



Universidad Autónoma del Estado de México
UAEM

Unidad 4. Estructuras de control

Tema. Estructuras de Iteración (while)

Juan Pablo Cobá Juárez Pegueros

Programación Avanzada

Bioingeniería Médica

Facultad de Medicina

05/09/2017



- ¿Qué son las estructuras de control repetitivas ?
- ¿Cómo se ejecuta?
- ¿Cómo estructura la condición de repetición ?
- Tipos de estructuras de repetición
- Ciclo Mientras
- Diagrama de Flujo y Pseudo código
- Componentes
- Variable de control
- Condición
- Incremento
- Ejemplo
- Evaluación de la condición
- Codificación en c
- Encontrar los factores de un número
- ¿Cuál es el factor de un número?
- Lógica para encontrar todos los factores de un número
- Codificación en c
- Calcular la tabla de multiplicar de un numero
- Lógica para calcular la tabla de multiplicar de un numero
- Codificación en c
- Bibliografía



Objetivo: Aplicar las estructuras de control en el lenguaje de programación C codificando instrucciones sintáctica y semánticamente correctas para controlar el flujo de ejecución en un programa computacional.

Estructuras de Iteración (while)

Unidad 4. Estructuras de control



- Son estructuras que permiten permitir la ejecución repetida de una secuencia de instrucciones que pueden ser:
 - Instrucciones de entrada y salida
 - Estructuras de Selección (if, if-else, switch)



- El número de veces que el bloque de instrucciones se ejecutará se puede especificar de manera **explícita** o a través de **una condición** que indica cuando se ejecuta de nuevo o cuando no



- **explícita** nosotros conocemos el valor inicial y el valor final del ciclo
- **condición** el valor que regrese la condición determina el inicio o fin del ciclo

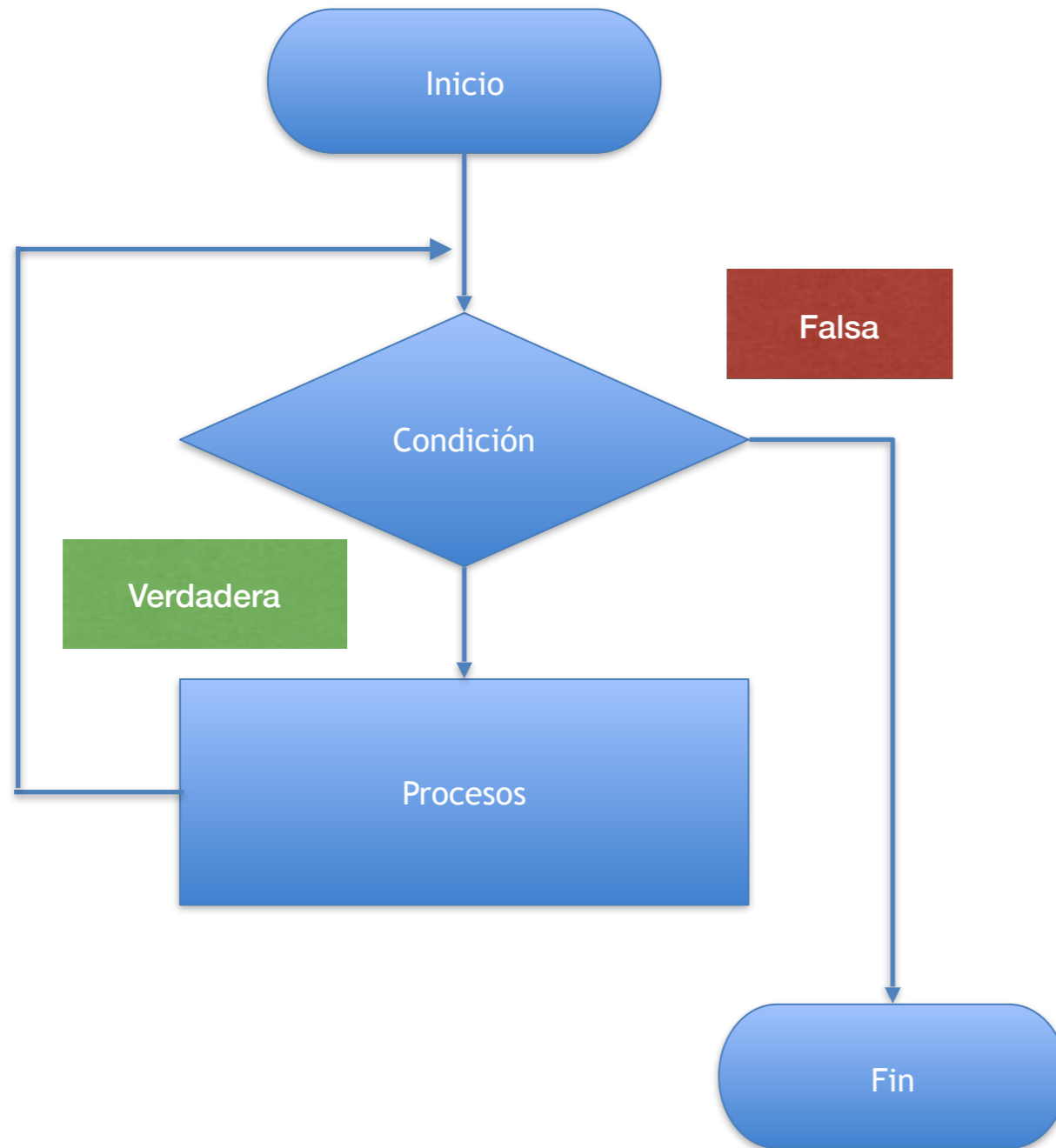


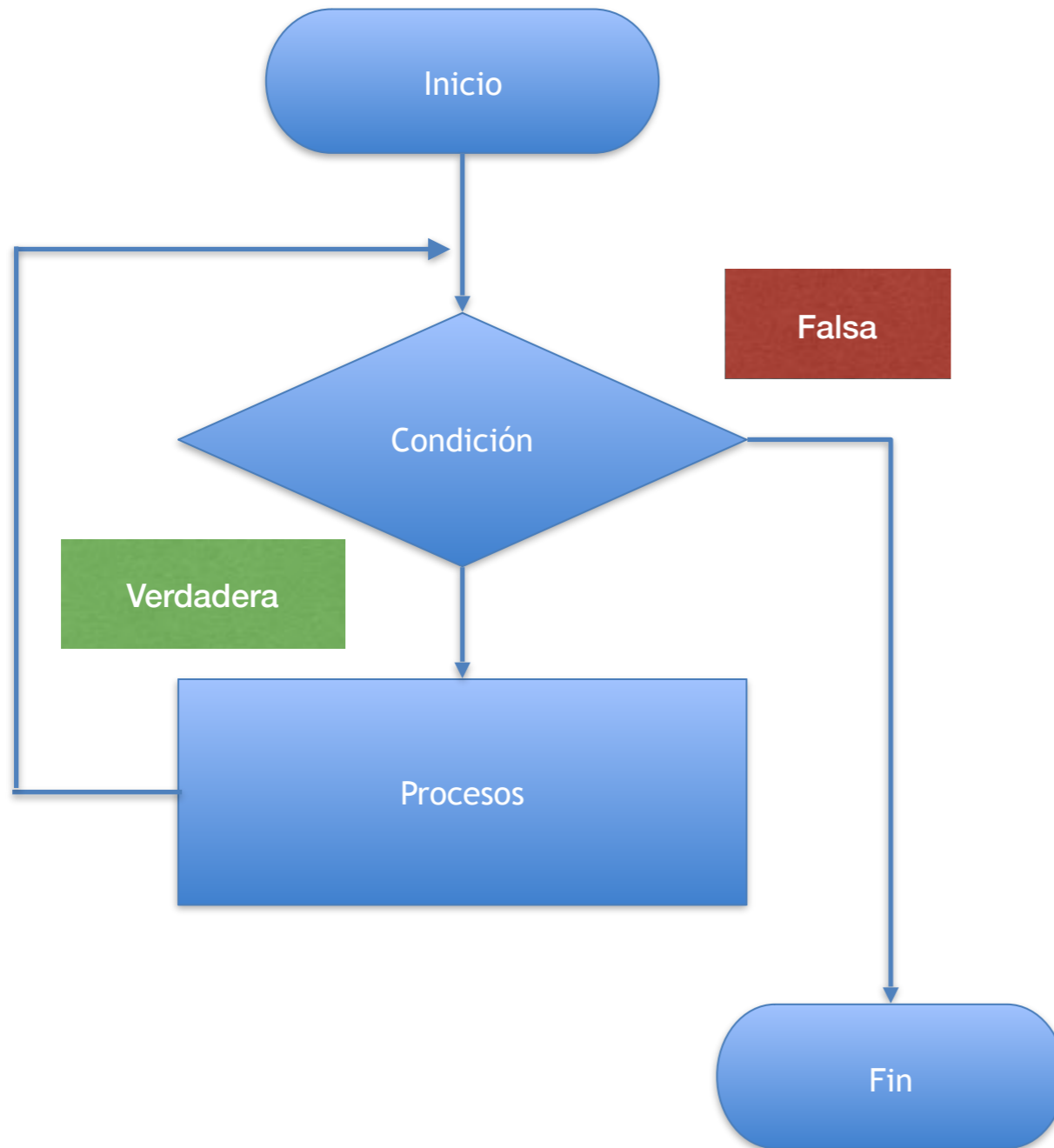
Existen tres tipos principales de sentencias de repetición

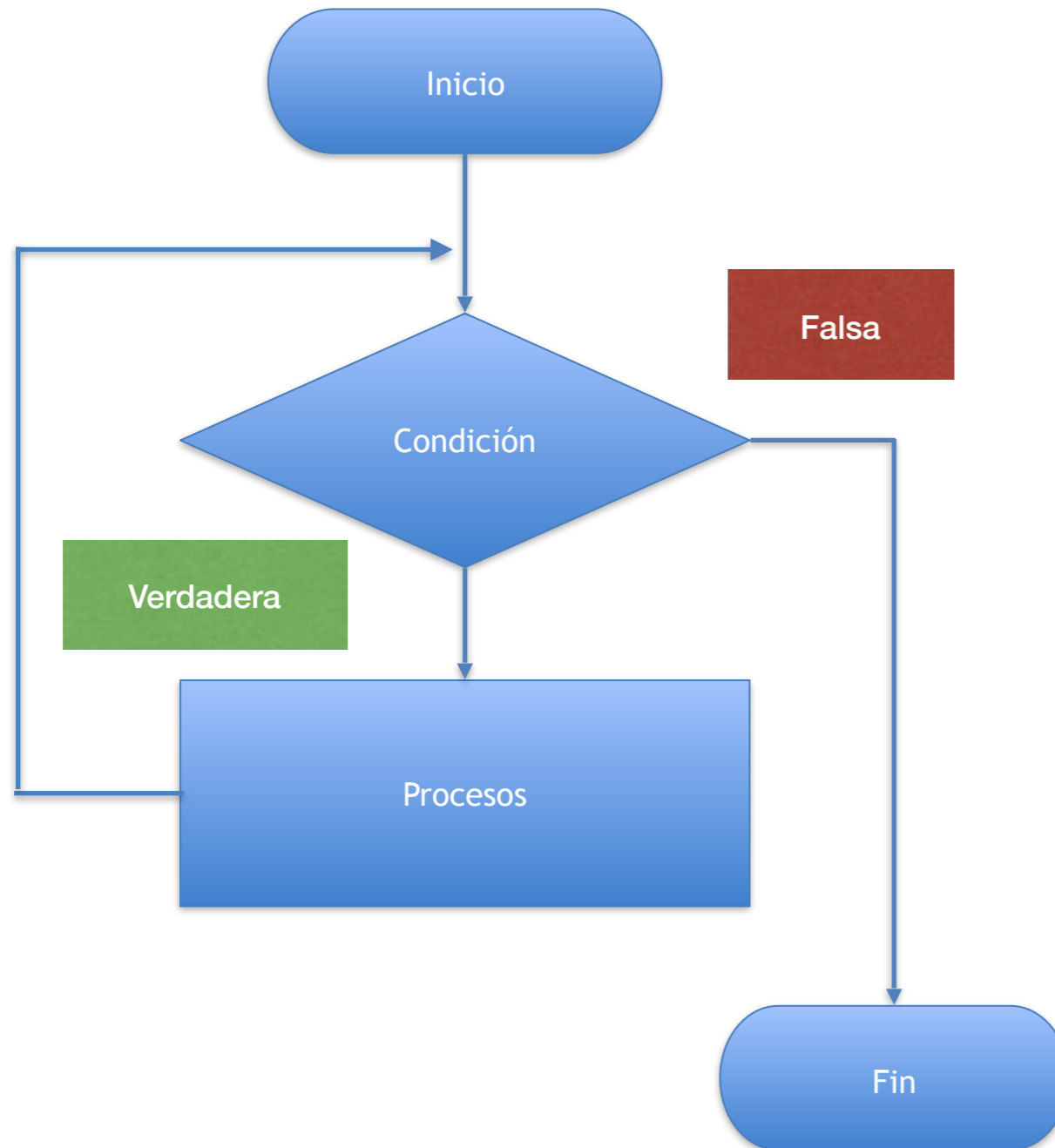
- **Ciclo MIENTRAS**
- **Ciclo REPITE-MIENTRAS**
- **Ciclo PARA**



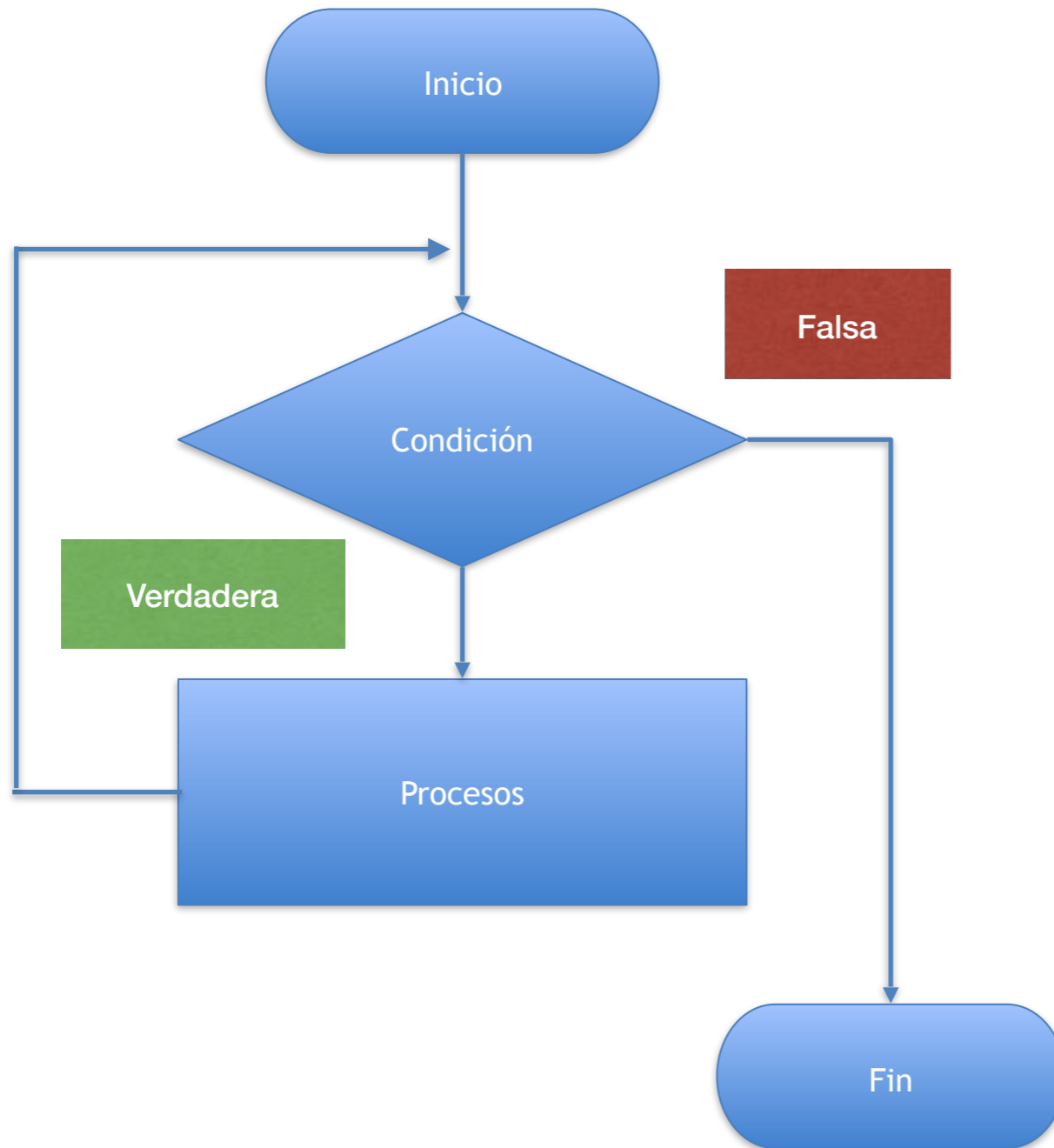
- El CICLO WHILE ejecuta un bloque de acciones ‘mientras’ que una condición dada se cumpla
- Si la condición no se cumple, el bloque no se ejecuta



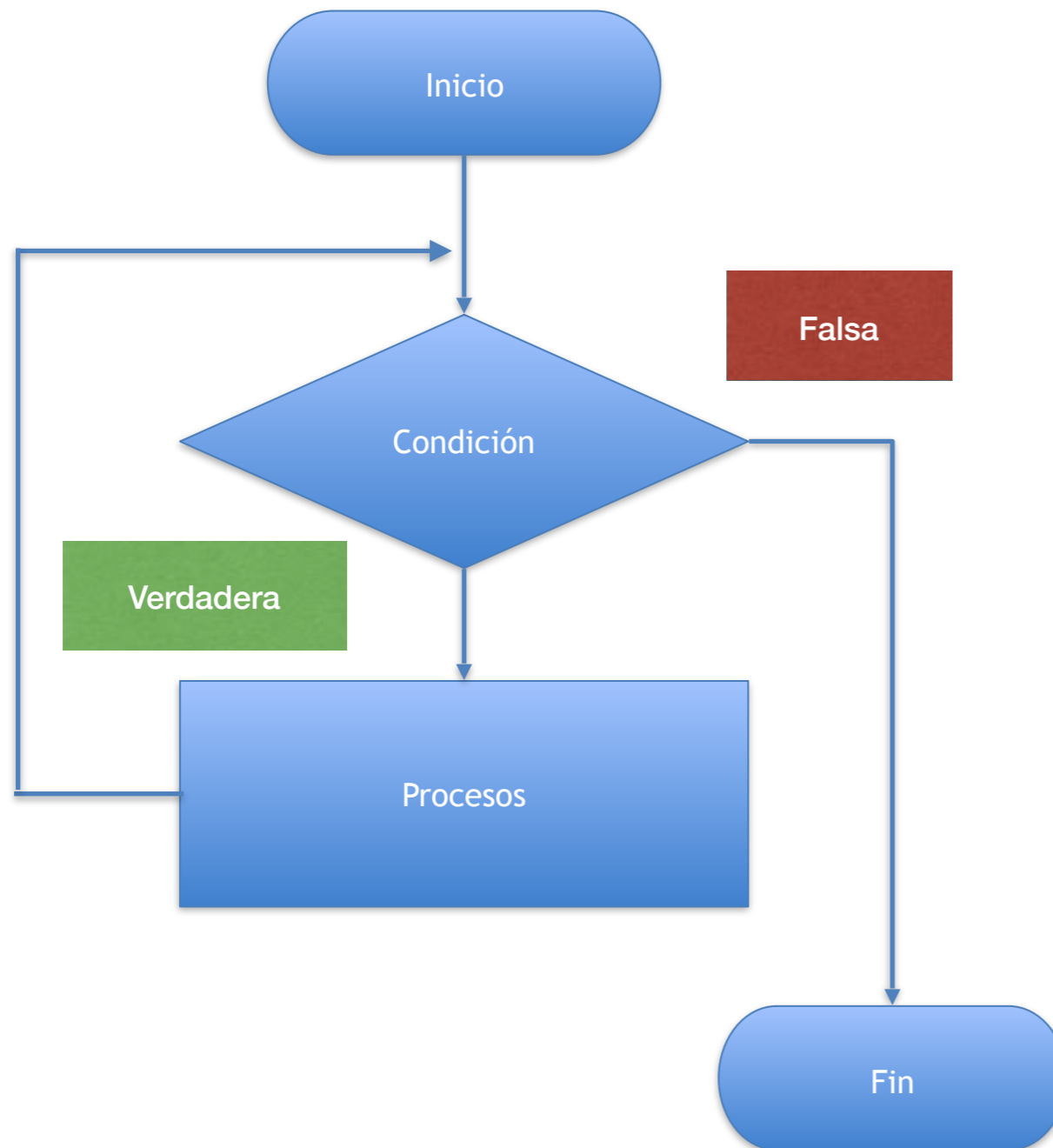




While(condición)



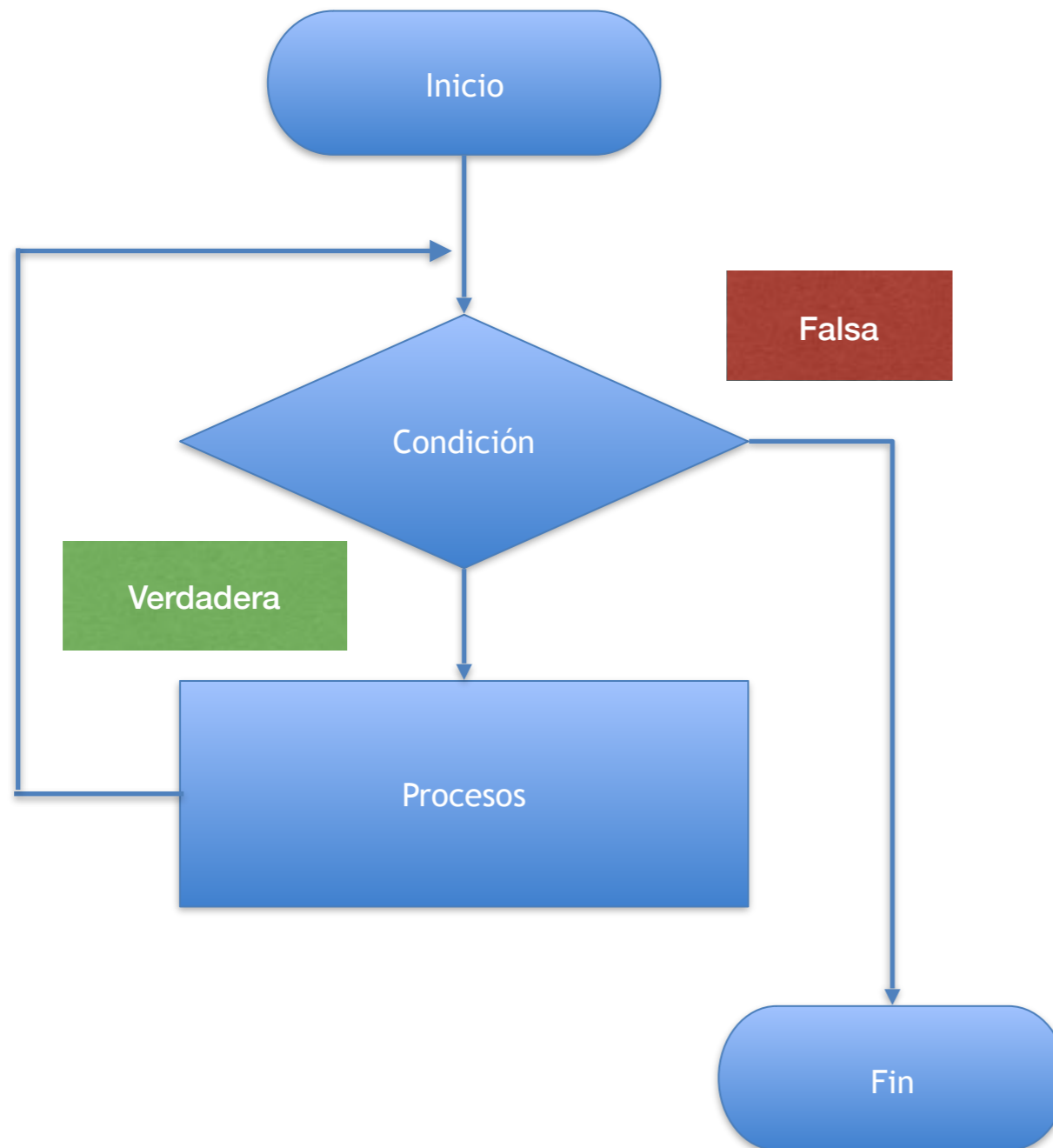
While(condición)
{



While(condición)

{

Bloque de
instrucciones



While(condición)

{

Bloque de
instrucciones

}



- De forma general un ciclo repetitivo while esta conformado de los siguientes componentes:



- De forma general un ciclo repetitivo while esta conformado de los siguientes componentes:
 - Variable de control de ciclo



- De forma general un ciclo repetitivo while esta conformado de los siguientes componentes:
 - Variable de control de ciclo
 - Condición



- De forma general un ciclo repetitivo while esta conformado de los siguientes componentes:
 - Variable de control de ciclo
 - Condición
 - Incremento



- Variable de control se utiliza para “controlar” el flujo de control del programa
- Puede visualizarse como un contador que puede incrementar o decrementar su valor



- Controla el momento en que se repite la estructura y el momento en que debe parar
- Su función es la de un interruptor que se encuentra encendido si la condición es evaluada como verdadera y se cierra de ser falsa.



- Su función es modificar el valor de la variable de control, lo que permite modificar el valor dentro de la condición en cada iteración.



- `#include <stdio.h>`
- `int main(int argc, char *argv[]) {`
- `int i=0;` ← Variable de control
- `int n=0, suma=0;`
- `scanf("%d",&n);`
- `while (i<=n) {` ← Condición
- `suma=suma+i;`
- `i=i+1;` ← Incremento
- `}`
- `printf("el valor de la sum es %d", suma);`
- `}`



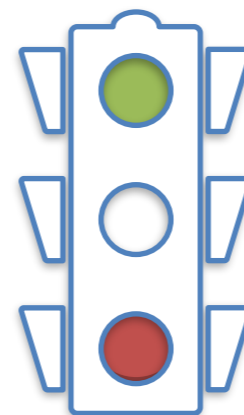
- Crear un algoritmo que solicite un número al usuario, el programa debe finalizar solo si el usuario introduce un número positivo.



- Establecer la entrada de datos
- Se requiere leer un numero



Número	$\text{numero} \leq 0$	Evaluación
-1	$-1 \leq 0$	VERDADERO
0	$0 \leq 0$	VERDADERO
10	$10 \leq 0$	FALSO
4	$4 \leq 0$	FALSO



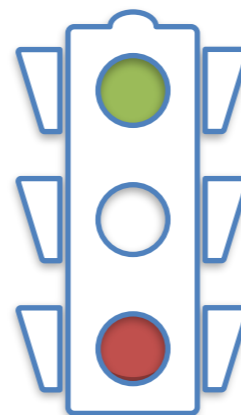
Se repite

Se detiene



- Establecer la condición
- $(\text{numero} \leq 0)$

Número	$\text{numero} \leq 0$	Evaluación
-1	$-1 \leq 0$	VERDADERO
0	$0 \leq 0$	VERDADERO
10	$10 \leq 0$	FALSO
4	$4 \leq 0$	FALSO



Se repite

Se detiene



- Establecer el bloque de instrucciones que se debe repetir
- Imprimir Error
- Imprimir “introduce un número positivo”
- Leer numero



- Establecer el bloque de instrucciones que se debe repetir
- Imprimir Error
- Imprimir “introduce un número positivo”
- Leer numero





- `#include <stdio.h>`
- `int main(int argc, char *argv[]) {`
- `int i=0;`
- `int num=0, suma=0;`
- `printf("introduce un número positivo\n");`
- `scanf("%d",&num);`
- `while (num<=0) {`
- `printf("Error\n");`
- `printf("introduce un número positivo\n");`
- `scanf("%d",&num);`
- `}`
- `}`



Universidad Autónoma del Estado de México
UAEM

Ejercicios



Escriba un algoritmo que permita ingresar un número del usuario y encuentre todos los factores del número dado usando el ciclo de repetición **while** que permita encontrar factores de un número



El factor de cualquier número es un número entero que divide exactamente el número en un número entero sin dejar ningún resto.

Por ejemplo: 2 es un factor de 6 porque 2 divide 6 exactamente dejando ningún residuo.



Paso a paso la lógica descriptiva para encontrar todos los factores de un número.

- 1.Introduzca el número del usuario. Guárdelo en la variable num**
- 2.Ejecutar el ciclo mientras i sea menor o igual que num. La condición del ciclo debe ser similar a ($i \leq \text{num}$)**
- 3.Para cada iteración dentro de ciclo compruebe si la variable de control del ciclo i es un factor de num o no. Para comprobar el factor, verificamos la divisibilidad del número realizando la división del módulo, es decir, si ($\text{num} \% i == 0$) i es un factor de num.**
Si i es un factor de num, entonces imprime el valor de i.
- 4.Incrementar i en 1**
- 5.Establecer el fin del ciclo**



```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {  
    int i=1;  
    int num;  
    printf("introduce un digito");  
    scanf("%d",&num);  
    while (i <= num)  
    {  
        if ((num%i) == 0) {  
            printf("%d ",i);  
        }  
        i=i+1;  
    }  
}
```



Escriba un algoritmo que permita introducir un número al usuario y calcule la tabla de multiplicación del número dado usando el ciclo While .



1. Establecer i como la variable de control e iniciarla en 1
2. Inicializar otra variable para almacenar producto, es decir, $\text{producto} = 1$
3. Introduzca el número del usuario. Guárdelo en la variable num
4. Mientras la condición $(i \leq 10)$ sea verdadera Multiplique num por i y guarde el resultado en la variable producto
5. Imprime el valor de la variable producto .
6. Incrementa la variable de control $i \leftarrow i + 1$
7. Establecer el fin del ciclo



```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int i=1;
    int num;
    int producto=0;
    printf("introduce un digito");
    scanf("%d",&num);

    while (i <= 10){
        producto =num*i;
        printf("%d ",producto);
        i=i+1;
    }
}
```



Universidad Autónoma del Estado de México
UAEM

Gracias !!!



- Kernighan y Ritchie 1995: El lenguaje de programación C, México, Prentice Hall.
- Cairó, Osvaldo. 2006: Fundamentos de Programación. Piensa en C, México: Pearson Prentice Hall
- Paul Deitel 2009: C:How to program, US: Prentice Hall Samuel Harbison 2002: C: A reference manual, US: Prentice Hall
- Aitken y Jones. 1994: Aprendiendo C en 21 días, México: Prentice Hall
- Gottfried, Byron. 2005: Programación en C (2a. Edición Revisada). México: Mc Graw Hill
- Joyanes, Aguilar Luis y Zahonero, Martínez I. 2005: Programación en C, Metodología, Estructura de Datos y Objetos, México: McGraw Hill.
- Joyanes, Aguilar Luis y Zahonero, Martínez I. 2002: Programación en C, Libro de problemas, México: McGraw Hill

PROGRAMACIÓN AVANZADA

Diapositivas

Material Didáctico sólo visión

Nombre del material:

Estructuras de Iteración while.

Contenido

Presentación.....2
Objetivo de la asignatura3
Guion explicativo para el uso del material..... 3
Orden de las diapositivas.....3

Juan Pablo Cobá Juárez Pegueros
Facultad de Ingeniería
Bioingeniería Médica

Presentación

El perfil de egreso del Licenciado en Bioingeniería Médica es el de un profesional que colabora, con equipos multidisciplinarios, en el cuidado de la salud y la mejora de la calidad de vida del ser humano, en particular, en el ámbito de la tecnología aplicada al servicio médico directo al paciente (diagnóstico, prevención, tratamiento y rehabilitación).

Por lo anterior es importante formar profesionales de la Bioingeniería Médica, con un alto sentido de responsabilidad, de ética y vocación de servicio, y con las competencias y aprendizajes necesarias.

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras.

Objetivo General de la Unidad de Aprendizaje

Implementar programas utilizando estructuras de datos y estructuras de control. Utilizando un lenguaje de alto nivel para la solución de problemas de bioingeniería

Guion explicativo para el uso del material didáctico

Las unidades del programa son las siguientes:

UNIDAD 1. INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN C
UNIDAD 2. TIPOS, OPERADORES Y EXPRESIONES
UNIDAD 3. ENTRADA Y SALIDA DE DATOS
UNIDAD 4. ESTRUCTURAS DE CONTROL
UNIDAD 5. ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES
UNIDAD 6. PROGRAMACIÓN MODULAR

El material se encuentra enfocado exclusivamente a la Unidad 4, específicamente en el tema 3 Estructuras de Iteración while

Este material está dirigido a toda persona interesada en el tema, pero específicamente a los alumnos de la unidad de aprendizaje Programación Avanzada.

El uso de este material es sencillo, ya que sólo contiene imágenes e ideas centrales del tema, que facilitan la concentración del alumno.

Orden de las diapositivas

1. ¿Qué son las estructuras de control repetitivas?
2. ¿Cómo se ejecuta?
3. ¿Cómo estructura la condición de repetición?
4. Tipos de estructuras de repetición
5. Ciclo Mientras
6. Diagrama de Flujo y Pseudo código
7. Componentes
8. Variable de control
9. Condición
10. Incremento
11. Ejemplo
12. Evaluación de la condición
13. Codificación en c
14. Encontrar los factores de un número
15. ¿Cuál es el factor de un número?
16. Lógica para encontrar todos los factores de un número
17. Codificación en c
18. Calcular la tabla de multiplicar de un numero
19. Lógica para calcular la tabla de multiplicar de un numero
20. Codificación en c
21. Bibliografía