadera.

```
Mientras <condición> Hacer
    <instrucciones>
FinMientras
```

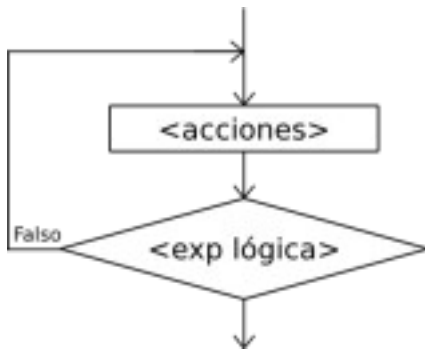
- Al ejecutarse esta instrucción, la condición es evaluada. Si la condición resulta verdadera, se ejecuta una vez la secuencia de instrucciones que forman el cuerpo del ciclo. Al finalizar la ejecución del cuerpo del ciclo se vuelve a evaluar la condición y, si es verdadera, la ejecución se repite. Estos pasos se repiten mientras la condición sea verdadera.
- Se puede dar la circunstancia que las instrucciones del bucle no se ejecuten nunca, si al evaluar por primera vez la condición resulta ser falsa.
- Si la condición siempre es verdadera, al ejecutar esta instrucción se produce un ciclo infinito. A fin de evitarlo, las instrucciones del cuerpo del ciclo deben contener alguna instrucción que modifique la o las variables involucradas en la condición, de modo que ésta sea falsificada en algún momento y así finalice la ejecución del ciclo.

Ejemplo:

Crea un programa que pida al usuario una contraseña, de forma repetitiva mientras que no introduzca “asdasd”. Cuando finalmente escriba la contraseña correcta, se le dirá “Bienvenido” y terminará el programa.

```
Proceso login
    Definir secreto, clave como cadena;
    secreto <- "asdasd";
    Escribir "Dime la clave:";
    Leer clave;
    Mientras clave<>secreto Hacer
        Escribir "Clave incorrecta!!!";
        Escribir "Dime la clave:";
        Leer clave;
    FinMientras
    Escribir "Bienvenido!!!";
    Escribir "Programa terminado";
FinProceso
```

Estructuras repetitivas: Repetir - Hasta Que



La instrucción **Repetir -Hasta Que** ejecuta una secuencia de instrucciones hasta que la condición sea verdadera.

Repetir

<instrucciones>

Hasta Que <condición>

- Al ejecutarse esta instrucción, la secuencia de instrucciones que forma el cuerpo del ciclo se ejecuta una vez y luego se evalúa la condición. Si la condición es falsa, el cuerpo del ciclo se ejecuta nuevamente y se vuelve a evaluar la condición. Esto se repite hasta que la condición sea verdadera.
- Note que, dado que la condición se evalúa al final, las instrucciones del cuerpo del ciclo serán ejecutadas al menos una vez.
- Además, a fin de evitar ciclos infinitos, el cuerpo del ciclo debe contener alguna instrucción que modifique la o las variables involucradas en la condición de modo que en algún momento la condición sea verdadera y se finalice la ejecución del ciclo.

Ejemplo:

Crea un programa que pida al usuario una contraseña, de forma repetitiva mientras que no introduzca “asdasd”. Cuando finalmente escriba la contraseña correcta, se le dirá “Bienvenido” y terminará el programa.

Proceso login

Definir secreto, clave como cadena;

secreto <- "asdasd";

Repetir

 Escribir "Dime la clave:";

 Leer clave;

 Si clave<>secreto Entonces

 Escribir "Clave incorrecta!!!";

 FinSi

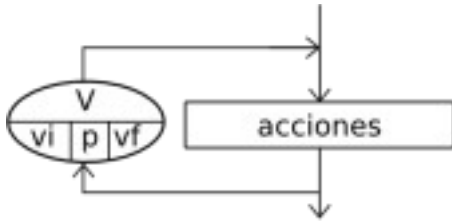
Hasta Que clave=secreto

 Escribir "Bienvenido!!!";

 Escribir "Programa terminado";

FinProceso

Estructuras repetitivas: Para



La instrucción **Para** ejecuta una secuencia de instrucciones un número determinado de veces.

```
Para <variable> <- <inicial> Hasta <final> [Con Paso <paso>] Hacer
    <instrucciones>
FinPara
```

- Al ingresar al bloque, la variable **<variable>** recibe el valor **<inicial>** y se ejecuta la secuencia de instrucciones que forma el cuerpo del ciclo.
- Luego se incrementa la variable **<variable>** en **<paso>** unidades y se evalúa si el valor almacenado en **<variable>** superó al valor **<final>**.
- Si esto es falso se repite hasta que **<variable>** supere a **<final>**.
- Si se omite la cláusula **Con Paso <paso>**, la variable **<variable>** se incrementará en 1.

Ejemplo

Escribir en pantalla del 1 al 10.

```
Proceso Contar
    Definir var como Entero;
    Para var<-1 Hasta 10 Hacer
        Escribir Sin Saltar var, " ";
    FinPara
FinProceso
```

Escribir en pantalla de 10 al 1.

```
Proceso ContarDescendente
    Definir var como Entero;
    Para var<-10 Hasta 1 Con Paso -1 Hacer
        Escribir Sin Saltar var, " ";
    FinPara
FinProceso
```

Escribir los número pares desde el 2 al 10.

```
Proceso ContarPares
    Definir var como Entero;
    Para var<-2 Hasta 10 Con Paso 2 Hacer
        Escribir Sin Saltar var, " ";
    FinPara
FinProceso
```