



**UNIVERSIDAD  
DE ANTIOQUIA**

# **LÓGICA Y REPRESENTACIÓN I**

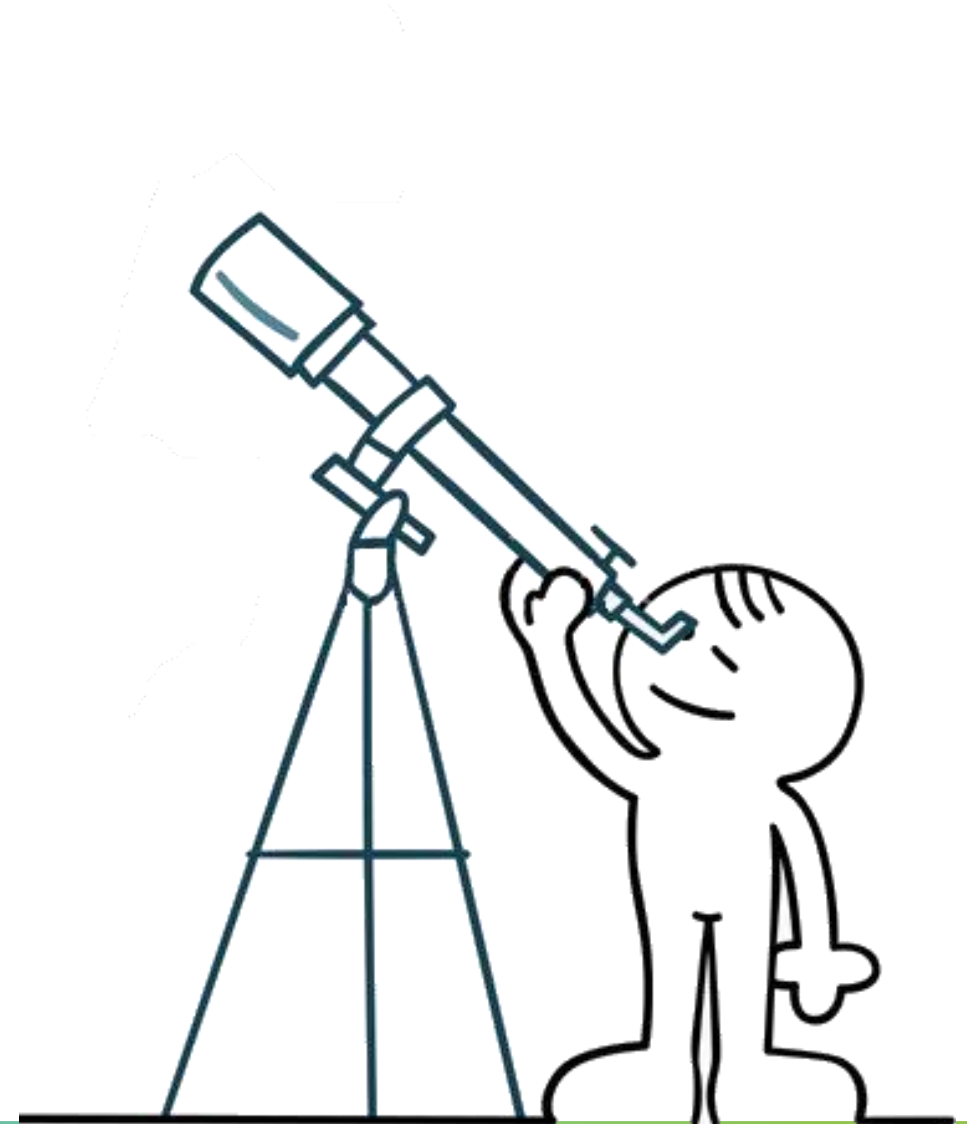
**PROGRAMA DE INGENIERÍA DE SISTEMAS**

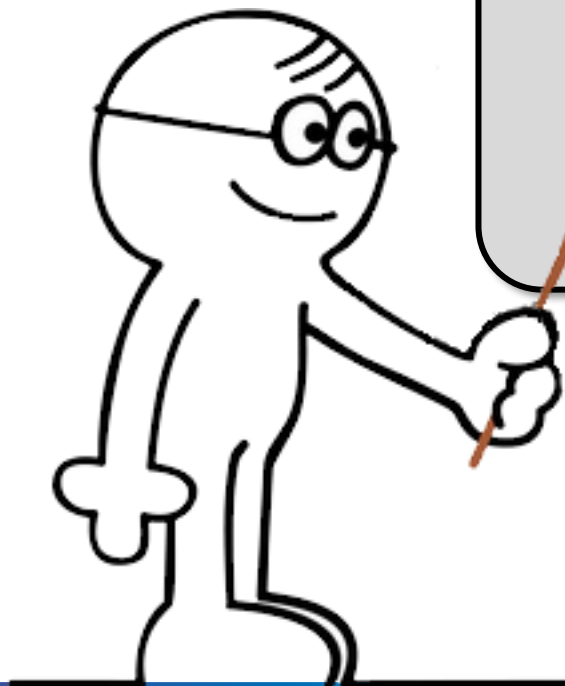
Material desarrollado por **Carlos Andrés Mera Banguero**

<https://github.com/carlosmera20/>

## 🦋 ESTRUCTURAS REPETITIVAS

- ✓ Introducción a las estructuras repetitivas
- ✓ Ciclo ***Para-Haga***
- ✓ Ciclo ***Mientras - Haga***





RECORDANDO: ESTRUCTURAS CONTROL

## ESTRUCTURAS DE CONTROL

Las estructuras de control permiten cambiar el flujo de ejecución de un programa, haciendo que ciertos bloques de código se ejecuten si y solo si se dan unas condiciones particulares.



### SELECTIVAS



Que permiten determinar si un bloque se ejecuta o no según si una **condición** es verdadera o falsa.

### REPETITIVAS



Que permiten que un bloque de código se ejecuta múltiples veces según si una **condición** es verdadera o falsa.

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🦋 **INTRODUCCIÓN:** considere el siguiente problema:



**EJERCICIO:** escriba un algoritmo que muestre en la pantalla las palabras  
“Hola Mundo” 10 veces.



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🦋 **DISEÑO DE LA SOLUCIÓN:** como el algoritmo es bastante sencillo lo escribimos en una sola clase :

## PSEUDOCÓDIGO

```
Clase HolaMundo
    publico vacio saludar()
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
        Escribir("Hola Mundo!")
    Fin Metodo
Fin Clase
```

## Python

```
class HolaMundo:

    def saludar(self):
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
        print("Hola Mundo!")
```

¿QUÉ TIENE DE PARTICULAR ESTA SOLUCIÓN?



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🦋 TIPOS DE ESTRUCTURAS DE CONTROL REPETITIVAS: existen dos tipos básicos de estructuras de repetición:

## CICLO PARA-HAGA

Ciclo **PARA** se usa cuando se conoce explícitamente el número de veces que se debe repetir un bloque de código. Este ciclo usa una variable contadora que es la que controla el ciclo.

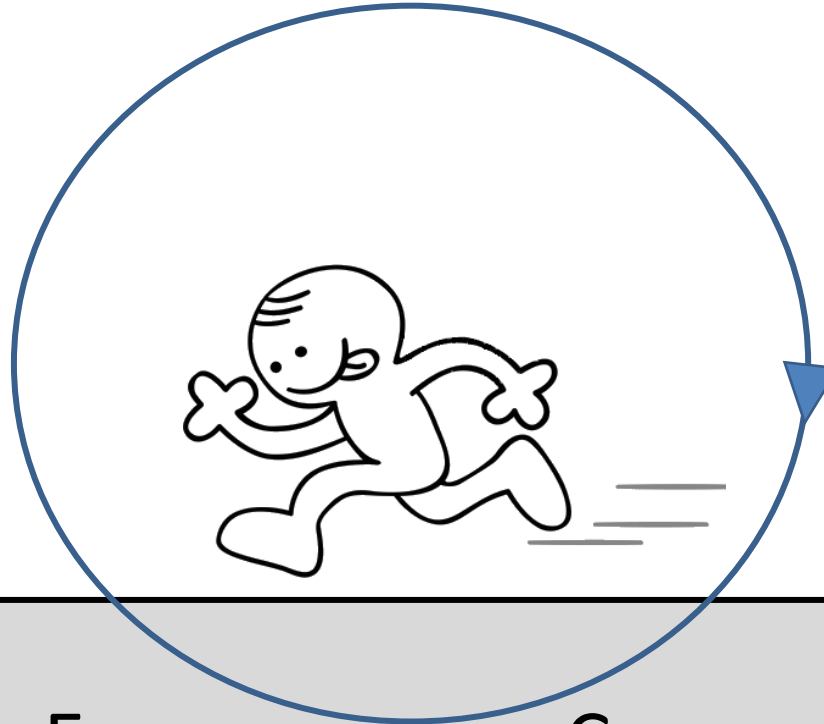


## CICLO MIENTRAS-HAGA

Ciclo **MIENTRAS**, se repite mientras que condición que se evalúa sea verdadera. Este ciclo usa o bien una variable contadora o centinela para controlar el ciclo.







ESTRUCTURAS DE CONTROL:  
ESTRUCTURA REPETITIVA PARA-HAGA

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

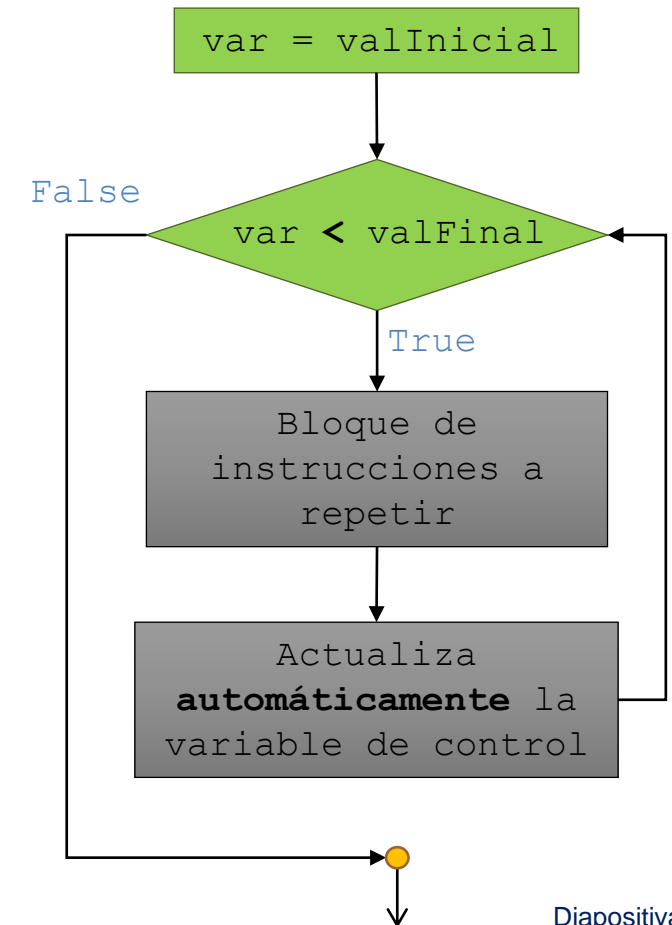
🚀 **CICLO PARA:** se usa cuando se conoce explícitamente el número de veces que se debe repetir un bloque de código. Este ciclo usa una **variable de control** (o variable del ciclo) que se utiliza para **contar** el número de veces que se repite el ciclo.

## Algoritmo

```
Para var = valInicial hasta valFinal incremento valInc Haga  
    <Instrucción_1>  
    <instrucción_2>  
    ...  
Fin_Para
```

## Python

```
for var in range(valInicial, valFinal, valInc):  
    <Instrucción_1>  
    <instrucción_2>  
    ...
```

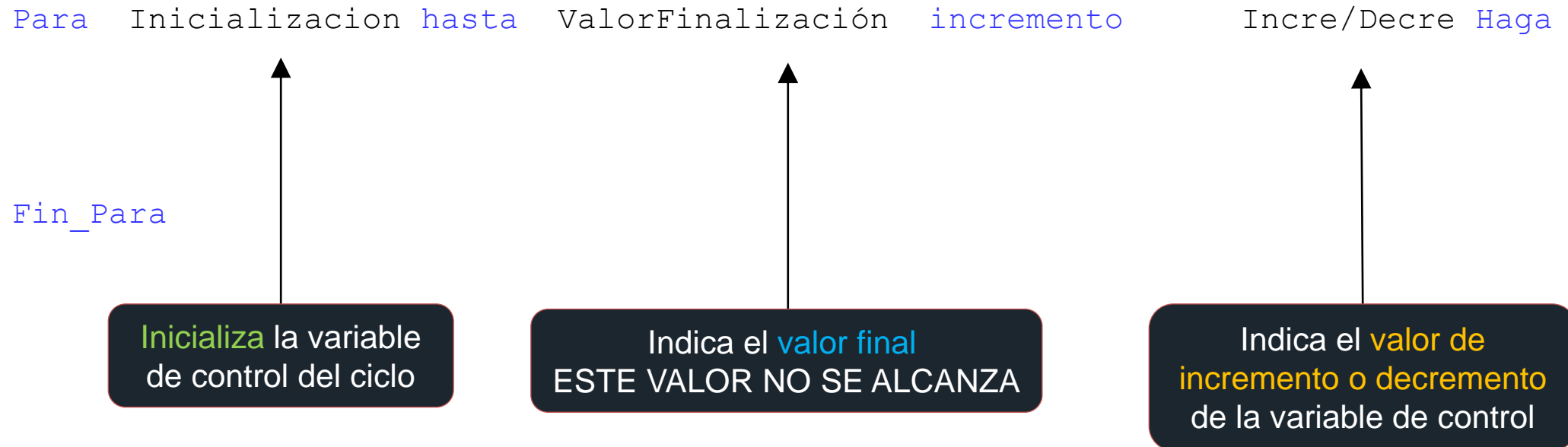


# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🦋 **CICLO PARA:** cada parte del ciclo tiene un significado:



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Escriba un algoritmo que muestre 10 veces en pantalla el mensaje “Hola Mundo”.



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🚀 **EJERCICIO:** escriba un algoritmo que muestre 10 veces en pantalla el mensaje “Hola Mundo”:

## PSEUDOCÓDIGO

**Clase** HolaMundo

publico vacio saludar()

Entero i

Para i=0 hasta 10 incremento 1 haga

    Escribir(“Hola Mundo!”)

Fin\_Para

Fin\_Metodo

**Fin\_Clase**

## Python

**class** HolaMundo:

def saludar(self):

    for i in range(0,10,1):

        print(“Hola Mundo!”)

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Haga un algoritmo que pida el nombre de 5 personas y las salute.



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🚀 **EJERCICIO:** haga un algoritmo que pida el nombre de 5 personas y las salude:

## PSEUDOCÓDIGO

**Clase** Saludo

**publico vacio** saludar\_personas()

**Entero** i

**Cadena** nombre

**Para** i=1 **hasta** 6 **incremento** 1 **haga**

**Escribir**("Escribe el nombre de tu amigo # ", i)

**Leer**(nombre)

**Escribir**("Hola ", nombre, "!")

**Fin\_Para**

**Fin\_Metodo**

**Fin\_Clase**

## PYTHON

**class** Saludo:

**def** saludar\_personas(**self**):

**for** i in range(1,6,1):

            nombre = **input**(f"Ingresa el nombre de la persona # {i}:")

**print**(f"Hola {nombre}!")

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Desarrolle un algoritmo que sume los números enteros del 1 al 100 y escriba en pantalla el resultado.





# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🏆 **EJERCICIO:** Desarrolle un algoritmo que sume los números enteros del 1 al 100 y escriba en pantalla el resultado:

## PSEUDOCÓDIGO

**Clase** Suma100Números

publico vacio sumar\_numeros()

Entero suma=0, i=0

Para i=1 hasta 101 incremento 1 haga

suma = suma + i

Fin\_Para

Escribir("La suma de los 100 números es: ", suma)

Fin\_Clase

## PYTHON

**class** Suma100Numeros:

**def** sumar\_numeros(**self**):

suma = 0

**for** i in range(1,101,1):

suma += i

print(f"La suma de los números es {suma}")

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Escriba un algoritmo que promedie los números pares entre dos números ingresados por el usuario.



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🦋 **EJERCICIO:** Escriba un algoritmo que promedie los números pares entre dos números ingresados por el usuario:

## PSEUDOCÓDIGO

**Clase** SumaNumerosPares

```
publico vacio sumar_pares()  
    Entero a, b, c, cont = 0, suma=0, i  
  
    Escribir("Ingrese dos números: ")  
    Leer(a, b)  
  
    Para i=a hasta b+1 incremento 1 haga  
        Si (i MOD 2 == 0) Entonces  
            suma = suma + i  
            cont = cont + 1  
        Fin_Si  
    Fin_Para  
    c = suma/cont  
    Escribir ("El promedio de los pares es: ", c)  
Fin_Metodo  
Fin_Clase
```

## PYTHON

```
class SumaNumerosPares :  
  
    def sumar_pares(self):  
        suma = 0  
        cont = 0  
        a = int(input("Ingrese el primer número: "))  
        b = int(input("Ingrese el segundo número: "))  
        for i in range(a,b+1,1):  
            if (i%2 == 0):  
                suma += i  
                cont += 1  
        c = suma/cont  
        print(f"El promedio de los pares entre {a} y {b} es {c}")
```

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Escriba un algoritmo que muestre los números del 20 al 0, de dos en dos, en reversa.



🦋 **EJERCICIO:** Escriba un algoritmo que muestre los números del 20 al 0, de dos en dos, en reversa:

## PSEUDOCÓDIGO

```
Clase NumerosEnReversa
    publico vacio mostrar_numeros()
        Entero i
        Escribir("Números en reversa del 20 al 0: ")

        Para i=20 hasta -1 incremento -2 haga
            Escribir(i)
        Fin_Para
    Fin_Metodo
Fin_Clase
```

## PYTHON

```
class NumerosEnReversa :

    def mostrar_numeros(self):

        for i in range(20,-1,-2):
            print(f"Numero {i} ")
```

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Haga un algoritmo que pida al usuario el número de estudiantes del curso de Lógica y Programación. El algoritmo debe preguntar el nombre de cada estudiante y luego mostrar el listado de los mismos.





**EJERCICIO:** Haga un algoritmo que pida la información de los 287 empleados de una empresa. El algoritmo debe almacenar el nombre, la edad, el peso, el salario y la antigüedad de cada empleado. El algoritmo debe permitir mostrar cuanto se pagará en total por prestaciones sociales, la edad promedio entre todos los empleados y generar un listado con los empleados que pesen más de 100 kg y que sean menores a 25 años o mayores a 40 años.

Tenga en cuenta que las prestaciones se utiliza la fórmula:

$$p = (a * s) / 12$$

Donde, p: prestaciones, a: antigüedad, s: salario



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🦋 **ANÁLISIS DEL PROBLEMA:** Identifique el cliente, usuario, los requisitos funcionales, las entradas, salidas y el proceso

## CLIENTE Y USUARIO

- 
- 



## ENTIDADES DEL MUNDO

- 
- 



## REQUERIMIENTOS FUNCIONALES

- 
- 



## ENTRADAS, SALIDAS Y EL PROCESO

- 
- 





# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

<b>CLIENTE</b>	La empresa
<b>USUARIO</b>	Personal de talento humano
<b>REQUERIMIENTOS FUNCIONALES</b>	<p>El sistema debe permitir:</p> <ul style="list-style-type: none"><li>▪ R1. Almacenar la información de los empleados de la empresa</li><li>▪ R2. Mostrar el valor total que se pagará por las prestaciones sociales de los empleados</li><li>▪ R3. Mostrar la edad promedio de los empleados</li><li>▪ R4. Generar un listado de empleados con ciertas condiciones</li></ul>
<b>ENTIDADES DEL MUNDO</b>	<ul style="list-style-type: none"><li>▪ Empleado: que representa un empleado de la empresa y del que se debe almacenar el nombre, la edad, el peso, el salario y la antigüedad</li><li>▪ Empresa: que almacena la información de los empleados</li></ul>
<b>ENTRADAS</b>	<ul style="list-style-type: none"><li>▪ El nombre del empleado, que es una cadena de texto</li><li>▪ La edad y la antigüedad del empleado, que son enteros positivos</li><li>▪ El peso y el salario del empleado, que son reales positivos</li></ul>
<b>SALIDAS</b>	<ul style="list-style-type: none"><li>▪ El valor total a pagar de las prestaciones sociales</li><li>▪ La edad promedio de los empleados</li><li>▪ Un listado de los empleados con nombre, edad y peso</li></ul>
<b>PROCESO</b>	<ul style="list-style-type: none"><li>▪ Las prestaciones de cada empleado se calculan como <math>p=(a*s)/12</math>, donde <math>a</math> es la antigüedad, y <math>s</math> el salario del empleado.</li><li>▪ La edad promedio se calcula como <math>\text{promedio} = (e_1 + e_2 + \dots + e_{287})/287</math>, donde <math>e_x</math> es la edad de cada empleado</li></ul>

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🦋 **ANÁLISIS DEL PROBLEMA:** Identifique el cliente, usuario, los requisitos funcionales, las entradas, salidas y el proceso

<b>IDENTIFICADOR</b>	R1
<b>NOMBRE</b>	Almacenar la información de los empleados
<b>RESUMEN</b>	El sistema debe permitir almacenar la información de los 287 empleados de la empresa. Dicha información corresponde al nombre, edad, peso, salario y antigüedad del empleado.
<b>ENTRADAS</b>	Nombre, edad, peso, salario y la antigüedad
<b>SALIDAS O RESULTADOS</b>	<ul style="list-style-type: none"><li>- Se almacena la información de los empleados</li><li>- Al terminar se muestra un mensaje en la pantalla indicando que la información ha sido almacenada</li><li>- Si alguno de los datos numéricos es menor a cero se muestra un error, igualmente si alguno de los datos llega vacío</li></ul>

🦋 **ANÁLISIS DEL PROBLEMA:** Identifique el cliente, usuario, los requisitos funcionales, las entradas, salidas y el proceso

IDENTIFICADOR	R2
NOMBRE	Mostrar el valor total que se pagara por prestaciones sociales de los empleados
RESUMEN	<p>El sistema debe calcular y mostrar el total de prestaciones sociales que se deben pagar de los empleados, teniendo en cuenta la siguiente formula:</p> $total\_prestaciones = \sum_{i=1}^{287} (a_i * s_i) / 12$ <p>donde <math>a_i</math> y <math>s_i</math> es la antigüedad y el salario del empleado i.</p>
ENTRADAS	No hay entradas
SALIDAS O RESULTADOS	Se muestra en la pantalla el total de las prestaciones sociales de los empleados.

🦋 **ANÁLISIS DEL PROBLEMA:** Identifique el cliente, usuario, los requisitos funcionales, las entradas, salidas y el proceso

<b>IDENTIFICADOR</b>	R4
<b>NOMBRE</b>	Generar un listado de empleados con ciertas condiciones
<b>RESUMEN</b>	El sistema debe dar la opción a los empleados de recursos humanos la posibilidad de generar un listado de empleados que pesen más de 100 Kg y que sean menores de 25 o mayores de 40 años. Dicho listado se mostrará en la pantalla y contendrá el nombre de cada empleado, la edad y el peso.
<b>ENTRADAS</b>	No hay entradas
<b>SALIDAS O RESULTADOS</b>	Se muestra en la pantalla la lista de empleados que cumplen la condición del listado. Si no hay empleados que cumplan esa condición se indicará al usuario que no hay empleados que cumplan esa condición.

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Desarrolle un algoritmo que calcule el factorial de un número ingresado por teclado.

$$5! = 5*4*3*2*1$$



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Desarrolle un algoritmo que calcule el factorial de un número ingresado por teclado.

## PSEUDOCÓDIGO

**Clase** Factorial

**publico** **Real** calcular\_factorial(**Entero** n)

**Entero** i, fact=1

**Para** i=1 **hasta** n+1 **incremento** 1 **haga**

            fact = fact \* i

**Fin\_Para**

**return** fact

**Fin\_Metodo**

**Fin\_Clase**

## PYTHON

**class** Factorial:

**def** calcular\_factorial(**self**, n):

        fact = 1

**for** i in range(1,n+1,1):

            fact = fact \* i

**return** fact

# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**EJERCICIO:** Haga un algoritmo que indique en pantalla si un número ingresado es primo o no.



# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA

🏆 **EJERCICIO:** Haga un algoritmo que indique en pantalla si un número ingresado es o no primo:

## PSEUDOCÓDIGO

```
Clase NumeroPrimo
    publico vacio es_primo()
        Entero i, n
        Logico flag = true

        Escribir("Ingrese un número: ")
        Leer(n)
        Para i=2 hasta n incremento 1 haga
            Si (n MOD i == 0) Entonces
                flag = false
            Fin_si
        Fin_Para
        Si (flag == true) Entonces
            Escribir("El número es primo!")
        Sino
            Escribir("El número NO es primo!")
        Fin_si
    Fin_Metodo
Fin_Clase
```

## PYTHON

```
class NumeroPrimo:

    def es_primo(self):
        bandera = True
        n = int(input("Ingrese un número: "))
        for i in range(2,n//2,1):
            if (n % i ==0):
                bandera = False

        if (bandera == True):
            print(f"El número {n} es primo")
        else:
            print(f"El número {n} NO es primo")
```



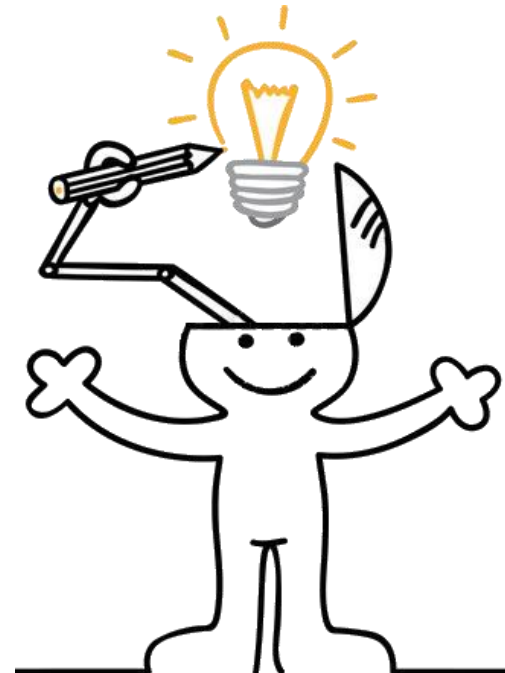
# ESTRUCTURAS DE CONTROL REPETITIVAS



UNIVERSIDAD  
DE ANTIOQUIA



**TAREA:** Escriba un algoritmo que pida al usuario el número de estudiantes del curso de Lógica y Representación I. Por cada estudiante, el algoritmo debe preguntar el nombre y las 5 notas de los exámenes de cada estudiante. El algoritmo debe mostrar el listado de los estudiantes con la nota final, calculada como el promedio de las 5 notas ingresadas.





**UNIVERSIDAD  
DE ANTIOQUIA**

# **LÓGICA Y REPRESENTACIÓN I**

**PROGRAMA DE INGENIERÍA DE SISTEMAS**

Material desarrollado por **Carlos Andrés Mera Banguero**

<https://github.com/carlosmera20/>