

Ciclos y contadores

Ciclos y contadores	1
¿Qué aprenderás?	2
Introducción	2
Iterar	3
Contando con while	3
¿Qué significa <code>i+= 1</code> ?	4
Operadores de asignación	4
Ejercicio: La bomba de tiempo	5
Contando segundos	5
Cuidado con las condiciones de borde	6



¡Comencemos!

¿Qué aprenderás?

- Conocer el concepto de iteración.
- Contar la cantidad de veces que un programa está dentro de un ciclo.
- Realizar programas donde el usuario ingrese múltiples datos hasta que decida detenerse

Introducción

En los capítulos anteriores resolvimos ejercicios de ciclos donde el fin del ciclo estaba determinado por el ingreso de un dato por parte del usuario.

En este capítulo resolveremos problemas que requieren una cantidad determinada de ciclos. Por ejemplo un programa que cuente de cero a diez o un problema de sumatorias.

¡Vamos con todo!



Iterar

Iterar es dar una vuelta al ciclo. Hay muchos problemas que se pueden resolver iterando. Por ejemplo repetir un mensaje 10 veces o contar desde cero hasta 10.

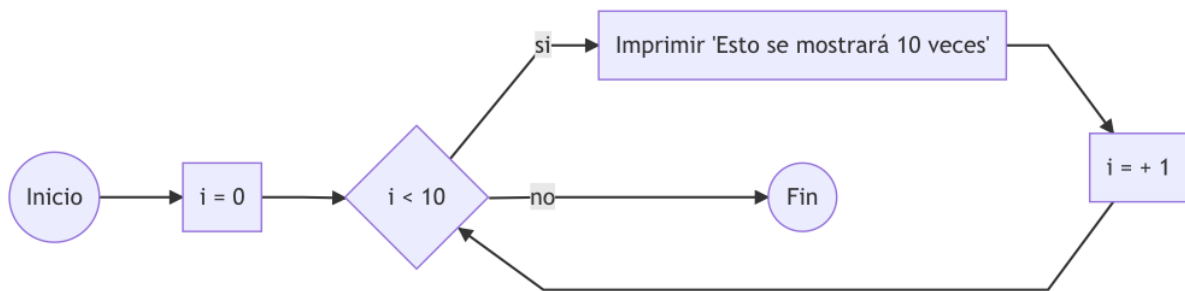


Imagen 1. Ciclo contador.
Fuente: Desafío Latam.

Contando con while

```
i = 0
while i < 10
  puts "Esto se mostrará 10 veces" # Código que queremos repetir.
  i += 1 # IMPORTANTE
end
```

La instrucción `puts "Esto se mostrará 10 veces"` se repetirá hasta que la variable `i` alcance el valor 10.

Para entonces, la comparación de la instrucción `while` se evaluará como `false` y saldremos del ciclo.

En programación, es una convención ocupar una variable llamada `i` como variable de iteración para operar en un ciclo.



IMPORTANTE: Si no aumentamos el valor de la variable `i` entonces nunca llegará a ser igual o mayor a 10, por ende, la comparación nunca se evaluará como `false` y entraremos en un ciclo infinito.

¿Qué significa `i+= 1`?

`+=` es un operador de asignación, muy similar a decir `a = 2` pero la diferencia es que con `+=` estamos diciendo el valor anterior más 1.

```
a = 2
a += 2 #aquí el valor de 'a' aumentó en dos y fue almacenado nuevamente
      en 'a'

puts a
# 4
```

Esto es lo mismo que escribir:

```
a = 2
a += 2
puts a
```

Operadores de asignación

La siguiente tabla muestra el comportamiento de los operadores de asignación:

Operador	Nombre	Ejemplo	Resultado
=	Asignación	a = 2	a toma el valor 2.
+=	Incremento y asignación	a += 2	a es incrementado en dos y asignado el valor resultante.
-=	Decremento y asignación	a -= 2	a es reducido en dos y asignado el valor resultante.
*=	Multiplicación y asignación	a *= 3	a es multiplicado por tres y asignado el valor resultante.
/=	División y asignación	a /= 3	a es dividido por tres y asignado el valor resultante.

Tabla 1. Operadores de asignación.

Fuente: Desafío Latam.

Ejercicio: La bomba de tiempo

Crearemos un algoritmo sencillo que realice una cuenta regresiva de 5 segundos.

Contar de forma regresiva es muy similar, solo debemos comenzar desde el valor correspondiente e ir disminuyendo su valor de uno en uno.

```
i = 5
while(i > 0)
  i -= 1
  puts i
end
```

```
4
3
2
1
0
```

Contando segundos

Existe una instrucción llamada `sleep` que nos permite esperar un tiempo determinado antes de continuar.

```
i = 5
while(i > 0) # Cuando llegue a cero terminamos.
  i -= 1 # En cada iteración descontamos 1.
  puts i
  sleep 1
end
```

```
4
3
2
1
0
```

Cuidado con las condiciones de borde

No es lo mismo.

```
i = 5
while(i > 0)
  i -= 1
  puts i
  sleep 1
end
```

Que:

```
i = 5
while(i >= 0) # el valor cero cumple la condición
  i -= 1
  puts i
  sleep 1
end
```