



# PROGRAMA 111 MIL

## Clase N°3

### TÉCNICAS DE PROGRAMACIÓN



## Desarrollo de Programas – Estructuras de Programación

### 1. ESTRUCTURAS DE CONTROL

- SECUENCIALES
  - SELECTIVAS O DE DECISIÓN
  - REPETITIVAS
- LAS ESTRUCTURAS DE CONTROL DETERMINAN EL ORDEN EN QUE DEBEN EJECUTARSE LAS INSTRUCCIONES DE UN ALGORITMO: SI SERÁN RECORRIDAS UNA LUEGO DE LA OTRA, SI HABRÁ QUE TOMAR DECISIONES SOBRE SI EJECUTAR O NO ALGUNA ACCIÓN O SI HABRÁ REPETICIONES.



## Desarrollo de Programas – Estructuras de Programación

### 2. ESTRUCTURA SECUENCIAL

- ES LA ESTRUCTURA EN DONDE **UNA ACCIÓN (INSTRUCCIÓN) SIGUE A OTRA DE MANERA SECUENCIAL**. LAS TAREAS SE DAN DE TAL FORMA QUE LA SALIDA DE UNA ES LA ENTRADA DE LA QUE SIGUE Y ASÍ EN LO SUCESIVO HASTA CUMPLIR CON TODO EL PROCESO. ESTA ESTRUCTURA DE CONTROL ES LA MÁS SIMPLE, PERMITE QUE LAS INSTRUCCIONES QUE LA CONSTITUYEN **SE EJECUTEN UNA TRAS OTRA** EN EL ORDEN EN QUE SE LISTAN. POR EJEMPLO, CONSIDÉRESE EL SIGUIENTE FRAGMENTO DE UN ALGORITMO:





## Desarrollo de Programas – Estructuras de Programación

### 2. ESTRUCTURA ALTERNATIVA

- ESTAS ESTRUCTURAS DE CONTROL SON DE GRAN UTILIDAD PARA CUANDO EL **ALGORITMO A DESARROLLAR REQUIERA UNA DESCRIPCIÓN MÁS COMPLICADA** QUE UNA LISTA SENCILLA DE INSTRUCCIONES. ESTE ES EL CASO CUANDO EXISTE UN NÚMERO DE POSIBLES ALTERNATIVAS QUE RESULTAN DE LA EVALUACIÓN DE UNA DETERMINADA CONDICIÓN. ESTE TIPO DE ESTRUCTURAS SON UTILIZADAS PARA TOMAR **DECISIONES LÓGICAS**, ES POR ESTO QUE TAMBIÉN SE **DENOMINAN ESTRUCTURAS DE DECISIÓN O SELECTIVAS**.
- EN ESTAS ESTRUCTURAS, SE REALIZA UNA **EVALUACIÓN DE UNA CONDICIÓN** Y DE ACUERDO AL RESULTADO, EL ALGORITMO REALIZA UNA DETERMINADA ACCIÓN. LAS CONDICIONES SON ESPECIFICADAS UTILIZANDO **EXPRESIONES LÓGICAS**.



## Desarrollo de Programas – Estructuras de Programación

### 2. ESTRUCTURA ALTERNATIVA

- LAS ESTRUCTURAS SELECTIVAS/ALTERNATIVAS PUEDEN SER:

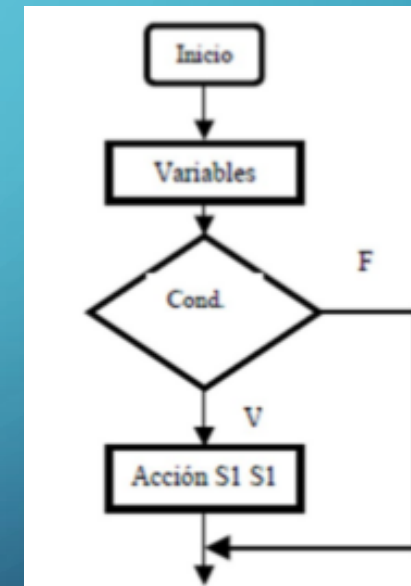
- SIMPLES
- DOBLES
- MÚLTIPLES



## Desarrollo de Programas – Estructuras de Programación

### 2. ALTERNATIVA SIMPLE (IF-THEN)

- LA ESTRUCTURA ALTERNATIVA SIMPLE SI-ENTONCES (EN INGLÉS IF-THEN) LLEVA A CABO UNA ACCIÓN AL CUMPLIRSE UNA DETERMINADA CONDICIÓN. LA SELECCIÓN SI-ENTONCES EVALÚA LA CONDICIÓN Y:
- SI LA CONDICIÓN ES VERDADERA, EJECUTA LA ACCIÓN S1
- SI LA CONDICIÓN ES FALSA, NO EJECUTA NADA.





## Desarrollo de Programas – Estructuras de Programación

### 2. ALTERNATIVA SIMPLE (IF-THEN)

En español:

```
Si <condición>  
Entonces <acción S1>  
Fin_si
```

En Inglés:

```
If <condición>  
Then <acción S1>  
End_if
```

Ejemplo:

INICIO

ENTERO edad = 18

SI (edad > 18)

ENTONCES:

    puede manejar un auto

FIN\_SI

FIN

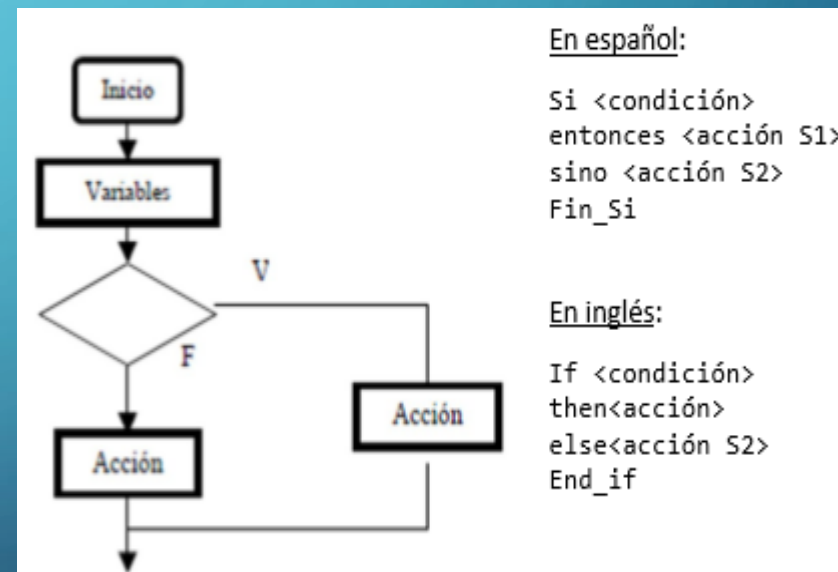




## Desarrollo de Programas – Estructuras de Programación

### 3. ALTERNATIVA DOBLE (IF-THEN-ELSE)

- EXISTEN LIMITACIONES EN LA ESTRUCTURA ANTERIOR, Y SE NECESITARÁ NORMALMENTE UNA ESTRUCTURA QUE PERMITA ELEGIR DOS OPCIONES O ALTERNATIVAS POSIBLES, DE ACUERDO AL CUMPLIMIENTO O NO DE UNA DETERMINADA CONDICIÓN:
- SI LA CONDICIÓN ES VERDADERA, SE EJECUTA LA ACCIÓN S1
- SI LA CONDICIÓN ES FALSA, SE EJECUTA LA ACCIÓN S2







## Desarrollo de Programas – Estructuras de Programación

### 3. ALTERNATIVA DOBLE (IF-THEN-ELSE)

Ejemplo:

INICIO

    BOOLEANO afueraLlueve = verdadero

    SI (afueraLlueve es verdadero)

        ENTONCES:

            me quedo viendo películas

    SINO:

        salgo al parque a tomar mates

    FIN\_SI

FIN



## Desarrollo de Programas – Estructuras de Programación

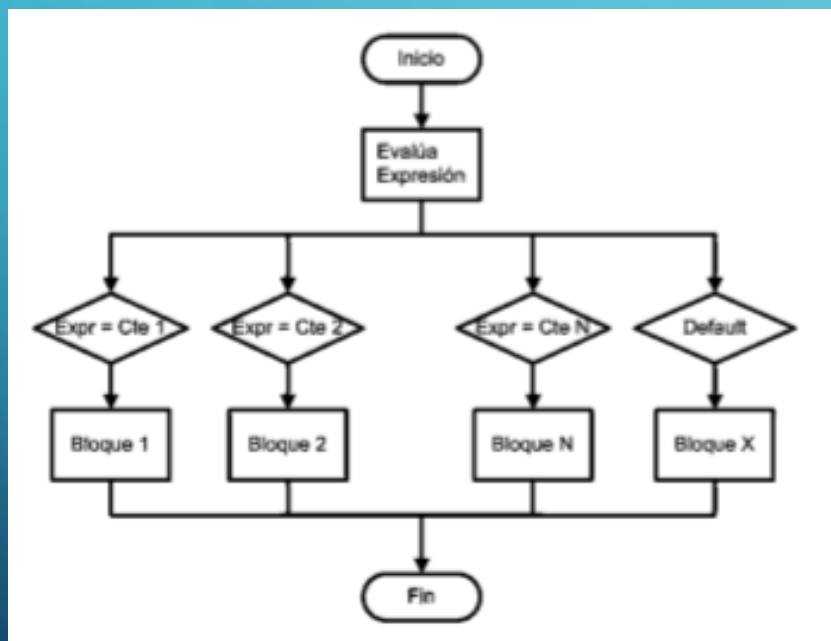
### 4. ALTERNATIVA DECISIÓN MÚLTIPLE (CASE)

- SE UTILIZA CUANDO EXISTEN **MÁS DE DOS ALTERNATIVAS PARA ELEGIR**. ESTO PODRÍA SOLUCIONARSE POR MEDIO DE ESTRUCTURAS ALTERNATIVAS SIMPLES O DOBLES, **ANIDADAS O EN CASCADA**. SIN EMBARGO, SE PUEDEN PLANTEAR SERIOS **PROBLEMAS DE ESCRITURA DEL ALGORITMO**, DE COMPRENSIÓN Y DE LEGIBILIDAD, SI EL NÚMERO DE ALTERNATIVAS ES GRANDE. EN ESTA ESTRUCTURA, SE EVALÚA UNA CONDICIÓN O EXPRESIÓN QUE PUEDE TOMAR **N VALORES**. SEGÚN EL VALOR QUE LA EXPRESIÓN TENGA EN CADA MOMENTO SE EJECUTAN LAS ACCIONES CORRESPONDIENTES AL VALOR.



## Desarrollo de Programas – Estructuras de Programación

### 4. ALTERNATIVA DECISIÓN MÚLTIPLE (CASE)



Ejemplo en pseudocódigo:

INICIO

ENTERO posicionDeLlegada = 3

SEGUN SEA posicionDeLlegada

1: entregar medalla de oro

2: entregar medalla de plata

3: entregar medalla de bronce

otro: entregar mención especial

FIN



## Desarrollo de Programas – Estructuras de Programación

### 4. ALTERNATIVA DECISIÓN MÚLTIPLE (CASE)

- ES IMPORTANTE MENCIONAR QUE LA ESTRUCTURA ANTERIOR PUEDE SER ESCRITA USANDO LOS CONDICIONALES VISTOS ANTERIORMENTE DE LA SIGUIENTE FORMA:
- ¿CUAL ES EL ERROR EN ESTE EJEMPLO?

```
INICIO
ENTERO posicionDeLlegada = 3
SI (posicionDeLlegada = 1)
    ENTONCES:
        entregar medalla de oro
    SINO:
        SI (posicionDeLlegada = 2)
            ENTONCES:
                entregar medalla de plata
            SINO:
                SI (posicionDeLlegada = 3)
                    ENTONCES:
                        entregar medalla de bronce
                    SINO:
                        entregar mención especial
                FIN_SI
            FIN_SI
        FIN_SI
    FIN
FIN
```



## Desarrollo de Programas – Estructuras de Programación

### 5. ESTRUCTURA REPETITIVA O ITERATIVA

- DURANTE EL **PROCESO DE CREACIÓN DE PROGRAMAS**, ES MUY COMÚN, ENCONTRARSE CON QUE UNA OPERACIÓN O CONJUNTO DE OPERACIONES DEBEN **REPETIRSE MUCHAS VECES**. PARA ELLO ES IMPORTANTE CONOCER LAS ESTRUCTURAS DE ALGORITMOS QUE PERMITEN REPETIR UNA O VARIAS ACCIONES, UN **NÚMERO DETERMINADO DE VECES**. LAS ESTRUCTURAS QUE REPITEN UNA SECUENCIA DE INSTRUCCIONES UN NÚMERO DETERMINADO DE VECES SE DENOMINAN **BUCLES**. Y CADA REPETICIÓN DEL BUCLE SE LLAMA **ITERACIÓN**. TODO BUCLE TIENE QUE LLEVAR ASOCIADA UNA **CONDICIÓN**, QUE ES LA QUE VA A DETERMINAR CUÁNDO SE REPITE EL BUCLE Y CUANDO DEJA DE REPETIRSE.



## Desarrollo de Programas – Estructuras de Programación

### 5. ESTRUCTURA REPETITIVA O ITERATIVA

- UN BUCLE SE DENOMINA TAMBIÉN **LAZO** O **LOOP**. HAY QUE PRESTAR ESPECIAL ATENCIÓN A LOS **BUCLES INFINITOS**, HECHO QUE OCURRE CUANDO LA CONDICIÓN DE FINALIZACIÓN DEL BUCLE NO SE LLEGA A CUMPLIR NUNCA. SE TRATA DE UN **FALLO MUY TÍPICO**, HABITUAL SOBRE TODO ENTRE PROGRAMADORES PRINCIPIANTES.
- HAY DISTINTOS TIPOS DE BUCLES:
- MIENTRAS, EN INGLÉS: **WHILE**
- HACER MIENTRAS, EN INGLÉS: **DO WHILE**.
- PARA, EN INGLÉS: **FOR**



## Desarrollo de Programas – Estructuras de Programación

### 6. ESTRUCTURA MIENTRAS (WHILE)

- ESTA ESTRUCTURA REPETITIVA **“MIENTRAS”**, ES EN LA QUE EL CUERPO DEL BUCLE SE REPITE SIEMPRE QUE SE CUMPLA UNA **DETERMINADA CONDICIÓN**.

Diagrama de Flujo



Figura 19: Diagrama de Flujo de la estructura “mientras”

Pseudocódigo:

```
Mientras <cond> hacer  
  <acciones>  
fin mientras
```

Ejemplo:

```
INICIO  
  BOOLEANO tanquelleno = falso  
  MIENTRAS (tanquelleno == falso)  
  HACER:  
    llenar tanque  
  FIN_MIENTRAS  
  // el tanque ya está lleno :)  
FIN
```





## Desarrollo de Programas – Estructuras de Programación

### 6. ESTRUCTURA HACER-MIENTRAS (DO WHILE)

- ESTA ESTRUCTURA ES MUY SIMILAR A LA ANTERIOR, SÓLO QUE A DIFERENCIA DEL WHILE EL CONTENIDO DEL BUCLE SE **EJECUTA SIEMPRE AL MENOS UNA VEZ**, YA QUE LA EVALUACIÓN DE LA CONDICIÓN SE ENCUENTRA AL FINAL. DE ESTA FORMA GARANTIZAMOS QUE LAS ACCIONES DENTRO DE ESTE BUCLE SEAN LLEVADAS A CABO, AUNQUE SEA UNA VEZ INDEPENDIENTEMENTE DEL VALOR DE LA CONDICIÓN.

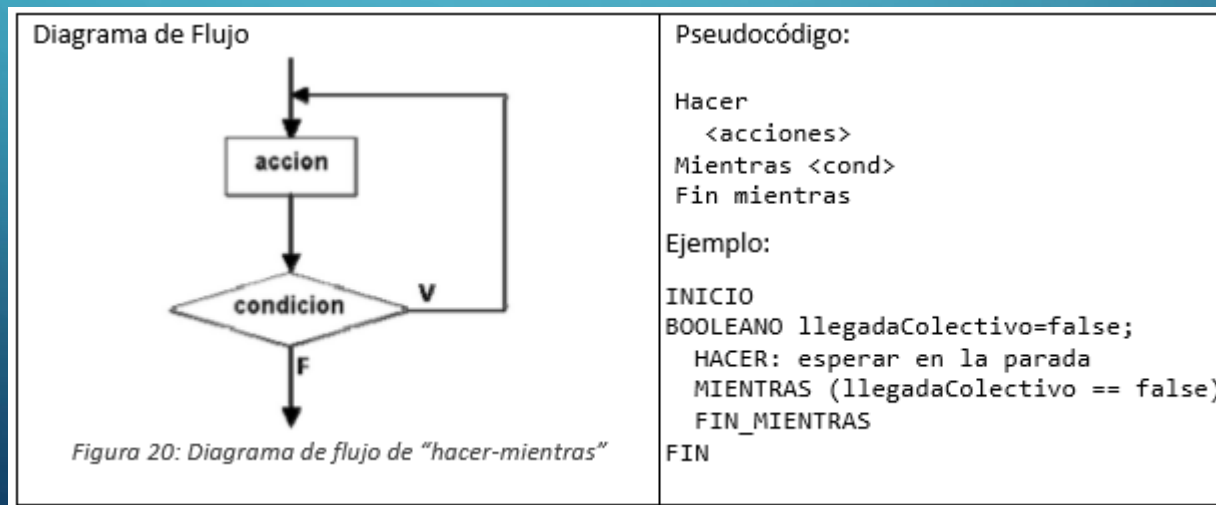


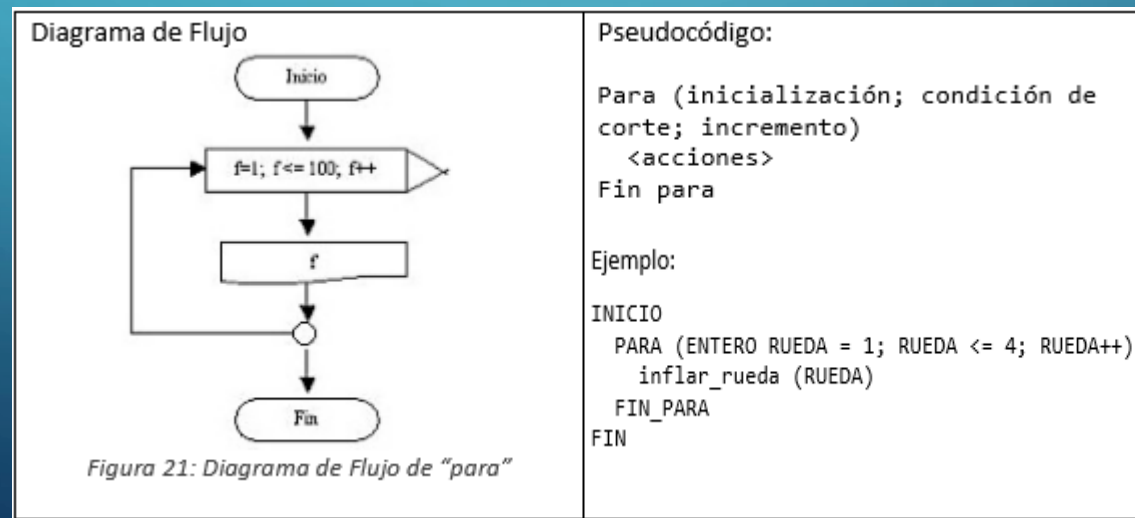
Figura 20: Diagrama de flujo de "hacer-mientras"



## Desarrollo de Programas – Estructuras de Programación

### 6. ESTRUCTURA PARA (FOR)

- LA ESTRUCTURA FOR ES UN POCO MÁS COMPLEJA QUE LAS ANTERIORES Y NOS PERMITE EJECUTAR UN CONJUNTO DE ACCIONES PARA CADA ELEMENTO DE UNA LISTA, O PARA CADA PASO DE UN CONJUNTO DE ELEMENTOS. SU IMPLEMENTACIÓN DEPENDE DEL LENGUAJE DE PROGRAMACIÓN, PERO EN TÉRMINOS GENERALES PODEMOS IDENTIFICAR TRES COMPONENTES: LA INICIALIZACIÓN, LA CONDICIÓN DE CORTE Y EL INCREMENTO.





## Practica – Estructuras de Programación

### 1. MANOS A LA OBRA

- PRUEBAS DE ESCRITORIO:

Algoritmo a Probar:	Prueba de Escritorio	
	Altura Persona	Salida
	1.50	"Puede ingresar a la montaña rusa"
	1.20	"No puede ingresar a la montaña rusa"
	1.30	"Puede ingresar a la montaña rusa"
INICIO validarAltura FLOAT alturaPermitida = 1.30 SI (alturaPersona >= alturaPermitida) ENTONCES: "Puede ingresar a la montaña rusa" SINO: "No puede ingresar a la montaña rusa" FIN_SI FIN	1.00	"No puede ingresar a la montaña rusa"



## Ejercicios Estructuras de Control – Estructuras de Programación

### 2. EJERCICIOS

#### 1. SECUENCIAL

ESCRIBIR UN ALGORITMO QUE PERMITA REALIZAR UNA SUMA DE DOS NÚMEROS ENTEROS. EL USUARIO DEBERÁ INGRESAR PRIMERO UN NÚMERO, LUEGO EL SIGUIENTE NÚMERO, Y EL SISTEMA ARROJARÁ EL RESULTADO CORRESPONDIENTE.

#### 2. ALTERNATIVA SIMPLE

ESCRIBIR UN ALGORITMO QUE PERMITA LOGUEARSE (REGISTRASE) A UN SISTEMA, INGRESANDO UN NOMBRE DE USUARIO Y LA CONTRASEÑA ADECUADA. CONSIDERAR QUE TANTO EL USUARIO COMO LA CONTRASEÑA ESTÁN FORMADOS SÓLO POR LETRAS. EL SISTEMA DEBERÁ VALIDAR QUE EL USUARIO Y LA CONTRASEÑA SEAN CORRECTAS, COMPARÁNDOLAS CON LO QUE ES SISTEMA TIENE REGISTRADO PARA ESE USUARIO.

**\*\*ACLARACIÓN, EN LOS SISTEMAS REALES, EL INICIO DE SESIÓN ES MUCHO MÁS COMPLEJO QUE LO QUE SE MUESTRA A CONTINUACIÓN. SE HA SIMPLIFICADO EL PROCESO, ABSTRAYENDO LA VALIDACIÓN A UNA FUNCIÓN DENOMINADA ESVALIDO() QUE RESUELVE LA VERIFICACIÓN DEL USUARIO Y SU CONTRASEÑA.**



## Ejercicios Estructuras de Control – Estructuras de Programación

### 2. EJERCICIOS

#### 3. ALTERNATIVA DOBLE

ESCRIBIR EL ALGORITMO QUE, A PARTIR DE LA CANTIDAD DE BANCOS DE UN AULA Y LA CANTIDAD DE ALUMNOS INSCRIPTOS PARA UN CURSO, PERMITA DETERMINAR SI ALCANZAN LOS BANCOS EXISTENTES. DE NO SER ASÍ, INFORMAR ADEMÁS CUANTOS BANCOS SERÍA NECESARIO AGREGAR.

#### 4. ALTERNATIVA DOBLE

DISEÑAR UN ALGORITMO QUE PERMITA APLICAR UN DESCUENTO DEL 10% AL MONTO TOTAL DE UNA COMPRA SI LA FORMA DE PAGO EMPLEADA ES DE CONTADO.



## Ejercicios Estructuras de Control – Estructuras de Programación

### 2. EJERCICIOS

#### 5. ALTERNATIVA MÚLTIPLE

DISEÑAR UN ALGORITMO QUE DEVUELVA EL NOMBRE DEL MES, A PARTIR DEL NÚMERO DE MES, INGRESADO POR TECLADO, POR EL USUARIO.

#### 6. REPETITIVA WHILE

DISEÑAR UN ALGORITMO QUE MUESTRE POR PANTALLA LA TABLA DE MULTIPLICACIÓN DEL NÚMERO QUE INGRESE EL USUARIO. PARA DEFINIR HASTA QUE NÚMERO DESEA QUE MUESTRE LA TABLA DE MULTIPLICACIÓN EL USUARIO TAMBIÉN DEBERÁ INGRESAR ESTE VALOR.





## Ejercicios Estructuras de Control – Estructuras de Programación

### 2. EJERCICIOS

#### 7. REPETITIVA DO WHILE

DISEÑAR UN ALGORITMO QUE MUESTRE POR PANTALLA LA TABLA DE MULTIPLICACIÓN DEL NÚMERO QUE INGRESE EL USUARIO. PARA DEFINIR HASTA QUE NÚMERO DESEA QUE MUESTRE LA TABLA DE MULTIPLICACIÓN EL USUARIO TAMBIÉN DEBERÁ INGRESAR ESTE VALOR.

#### 8. REPETITIVA FOR

DISEÑAR UN ALGORITMO QUE REALICE EL PROMEDIO DE 4 NÚMEROS. LOS NÚMEROS PODRÁN SER DECIMALES Y SERÁN INGRESADOS POR PANTALLA POR EL USUARIO.