

# **CATEDRA INFORMATICA CBI**

## **FACET – UNT**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION**

**CARRERAS:**

**ING INFORMATICA**

**ING AZUCARERA**

**ING QUIMICA**

**ING INDUSTRIAL**

**ING MECANICA**

**AGRIMENSURA**

**ING GEODESICA Y GEOFISICA**

**2023**



**UNIVERSIDAD  
NACIONAL  
DE TUCUMÁN**



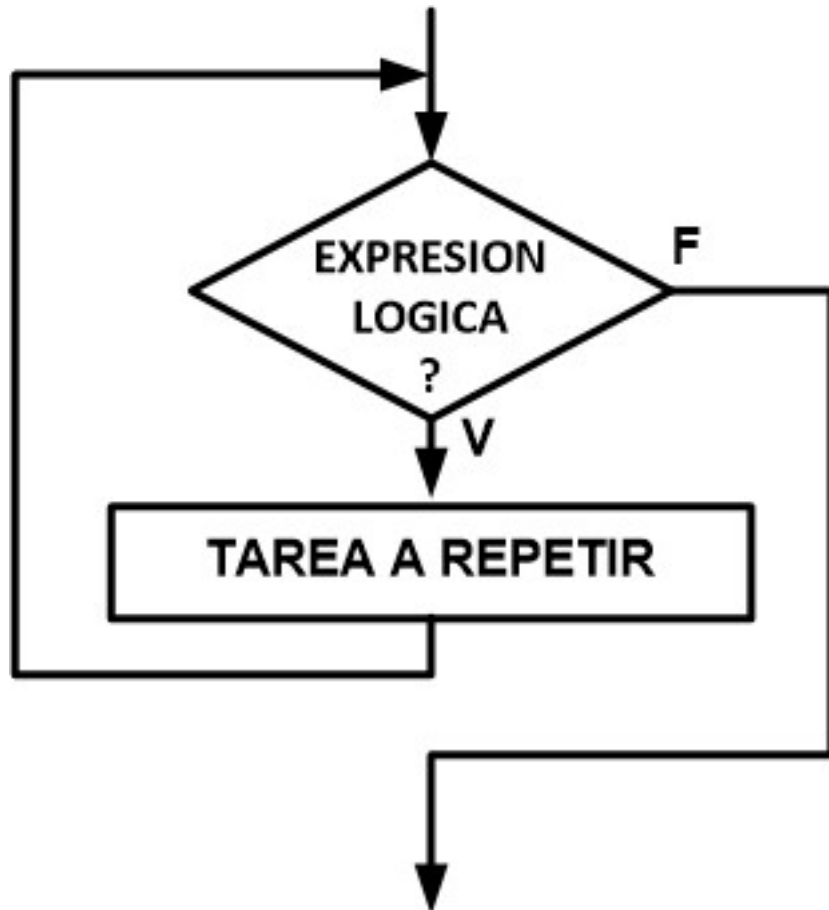
# **CODIFICACION DE UN ALGORITMO EN DIAGRAMA DE FLUJO A LENGUAJE ESTRUCTURADO “C”**

## **ESTRUCTURAS REPETITIVAS**

# CATEDRA INFORMATICA CBI FACET - UNT

## CÓMO PASAR EL ALGORITMO A CODIGO

### Mientras While



### ESTRUCTURA DE REPETICION WHILE

/\*Dante Fernandez,

13/5/21,

Ejemplo de WHILE (Mientras)\*/

#include <stdio.h>

int main() {

int x,i=1; // Declaración de variables

printf("Ingrese valor de x= ");

scanf("%d",&x);

while (i<=x)

{

printf("%d \n",i);

i++; //Incremento de i

}

return (0);

}

# CATEDRA INFORMATICA CBI FACET - UNT

## DETALLES SOBRE ESTE PROGRAMA (1)

1- Declaración de variables. Vemos aquí que podemos declarar varias variables del mismo tipo al mismo tiempo. Además le podemos asignar el valor inicial en la misma sentencia. La asignación se hace con el signo = , por ejemplo `i=1`.

Hay distintos tipos de variables:

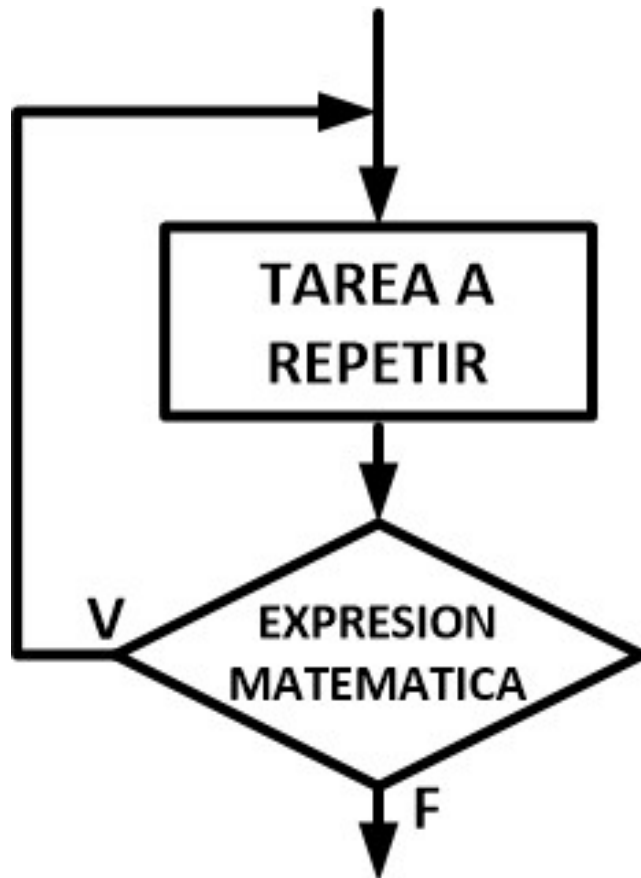
Identificador	Tipo	Tamaño en bits	Intervalo
<code>char</code>	Carácter	8 (normalmente)	0 a 255
<code>int</code>	Entero con signo	16 (depende de la implementación)	-32 768 a 32 767
<code>float</code>	Flotante simple	32 (normalmente)	3.4E-38 a 3.4E+38 (aprox. 6 dígitos de precisión)
<code>double</code>	Flotante doble	64 (normalmente)	1.7E-308 a 1.7E+308 (aprox. 12 dígitos de precisión)
<code>void</code>	Nada Genérico	0	sin valor

2- Incremento de `i`. En la codificación en C hay algunos modismos interesantes: `i++` equivale a `i=i+1`  
Otro ejemplo: `i--` equivale a `i=i-1`

# CATEDRA INFORMATICA CBI FACET - UNT

## CÓMO PASAR EL ALGORITMO A CODIGO

### Hacer Mientras Do While



### ESTRUCTURA DE REPETICION DO-WHILE

```
/*Dante Fernandez,  
13/5/21,  
Ejemplo de DO WHILE (Hacer Mientras)*/  
#include <stdio.h>  
int main() {  
    int x,i=1;  
    printf("Ingrese valor de x= ");  
    scanf("%d",&x); //Desde teclado  
    do  
    {  
        printf("%d\n",i); //En pantalla  
        i++;  
    }  
    while (i<=x);  
    return (0);  
}
```

# CATEDRA INFORMATICA CBI FACET - UNT

## DETALLES SOBRE ESTE PROGRAMA (2)

1- Desde teclado. Cuando se ingresa un valor desde teclado, se usa el comando *scanf* y se declara el tipo de dato entre comillas como si fuera un texto. Después de una coma (,) se pone la variable con un signo & adelante que indica al compilador que debe memorizar la ubicación en la memoria de esa variable.  
*scanf("%d",&x);* %d indica que el tipo de dato es entero/decimal

2- En pantalla. Cuando el comando *printf* debe mostrar un texto que además lleva valores de variables, primero declara la posición de la variable en el texto, dentro de las comillas, con el símbolo %d. Después de las comillas y separadas por comas (,) se declaran las variables en el orden en que van en el texto.

Ejemplo: *printf("Las %d casas contienen %d ventanas",i,n);*

Si *i=3* y *n=50*, entonces el mensaje que sale en la pantalla es:

**Las 3 casas contienen 50 ventanas**

# CATEDRA INFORMATICA CBI FACET - UNT

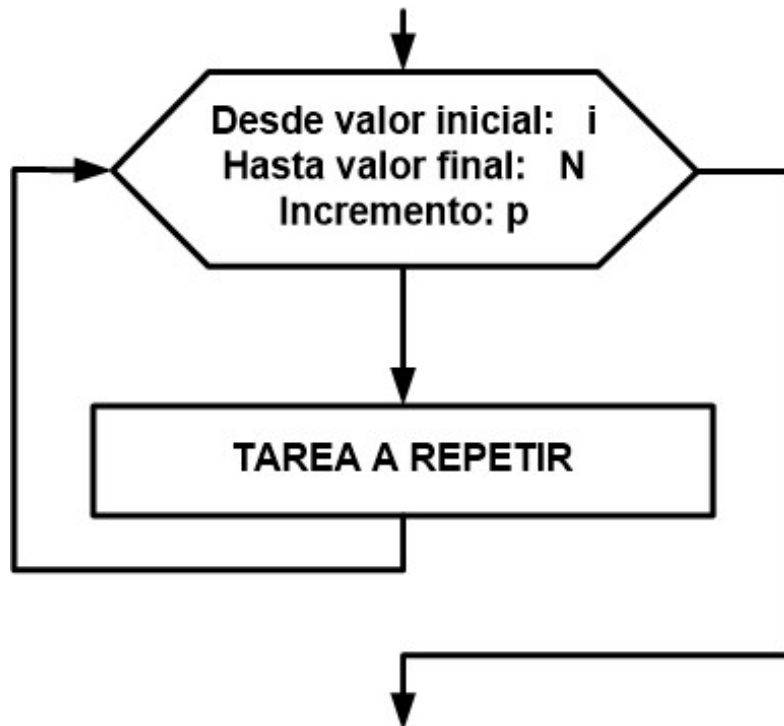
## CÓMO PASAR EL ALGORITMO A CODIGO

Para  
For Next

### ESTRUCTURA DE REPETICION FOR NEXT

/\*Dante Fernandez,  
13/5/21,  
Ejemplo de FOR NEXT (Para)\*/  
#include <stdio.h>

```
int main() {  
    int x,i;  
    printf("Ingrese valor de x= ");  
    scanf("%d",&x);  
    for (i=1;i<=x;i++)  
    {  
        printf("%d\n",i);  
    }  
    return (0);  
}
```





# CATEDRA INFORMATICA CBI FACET - UNT

## DETALLES SOBRE ESTE PROGRAMA (3)

1- EL COMANDO *printf* TIENE VARIOS PARAMETROS QUE PUEDEN SER ACTIVADOS. VER LA SIGUIENTE TABLA:

Código	Significado	Código	Significado
'\n'	nueva línea	'\''	comilla simple
'\r'	retorno de carro	'\"'	comillas
'\f'	nueva página ( <i>form feed</i> )	'\\ '	barra invertida
'\t'	tabulador horizontal	'\? '	interrogación
'\b'	retroceso ( <i>backspace</i> )		
'\nnn'	cualquier carácter (donde nnn es el código ASCII expresado en octal)		
'\xnn'	cualquier carácter (donde nn es el código ASCII expresado en hexadecimal)		

2- EN LA ESTRUCTURA FOR NEXT QUEDA CLARO QUE HAY QUE CONFIGURAR: UNA ASIGNACION, UNA COMPARACIÓN LOGICA Y UN INCREMENTO (PASO)



# CATEDRA INFORMATICA CBI FACET - UNT

## DETALLES SOBRE C (1)

### EXPRESIONES ARITMÉTICAS

Operación	Símbolo	en c
multiplicación	*	*
división	/	/ (al menos un operando real)
división entera	div	/ (Todos los operandos enteros)
módulo (resto de la división entera)	mod	%
suma	+	+
resta	-	-

# CATEDRA INFORMATICA CBI FACET - UNT

## DETALLES SOBRE C (2)

### EXPRESIONES DE RELACIÓN

Operación	Símbolo	en C
igual que	=	==
distinto que	<>	!=
mayor que	>	>
mayor o igual que	>=	>=
menor	<	<
menor o igual que	<=	<=

**TANTO LAS EXPRESIONES DE RELACIÓN COMO LAS EXPRESIONES LÓGICAS,  
SE USAN PARA PREGUNTAR EN TODOS LOS ROMBOS DE DECISIÓN**

# CATEDRA INFORMATICA CBI FACET - UNT

## DETALLES SOBRE C (3)

### EXPRESIONES LÓGICAS

Operación	Símbolo	en C
negación	not (no)	!
producto lógico	and (y)	&&
suma lógica	or (o)	

**TANTO LAS EXPRESIONES DE RELACIÓN COMO LAS EXPRESIONES LÓGICAS, SE USAN PARA PREGUNTAR EN TODOS LOS ROMBOS DE DECISIÓN**

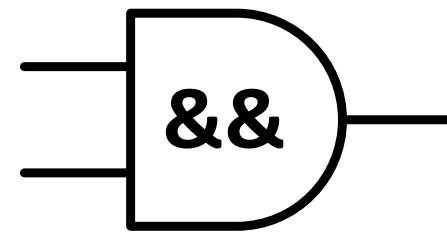
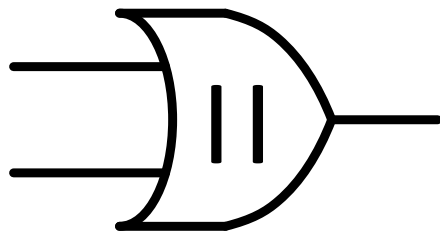
# CATEDRA INFORMATICA CBI FACET - UNT

## DETALLES SOBRE C (3)

### EXPRESIONES LÓGICAS – TABLA DE VERDAD

OR (    )					
ENTRADA 1		ENTRADA 2		SALIDA	
0	F	0	F	0	F
0	F	1	V	1	V
1	V	0	F	1	V
1	V	1	V	1	V

AND ( && )					
ENTRADA 1		ENTRADA 2		SALIDA	
0	F	0	F	0	F
0	F	1	V	0	F
1	V	0	F	0	F
1	V	1	V	1	V



FALSO EQUIVALE A “0” Y VERDADERO EQUIVALE A “1”