

TALLER DE ALGORITMOS

ESTRUCTURA DE CONTROL ITERATIVA

Abel García Nájera

Karen Miranda Campos

Saúl Zapotecas Martínez

Universidad Autónoma Metropolitana **Unidad Cuajimalpa**

26 de octubre de 2023

ESTRUCTURA DE CONTROL ITERATIVA

Iterativa

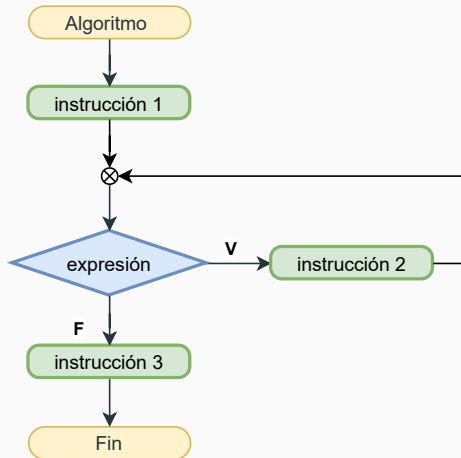
La estructura de control iterativa (o de repetición) permite ejecutar **más de una vez** una instrucción o un conjunto de ellas.

Esta estructura de control es especialmente útil para evitar repetir las instrucciones explícitamente una y otra vez cuando se debe ejecutar varias veces.

Control previo

Permite ejecutar de forma cíclica un bloque de instrucciones, siempre y cuando se satisfaga cierta **condición de control**

El ciclo se repetirá **mientras** dicha condición de control devuelva como resultado un valor **Verdadero**.



instrucción 1

mientras expresión evalúa **Verdadero** hacer

 instrucción 2

fin mientras

instrucción 3

RESOLUCIÓN DE PROBLEMAS

Problema

La división es una operación matemática que consiste en averiguar cuántas veces “cabe” un número, llamado divisor, en otro, llamado dividendo, y cuyo resultado recibe el nombre de cociente.

En el caso de la división entera, se puede tener un residuo, o resto, resultado de que el divisor no cabe exactamente un número entero de veces en el dividendo.

La división entera se puede calcular utilizando restas sucesivas, es decir, contando el número de veces que se resta el divisor al dividendo y tomando el resultado de la última resta como el residuo.

¿Cuál es el resultado de la división entera de un número x entre un número y ?

Por ejemplo, si dividimos 14 entre 3, el resultado sería 4 y el residuo 2, obtenido de la siguiente manera:

$\begin{array}{r} _{ } 14 \\ _{ } 3 \\ \hline 11 \end{array}$	$\begin{array}{r} _{ } 11 \\ _{ } 3 \\ \hline 8 \end{array}$	$\begin{array}{r} _{ } 8 \\ _{ } 3 \\ \hline 5 \end{array}$	$\begin{array}{r} _{ } 5 \\ _{ } 3 \\ \hline 2 \end{array}$
---	--	---	---

Estado inicial Son números enteros positivos, el dividendo siempre es mayor que el divisor.

Datos de entrada El *dividendo* y el *divisor*.

Datos de salida El *cociente* y el *residuo*.

Instrucciones a repetir

Lo primero que tengo que saber es, justamente, cuáles instrucciones deben **repetirse**.

Instrucciones a repetir

Lo primero que tengo que saber es, justamente, cuáles instrucciones deben **repetirse**.

$dividendo \leftarrow dividendo - divisor$

Instrucciones a repetir

Lo primero que tengo que saber es, justamente, cuáles instrucciones deben **repetirse**.

dividendo \leftarrow *dividendo* – *divisor*

dividendo \leftarrow *dividendo* – *divisor*

Instrucciones a repetir

Lo primero que tengo que saber es, justamente, cuáles instrucciones deben **repetirse**.

dividendo \leftarrow *dividendo* – *divisor*

dividendo \leftarrow *dividendo* – *divisor*

dividendo \leftarrow *dividendo* – *divisor*

Instrucciones a repetir

Lo primero que tengo que saber es, justamente, cuáles instrucciones deben **repetirse**.

dividendo \leftarrow *dividendo* – *divisor*

dividendo \leftarrow *dividendo* – *divisor*

dividendo \leftarrow *dividendo* – *divisor*

dividendo \leftarrow *dividendo* – *divisor*

$\text{dividendo} \leftarrow \text{dividendo} - \text{divisor}$

$\text{cociente} \leftarrow \text{cociente} + 1$

Condición de paro

Lo segundo que tengo que saber es **cuándo** debo dejar de repetir las instrucciones.

- La **condición** es una expresión lógica o relacional.
- Debe ser verdadera para que se repita la instrucción.
- Los valores involucrados en la condición deben cambiar en alguna iteración para que, en algún momento, la condición sea falsa y terminen las repeticiones.

Condición de paro

Lo segundo que tengo que saber es **cuándo** debo dejar de repetir las instrucciones.

- La **condición** es una expresión lógica o relacional.
- Debe ser verdadera para que se repita la instrucción.
- Los valores involucrados en la condición deben cambiar en alguna iteración para que, en algún momento, la condición sea falsa y terminen las repeticiones.

dividendo \geq divisor

Inicialización

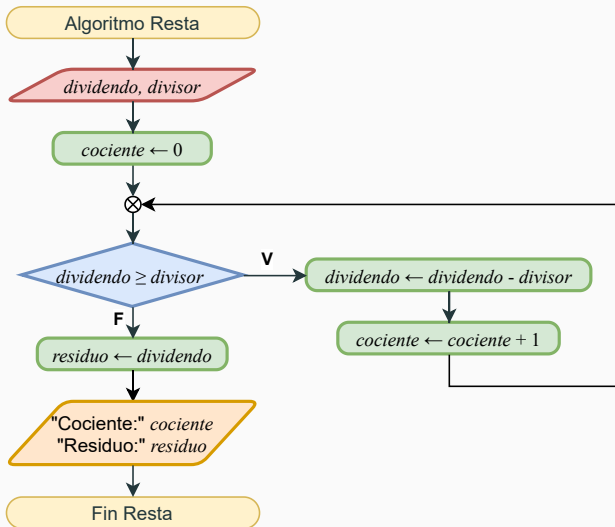
Por último, debo identificar los valores con los que deben **comenzar** los identificadores.

- Sea que adquieren un valor como entrada de datos.
- Sea que se les asigna un valor antes de comenzar las repeticiones.

NOTA: Una **no asignación** es diferente a asignar un valor de 0.

cociente $\leftarrow 0$

DIAGRAMA DE FLUJO



Algoritmo DivisionResta

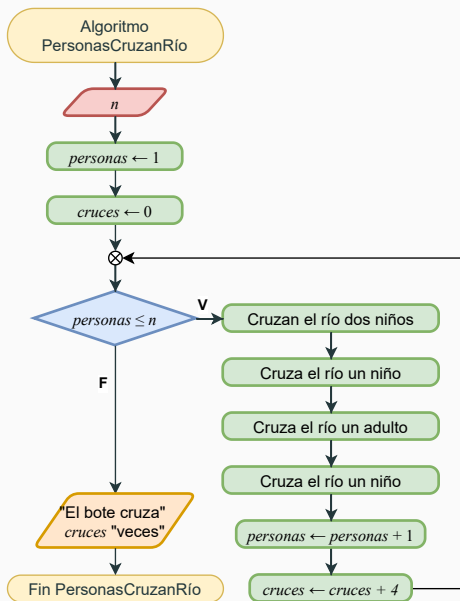
- 1: **leer** *dividendo*, *divisor*
- 2: *cociente* \leftarrow 0
- 3: **mientras** *dividendo* \geq *divisor* **hacer**
- 4: *dividendo* \leftarrow *dividendo* – *divisor*
- 5: *cociente* \leftarrow *cociente* + 1
- 6: **fin mientras**
- 7: *residuo* \leftarrow *dividendo*
- 8: **escribir** “El cociente es” *cociente*
- 9: **escribir** “El residuo es” *residuo*

Fin DivisionResta

Un grupo de 25 personas adultas debe cruzar un río ancho y profundo con ningún puente a la vista. Se dan cuenta que hay dos niños jugando con un bote de remos en la orilla. El bote es tan pequeño que sólo puede transportar, a lo mas, a dos niños o a un adulto. ¿Cómo pueden cruzar el río las personas y dejar a los niños en posesión del bote? ¿Cuántas veces pasa el bote de orilla a orilla en tu algoritmo?

Un grupo de n personas adultas debe cruzar un río ancho y profundo con ningún puente a la vista. Se dan cuenta que hay dos niños jugando con un bote de remos en la orilla. El bote es tan pequeño que sólo puede transportar a lo mas a dos niños o a un adulto. ¿Cómo pueden cruzar el río las personas y dejar a los niños en posesión del bote? ¿Cuántas veces pasa el bote de orilla a orilla en tu algoritmo?

DIAGRAMA DE FLUJO



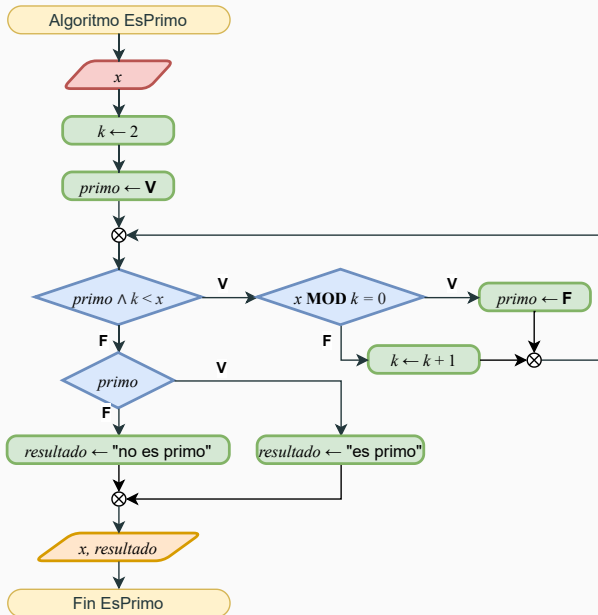
Se desea saber si un número x es primo.

Coloquialmente, se dice que un número primo es divisible entre 1 y entre sí mismo, pero, ¡todos los números cumplen esta condición!

Coloquialmente, se dice que un número primo es divisible entre 1 y entre sí mismo, pero, ¡todos los números cumplen esta condición!

Un número primo es un número natural mayor que 1, que únicamente tiene dos divisores distintos: 1 y él mismo. Es decir, un número x es primo si no existe un número entero k , con $1 < k < x$, tal que k divida a x .

DIAGRAMA DE FLUJO



Algoritmo EsPrimo

```
1: leer  $x$ 
2:  $k \leftarrow 2$ 
3:  $\text{primo} \leftarrow \text{verdadero}$ 
4: mientras  $\text{primo} \wedge k < x$  hacer
5:   si  $x \bmod 2 = 0$  entonces
6:      $\text{primo} \leftarrow \text{falso}$ 
7:   fin si
8:    $k \leftarrow k + 1$ 
9: fin mientras
10: si  $\text{primo}$  entonces
11:    $\text{resultado} \leftarrow \text{"es primo"}$ 
12: si no
13:    $\text{resultado} \leftarrow \text{"no es primo"}$ 
14: fin si
15: escribir  $x, \text{resultado}$ 
```

Fin EsPrimo