

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier  
mirror_ob.select = 0  
= bpy.context.selected_obj  
data.objects[one.name].se  
print("please select exactly
```

```
--- OPERATOR CLASSES ---
```

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

```
context):  
context.active_object is not
```

# ESTRUCTURAS REPETITIVAS

---

ING. CARLOS H. RUEDA C.



# CONTENIDO

---

1. Estructuras repetitivas
  - 1.1. Estructuras con un número determinado de iteraciones
  - 1.2. Estructura con un número indeterminado de iteraciones (mientras)
  - 1.3. Estructura con un número indeterminado de iteraciones (hasta)
  - 1.4. Comparación entre estructuras repetitivas
2. Ejemplos
3. Ejercicios

# 1. Estructuras repetitivas

---

Se utiliza para repetir una instrucción o un grupo de instrucciones de forma cíclica.

Su propósito radica en resolver problemas que implican acciones recurrentes o cíclicas.

# 1. Estructuras repetitivas

---

Se utiliza para repetir una instrucción o un grupo de instrucciones de forma cíclica. Su propósito radica en resolver problemas que implican acciones recurrentes o cíclicas.

# 1. Estructuras repetitivas

---

Hay dos tipos principales de bucles:

1. Con un número determinado de iteraciones.
2. Con un número indeterminado de iteraciones

# 1.1. Estructura con un número determinado de iteraciones

---

Son los casos en los que la estructura se ejecuta un número predeterminado de veces, conocido antes de su ejecución.

# 1.1. Estructura con un número determinado de iteraciones

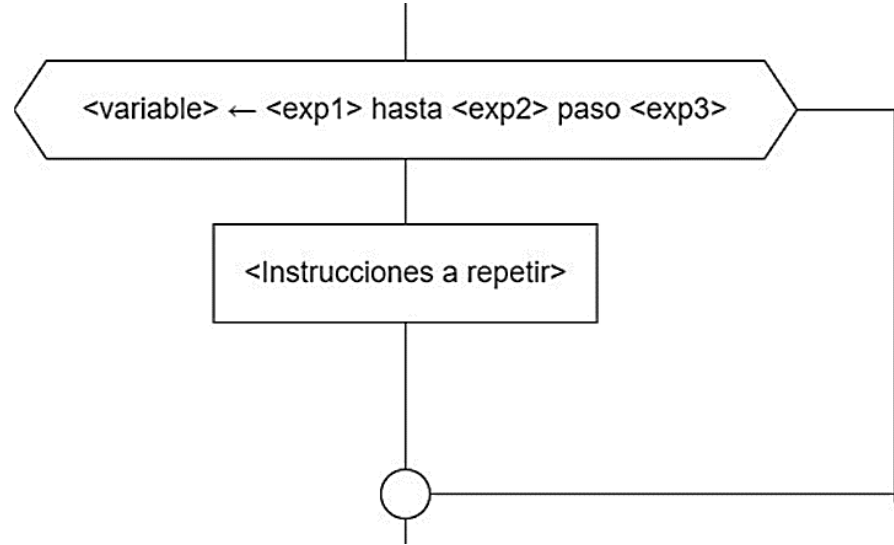
---

Comúnmente, a este tipo de estructuras se le denomina "**para**" (por la palabra "**para**" y el número de iteraciones) o, en inglés, "**for**".

# 1.1. Estructura con un número determinado de iteraciones

---

Estructura en DFD





# 1.1. Estructura con un número determinado de iteraciones

---

Estructura en pseudocódigo

```
Para <var> ← <exp1> hasta <exp2> paso <exp3> haga  
    <instrucciones a repetir>  
Fin-Para
```

## 1.2. Estructura con un número indeterminado de iteraciones (mientras)

---

Son los casos en los que el número de iteraciones no está precisamente definido y depende de un dato dentro del programa.

## 1.2. Estructura con un número indeterminado de iteraciones (mientras)

---

A estas estructuras se les llama comúnmente "Mientras Que" o "while" en inglés.

## 1.2. Estructura con un número indeterminado de iteraciones (mientras)

---

Las estructuras "mientras que" repiten un proceso un número "N" de veces, donde "N" puede ser constante o variable y *está determinado por una condición.*

## 1.2. Estructura con un número indeterminado de iteraciones (mientras)

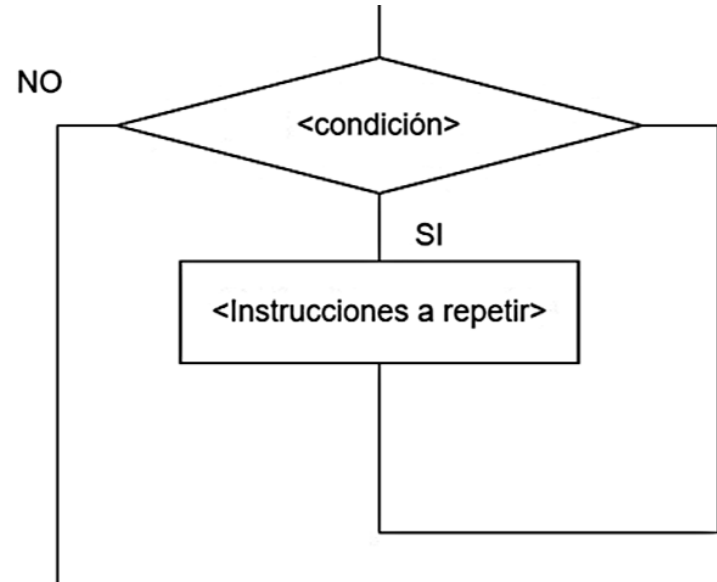
---

Esta condición **debe ser verdadera** para que el proceso continúe ejecutándose. Cuando la condición ya no se cumple, el proceso se detiene.

## 1.2. Estructura con un número indeterminado de iteraciones (mientras)

---

La estructura en DFD



## 1.2. Estructura con un número indeterminado de iteraciones (mientras)

---

La estructura en pseudocódigo

```
Mientras que <condición>  
    <instrucciones a repetir>  
Fin-Mientras
```

## 1.3. Estructura con un número indeterminado de iteraciones (hasta)

---

En las estructuras repetitivas dónde no se conoce con certeza cuántas repeticiones se llevarán a cabo, existe una opción llamada "repita hasta", útil *cuando se sabe que el proceso debe ejecutarse al menos una vez.*



## 1.3. Estructura con un número indeterminado de iteraciones (hasta)

---

En inglés, este tipo de estructura se denomina comúnmente "do while".

## 1.3. Estructura con un número indeterminado de iteraciones (hasta)

---

A diferencia de la estructura "Mientras Que", la estructura "Repita Hasta" lo hace hasta que se cumple una condición, no mientras esta sea verdadera.

## 1.3. Estructura con un número indeterminado de iteraciones (hasta)

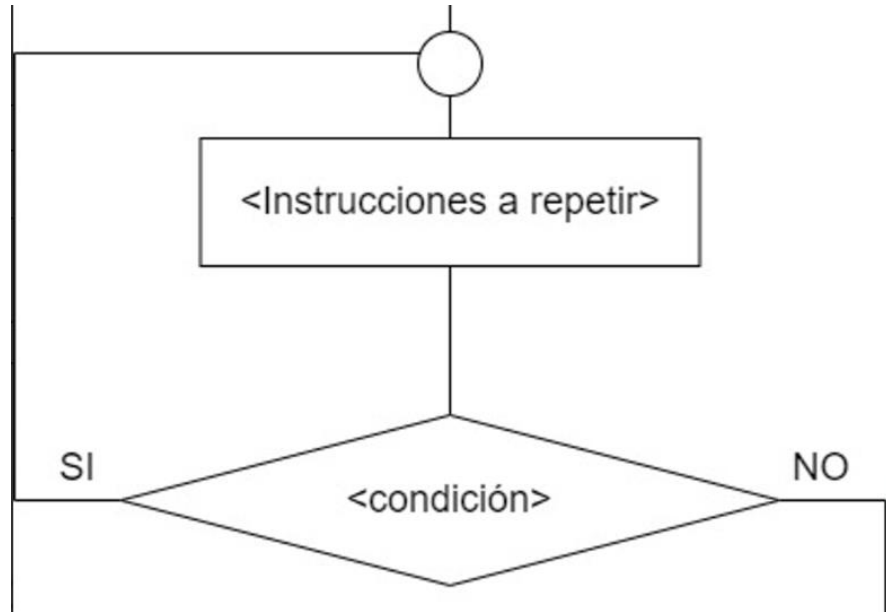
---

La característica clave aquí es que esta estructura asegura que el proceso se realice al menos una vez, ya que la condición se evalúa al final del proceso.

## 1.3. Estructura con un número indeterminado de iteraciones (hasta)

---

Estructura en DFD



## 1.3. Estructura con un número indeterminado de iteraciones (hasta)

---

Estructura en Pseudocódigo

```
Repita  
    <instrucciones a repetir>  
Hasta <condición>
```

# 3. Ejemplos

---

## **Estructura repetitiva para (for)**

Escribe un programa que imprima los números del 1 al 10 en la consola, uno por línea.

# 3. Ejemplos

---

## Estructura repetitiva para (for)

Crea un programa que pida al usuario ingresar un número entero y luego imprima la tabla de multiplicar de ese número del 1 al 10.

# 3. Ejercicio

---

## Estructura repetitiva para (for)

Desarrolla un programa que calcule la suma de los primeros 50 números enteros positivos.



# 3. Ejemplos

---

## **Estructura repetitiva mientras (while)**

Escribe un programa que pida al usuario ingresar un número y continúe pidiendo números mientras el número ingresado sea menor que 100..

# 3. Ejemplos

---

## **Estructura repetitiva mientras (while)**

Crea un programa que solicite al usuario una contraseña y continúe solicitándola hasta que el usuario ingrese la contraseña correcta.

# 3. Ejercicio

---

## Estructura repetitiva mientras (while)

Desarrolla un programa que calcule la suma de los dígitos de un número entero positivo ingresado por el usuario. El programa debe seguir ejecutándose hasta que el número ingresado sea 0.

# 3. Ejemplo

---

## **Estructura repetitiva repita hasta (do-while)**

Escribe un programa que pida al usuario ingresar un número y repita la solicitud hasta que el usuario ingrese un número par.

# 3. Ejemplo

---

## **Estructura repetitiva repita hasta (do-while)**

Crea un programa que pida al usuario ingresar su nombre y repita la solicitud hasta que el usuario ingrese un nombre que tenga al menos 5 caracteres.

# 3. Ejemplo

---

## **Estructura repetitiva repita hasta (do-while)**

Desarrolla un programa que genere un número aleatorio entre 1 y 100, y luego pida al usuario adivinarlo. El programa debe continuar solicitando al usuario adivinar el número hasta que lo haga correctamente. El programa debe darle pista diciéndole si es mayor o menor.

# 4. Ejercicios

---