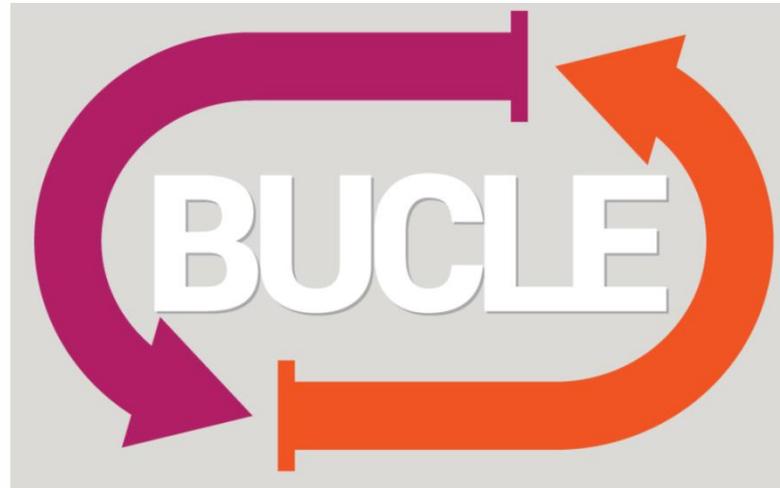


The logo for Silicon Misiones features the text "Silicon Misiones" in a bold, white, sans-serif font. The text is centered within a horizontal, rounded rectangular shape composed of several overlapping, semi-transparent colored areas: a red circle on the left, a light blue shape above the text, a green shape below the text, and a blue circle on the right. The background is white with green abstract shapes in the top-left and bottom-right corners.

Silicon Misiones

Misiones, República Argentina

INTRODUCCIÓN A LA PROGRAMACIÓN



CLASE 15/06

QUÉ ES UN BUCLE?

- Un bucle o ciclo, en programación, es una secuencia de instrucciones de código que se ejecuta repetidas veces, hasta que la condición asignada a dicho bucle deja de cumplirse



Ejemplo: si tengo que lavar los platos de la cena, realizaré una repetición de "lavar 1 plato", tantas veces como platos tenga para lavar:

Lavar1plato

Lavar1plato

.....

Lavar1plato

QUÉ ES UN BUCLE?

- ❑ En programación realizamos continuamente bucles para implementar algoritmos de todo tipo, desde los más básicos hasta los más complejos. Encontramos bucles en la programación, pero también en la vida real.

Ejemplo: Para subir una escalera, realizaré la repetición de "subir un escalón", tantas veces como escalones haya en la escalera.

subirunescalon

subirunescalon

.....

subirunescalon



POR QUÉ SON ÚTILES LOS BUCLES?

- En informática es habitual que las repeticiones se hagan decenas, cientos o miles de veces, por lo que no vamos a escribir la misma sentencia hasta aburrirnos, pues sería un gasto de tiempo innecesario y el código de los programas sería muy largo.

mientras que queden platos

lavar1plato

fin mientras

repetir

subir1escalon

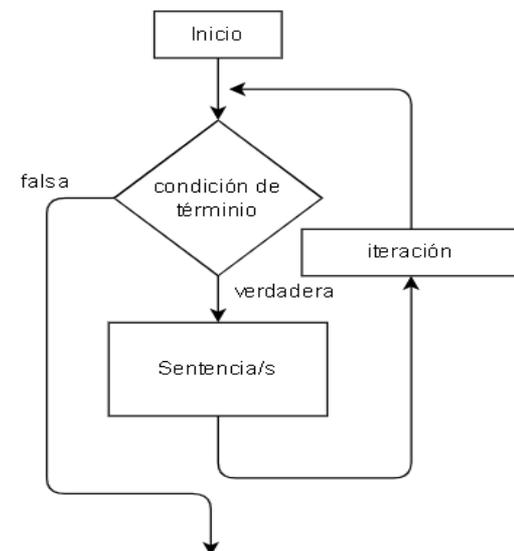
Hasta que terminen los escalones

CUÁLES SON LAS PARTES DE UN BUCLE?

- ❑ En programación debemos especificar de manera precisa todos los pasos necesarios para completar una tarea. Por eso, a la hora de programar es importante tener claro cómo definir un bucle, para que podamos implementarlos correctamente al escribir los algoritmos para nuestros programas.

Los bloques principales de un bucle:

- Definición de las **características de la repetición**
- **Cuerpo de la repetición**, que son las sentencias que se van a repetir.



CUERPO DEL BUCLE

- El cuerpo del bucle es muy fácil de identificar. Contiene una o más sentencias que necesitamos repetir en cada ejecución del bucle. Por ejemplo, para el caso de los platos, quizás la repetición incluya varias sentencias si somos más detallistas.

```
1  Proceso elwhile
2      escribir "Ingrese un número"
3      leer numero
4      c<-1
5      Mientras c<=numero Hacer
6          Imprimir "Vamos en el giro ", c
7          c<-c+1
8      Fin Mientras
9  FinProceso
```

Centinela que inicia en 1

Pregunta que controla el bucle

Incrementamos el centinela para permitir el fin del bucle

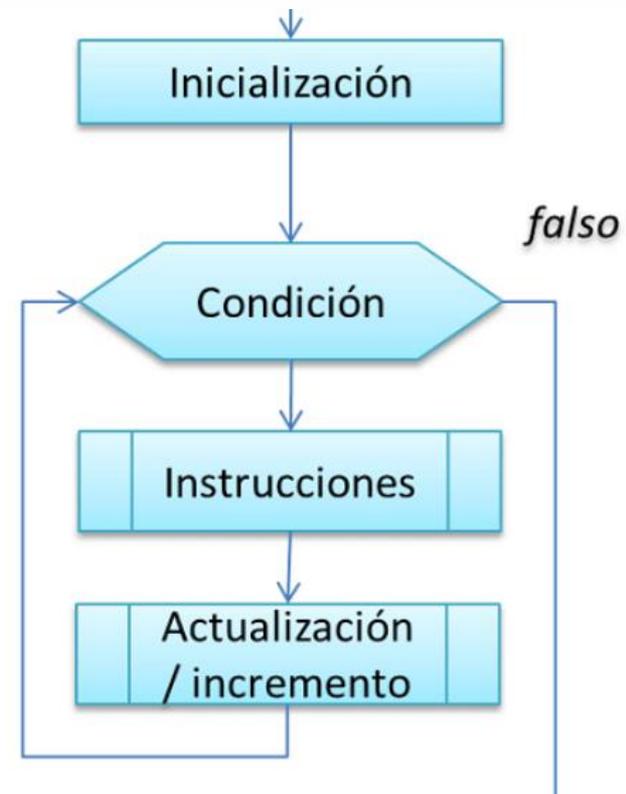
CARACTERÍSTICAS DE LA REPETICIÓN

- ❑ Nos permiten controlar el número de veces que se ejecutarán las sentencias del cuerpo del bucle, es decir nos permiten definir cuántas veces se va a iterar, o si se va a repetir el bucle hasta que se cumpla cierta condición.

```
Mientras contador ≤ num Hacer
    resultado = resultado * contador
    contador = contador +1
Fin Mientras
```

CARACTERÍSTICAS DE LA REPETICIÓN

- ❑ **Inicialización del bucle**
- ❑ **Condición** que debe cumplirse para continuar repitiendo el bucle
- ❑ Incremento, decremento, o **varianza** a realizar para cada repetición.



POR QUÉ NECESITAMOS TODO ESTO?

Recordemos que para definir algoritmos es necesario dar instrucciones precisas a los ordenadores. Si yo le pido a alguien "lava los platos de la pila", esa persona sabe de manera implícita que tendrá que comenzar por el primer plato y terminar con el último. Sin embargo, un ordenador no da por entendido nada, con lo que debemos ser muy precisos.



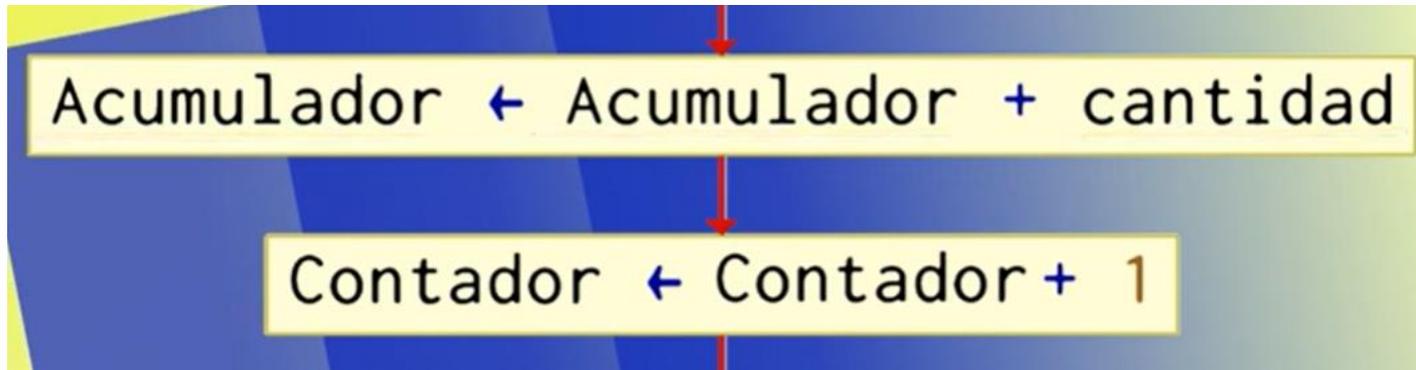
POR QUÉ NECESITAMOS TODO ESTO?

En los casos más básicos la cuenta de las repeticiones la llevaremos con una variable, que iremos **incrementando** en cada iteración para saber hasta dónde tenemos que repetir las sentencias del cuerpo del bucle.

Contador vs Acumulador

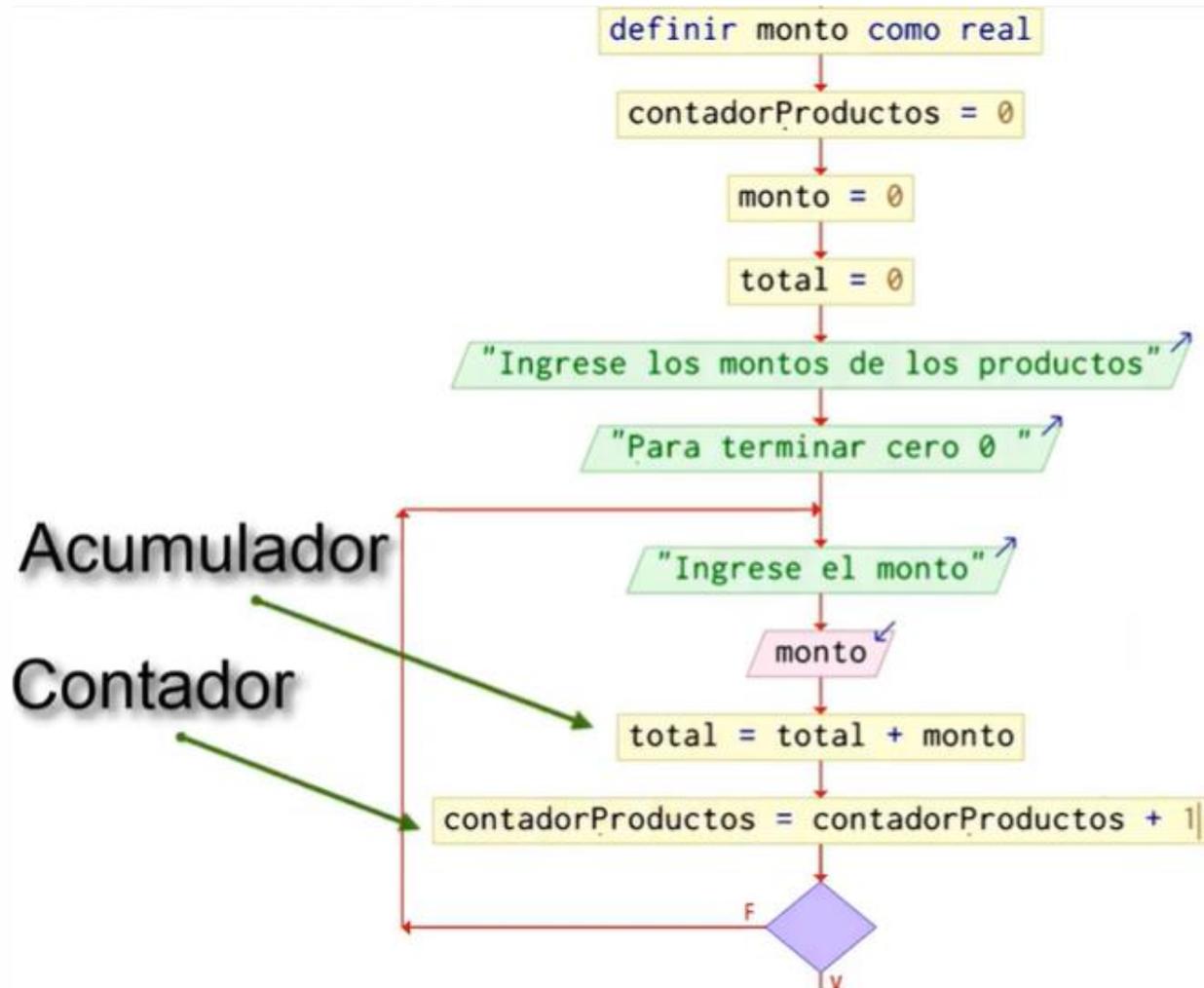


CONTADORES Y ACUMULADORES

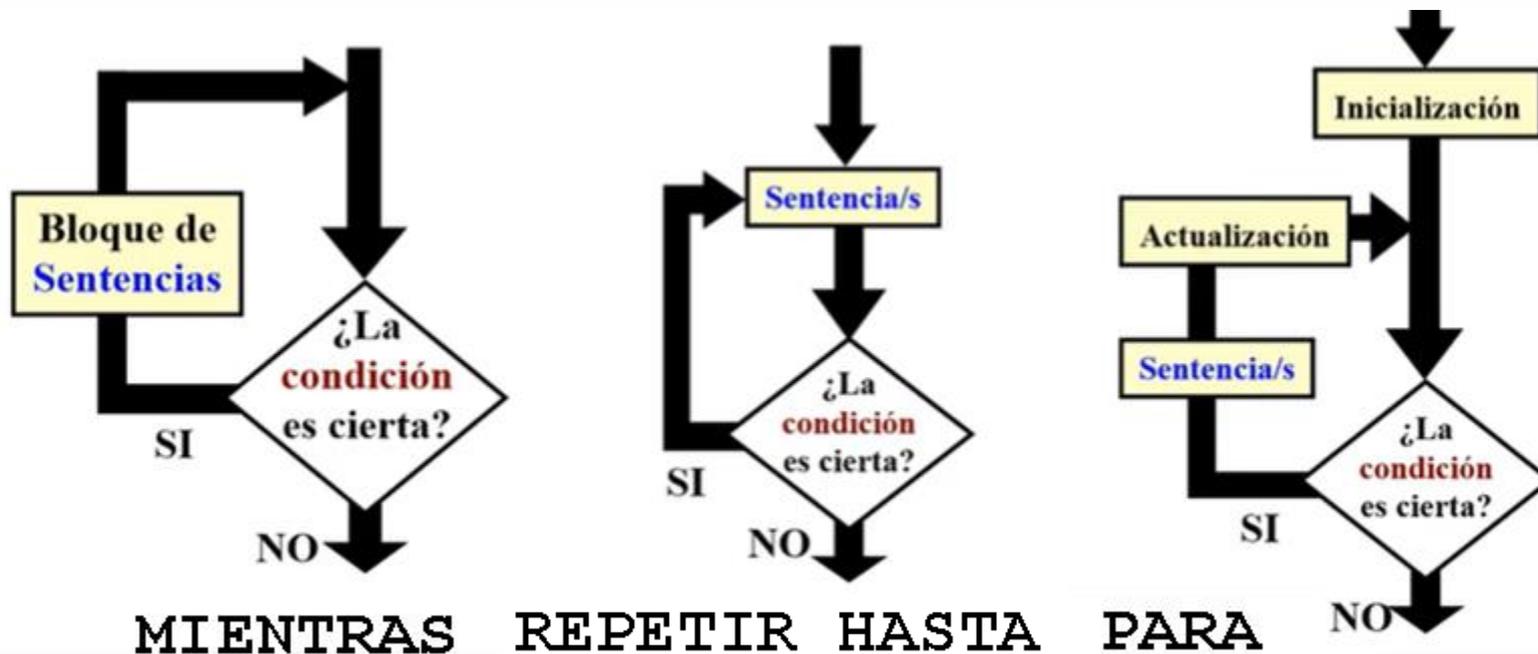


- ❑ Un **CONTADOR** es una variable cuyo valor se incrementa o decrementa en una cantidad constante.
- ❑ Un **ACUMULADOR** es una variable cuyo valor se incrementa o decrementa en una cantidad variable

CONTADORES Y ACUMULADORES



ESTRUCTURAS REPETITIVAS

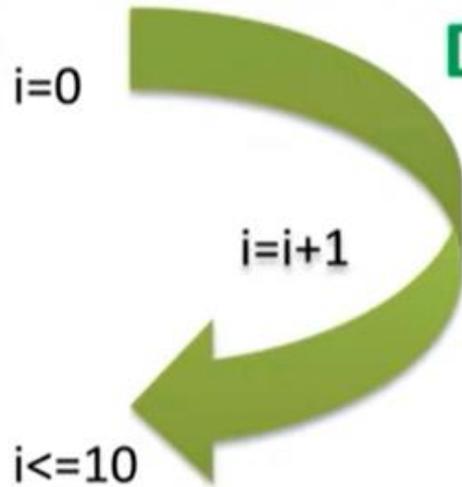


- ❑ En el **mientras** se evalúa la condición al principio, antes de entrar al ciclo. En **repetir hasta** se evalúa al final, por lo cual una vez toca hacer todo el bloque de acciones.

BUCLES

Ciclo Para, Ciclo Mientras, Ciclo Repetir Hasta

Diferencias y Ejercicios



ESTRUCTURA MIENTRAS

- ❑ Este tipo de ciclo se denomina **mientras** o en ingles **do-while**. Esto implica que vamos a repetir una acción determinada mientras algo ocurra, cuando eso que evaluamos deja de ocurrir, ya no seguimos repitiendo la acción. Incluso, si nunca ocurre, nunca la realizamos

Cuando llegamos a este paso de la receta debemos realizar dos acciones: Probar lo que estamos preparando y si necesita sal, agregarle sal. Probamos y si sigue necesitando le volvemos a poner sal.

```
Algoritmo CICLO_FALTA_SAL
```

```
Mientras LE_FALTA_SAL? Hacer
```

```
    // TAREAS A REALIZAR DENTRO DEL CICLO
```

```
Fin Mientras
```

```
FinAlgoritmo
```

ESTRUCTURA REPETIR HASTA

- ❑ Este ciclo se llama **hasta o repetir**, en ingles **DO-UNTIL**, implica justamente eso. Repetimos una acción, por lo menos una vez, hasta que se cumpla una condición.

```
Algoritmo CICLO_DERRETIR_CHOCOLATE
```

```
Repetir
```

```
    // TAREAS A REALIZAR DENTRO DEL CICLO
```

```
Hasta Que SE_DERRITIO_EL_CHOCOLATE?
```

```
FinAlgoritmo
```

ESTRUCTURA PARA

- ❑ Este tipo de ciclo se lo denomina **para** o en ingles **for next**, que implica que vamos a realizar una acción una determinada cantidad de veces. Básicamente se usa cuando **se sabe cuántas veces** debe realizarse algo.

```
Algoritmo CICLO_HUEVOS
  CANTIDAD_HUEVOS <- 10
  // cargamos los datos
  Para HUEVOS<-1 Hasta CANTIDAD_HUEVOS Hacer
  ..... // TAREAS A REALIZAR DENTRO DEL CICLO
  FinPara
FinAlgoritmo
```

EJERCICIOS

- Hacer un **algoritmo** en **Pseint** para calcular la suma de los n primeros números.
- Hacer un **algoritmo** en **Pseint** para determinar la media de una lista indefinida de números positivos y mayores a 0, se debe acabar el programa al ingresar el número cero.
- Hacer un **algoritmo** en **Pseint** para calcular la suma de los primeros cien números con un **ciclo repetir**.
- Hacer un **algoritmo** en **Pseint** para calcular la suma de los primeros cien números con un **ciclo mientras**, y luego probar con un ciclo para.

MUCHAS GRACIAS!!

Sigamos practicando...

