ITEDES

Educación Digital

Módulo:

Fundamentos de Ingeniería de Software

Segmento:

Algoritmos y Estructuras de Datos

Tema:

Estructura de Repetición Controlada por Centinela

Prof. Germán C. Basisty
german.basisty@itedes.com

Índice de Contenidos

Estructuras de Repetición Controladas por Centinela

Como ya hemos visto con anterioridad, las **estructuras de repetición** son estructuras de control de flujo que sirven para ejecutar un bloque de código de forma repetitiva.
En las *estructuras de repetición controladas por centinela*, a diferencia de las estructuras de repetición controladas por contador, la cantidad de ciclos en los cuales se iterará sobre un bloque de código no estará determinada al momento de comenzar dicho ciclo, sinó que el mecanismo de salida de la estructura depende de una condición de verdad que se evalúa en cada repetición.
Existen dos *estructuras de repetición controladas por centinela*:

- Estructura **WHILE** (MIENTRAS)
- Estructura **DO WHILE** (REPETIR)

Estructura WHILE (MIENTRAS)

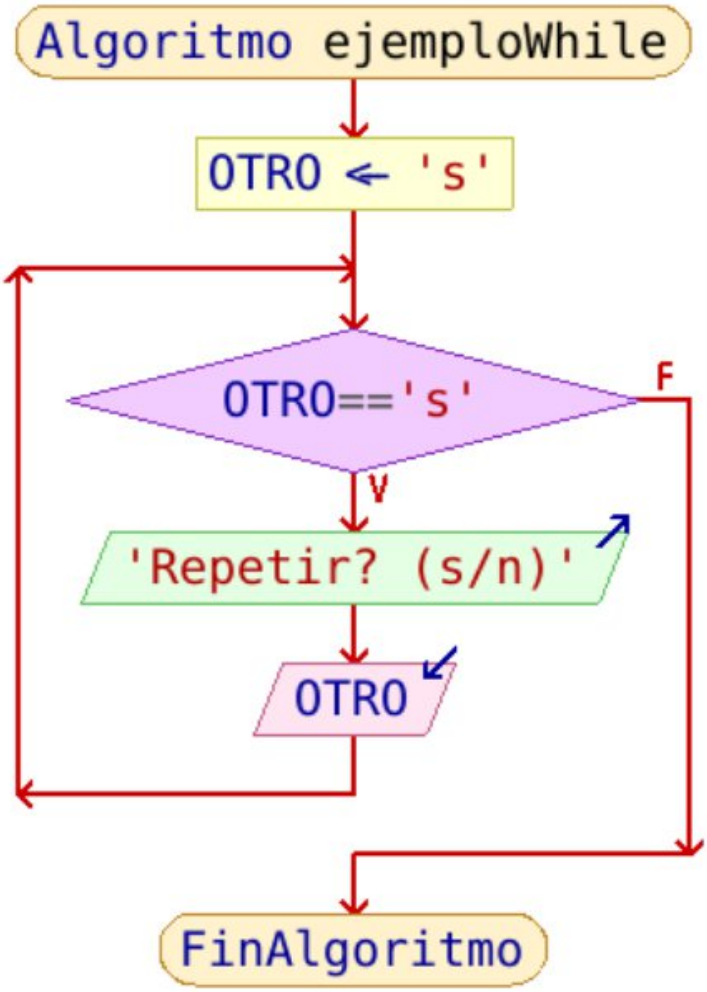
La estructura de repetición WHILE ejecuta un bloque de código siempre que la condición sea verdadera, siendo que la evaluación de la condición de verdad se realiza al principio de cada ciclo.

Ejemplo

Pseudocódigo

```
Algoritmo ejemploWhile
	otro = "s"
	Mientras otro == "s" Hacer
		Escribir "Repetir? (s/n)"
		Leer otro
	Fin Mientras
FinAlgoritmo
```

Diagrama de flujo



BASH

```
#!/bin/bash

declare continue="n"

while :
do
read -p "Continuar? (s/n): " continue
[[ “$continue” == "s" ]] || break
done

exit 0
```

Python

```
otro = "s"

while otro == "s":
        otro = input("Repetir? (s/n) ")
```

Java

```
import java.util.Scanner;

public class Ejemplo {
   public static void main(String args[]) {
      Scanner teclado = new Scanner(System.in);

      String otro = "s";

      while(otro.equals("s")) {
         System.out.print("Repetir (s/n): ");
         otro = teclado.nextLine();
      }
   }
}
```

C#

```
using System;

namespace cs
{
   class Program
   {
      static void Main(string[] args)
      {
         string otro = "s";

         while(otro == "s")
         {
            Console.Write("Repetir (s/n): ");
            otro = Int32.Parse(Console.ReadLine());
         }
      }
   }
}
```

JavaScript

```
function ejemplo() {
    let otro = 's';

    while(otro == 's') {
        otro = prompt('Repetir (s/n)')
    }
}
```

Estructura DO WHILE (REPETIR)

La estructura de repetición DO WHILE ejecuta un bloque de código siempre que la condición sea verdadera, siendo que la evaluación de la condición de verdad se realiza al final de cada ciclo. Esto garantiza que el bloque de código se ejecute por lo menos una vez.

Python y BASH no soportan la estructura DO WHILE de forma nativa, pero se puede emular en caso que sea necesario

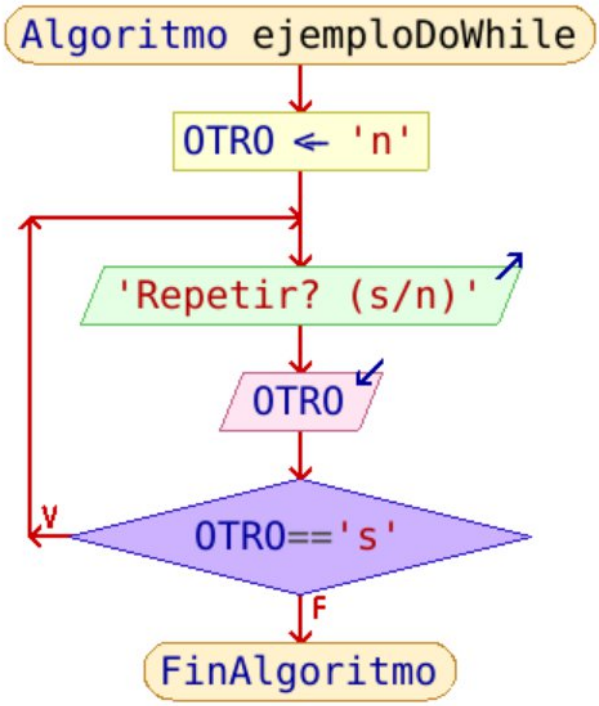
Ejemplo

Pseudocódigo

```
Algoritmo ejemploDoWhile
        otro = "n"

        Repetir
                Escribir "Repetir? (s/n)"
                Leer otro
        Mientras que otro == "s"
FinAlgoritmo
```

Diagrama de Flujo



Java

```
import java.util.Scanner;

public class Ejemplo {
    public static void main(String args[]) {
        Scanner teclado = new Scanner(System.in);

        String otro = "n";

        do {
            System.out.print("Repetir (s/n): ");
            otro = teclado.nextLine();
        } while(otro.equals("s"));
    }
}
```

C#

```
using System;

namespace cs
{
    class Program
    {
        static void Main(string[] args)
        {
            string otro = "n";

            do
            {
                Console.Write("Repetir (s/n): ");
                otro = Int32.Parse(Console.ReadLine());
            } while(otro == "s");
        }
    }
}
```

JavaScript

```
function ejemplo() {
    let otro = 'n';

    do {
        otro = prompt('Repetir (s/n)')
    } while(otro == 's');
}
```

Ejercitación

1) Pedir números por pantalla hasta que alguno sea mayor a 100. Presentar diagrama de flujo, pseudocódigo y código fuente funcionando en BASH, Python, Java, C# y HTML + JavaScript.

2) Desarrollar un algoritmo que solicite al usuario dos números y muestre el resultado de la suma de esos números por pantalla. Repetir esta operación mientras que el usuario así lo disponga. Presentar diagrama de flujo, pseudocódigo y código fuente funcionando en BASH, Python, Java, C# y HTML + JavaScript.

3) Escribir un programa que pida por teclado la base y la altura de un rectángulo. Verificar que tanto la base como la altura sean mayores que cero, y en caso contrario, repetir el ingreso de datos hasta que el valor sea correcto. Mostrar el área del rectángulo (base * altura). Presentar diagrama de flujo, pseudocódigo y código fuente funcionando en BASH, Python, Java, C# y HTML + JavaScript.

4) Desarrollar un algoritmo que lea números de forma indefinida hasta que se ingrese un cero. Mostrar la suma de todos los números ingresados. Presentar diagrama de flujo, pseudocódigo y código fuente funcionando en BASH, Python, Java, C# y HTML + JavaScript.

5) Desarrollar un algoritmo que lea números enteros positivos de forma indefinida hasta que se ingrese un cero. Mostrar cual es el mayor. Presentar diagrama de flujo, pseudocódigo y código fuente funcionando en BASH, Python, Java, C# y HTML + JavaScript.