



Universidad Autónoma del Estado de México

## Unidad 4. Estructuras de control

### Tema. Estructuras de Iteración (mientras ejercicios validación)

Juan Pablo Cobá Juárez Pegueros

Programación Avanzada

Bioingeniería Médica

Facultad de Medicina

05/09/2018



- Introducción
- ¿Cómo se ejecuta?
- ¿Cómo se establece una condición ?
- Diagrama de Flujo y Pseudo código
- Componentes
- Variable de control
- Incremento
- Validación
- Identificación de rangos
- Interpretación
- Ejemplo
- Pseudocódigo
- Ejemplo
- Pseudocódigo
- Bibliografía



Aplicar las estructuras de control computacionales utilizando instrucciones sintáctica y semánticamente correctas para controlar el flujo de control en un programa computacional expresado en diagrama de flujo y/o pseudocódigo, utilizar la técnica de pruebas de escritorio para validar la propuesta de solución

Estructuras de Iteración (mientras)

Unidad 4. Estructuras de control



Mucho de los problemas requieren que un proceso se efectúe de manera **repetitiva**, en ese momento es que se deben emplear estructuras que permitan controlar el **número de veces que estructura se emplea** con base en las **condiciones propias de cada problema**

.



- El número de veces que el bloque de instrucciones se ejecutará en el ciclo mientras se debe especificar a través de **una condición**,
- **Mientras** sea verdadera se ejecutará y cuando cambia y es **evaluada como falsa** se sale del ciclo y continua con las siguientes instrucciones.



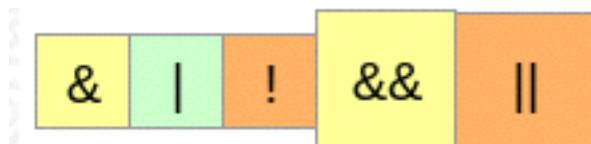
- Se debe hacer uso de los operadores relacionarles y lógicos





- Se puede crear relación con valor fijo

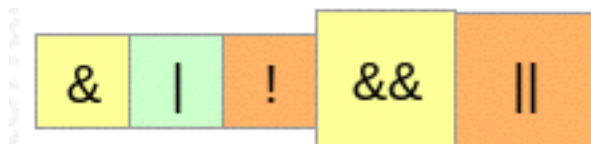
Variable	<	10
Variable	>	10
Variable	==	10
Variable	<=	10
Variable	>=	10
Variable	!=	10





- Se puede crear relacionando dos variables

Variable1	<	Variable2
Variable1	>	Variable2
Variable1	==	Variable2
Variable1	<=	Variable2
Variable1	>=	Variable2
Variable1	!=	Variable2





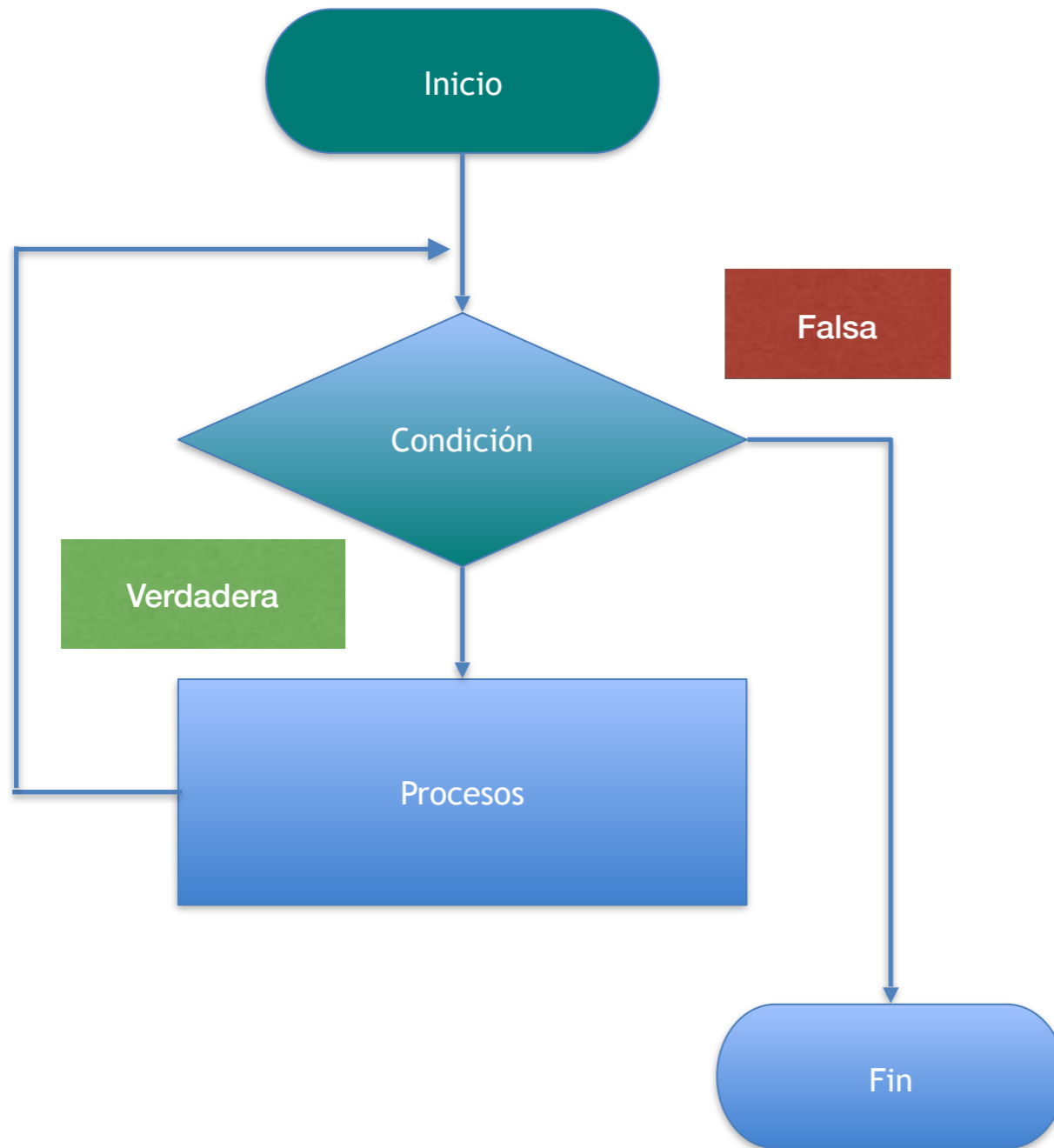


- Se puede crear relacionando dos variables y utilizando algún operador lógico

Variable1	<	Variable2	&&	Variable3	<	Variable4
Variable1	>	Variable2		Variable3	>	Variable4
Variable1	==	Variable2	&&	Variable3	==	Variable4
Variable1	<=	Variable2		Variable3	<=	Variable4
Variable1	>=	Variable2	&&	Variable3	>=	Variable4
Variable1	!=	Variable2		Variable3	!=	Variable4

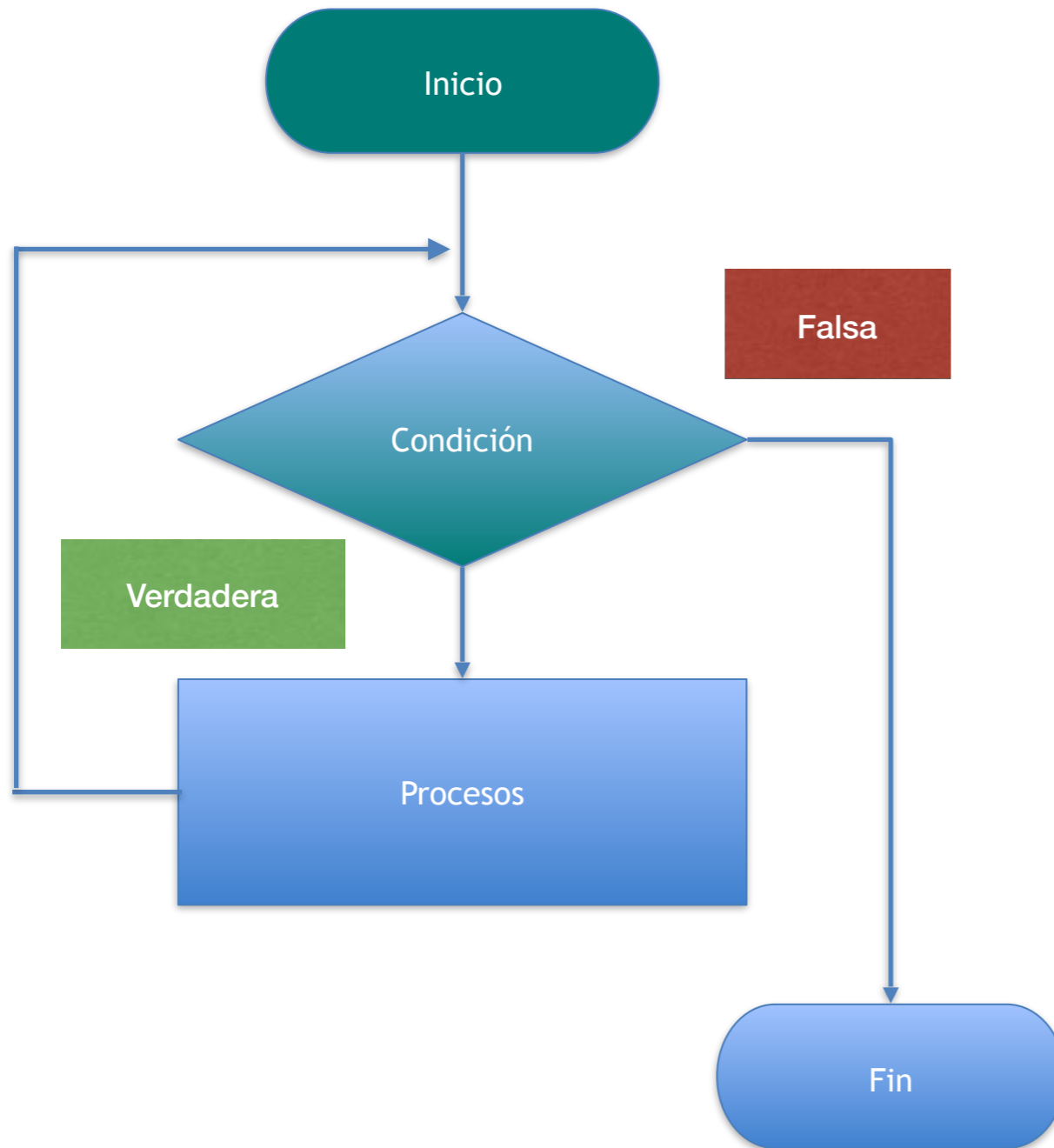


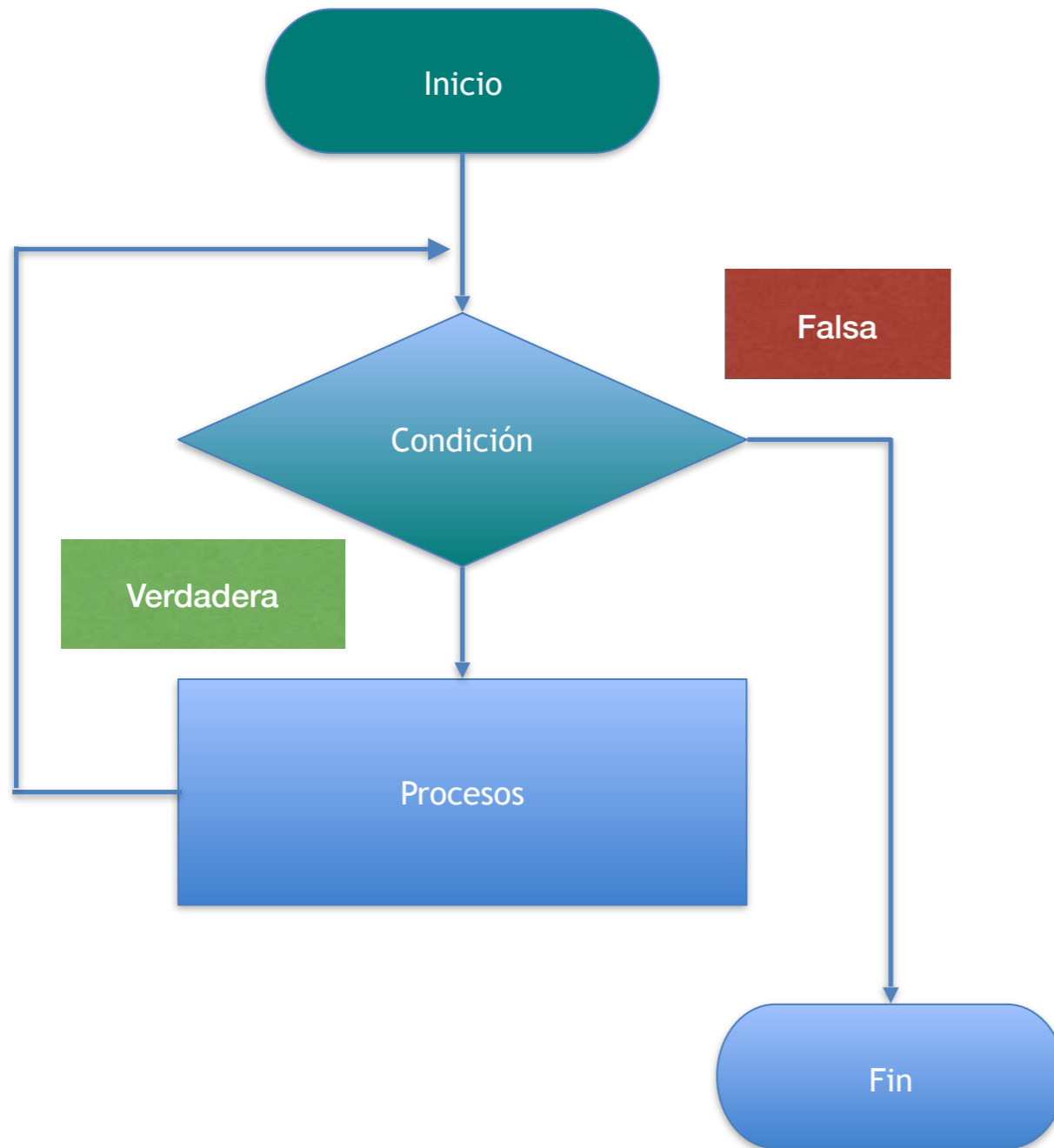
# Diagrama de Flujo y Pseudo código



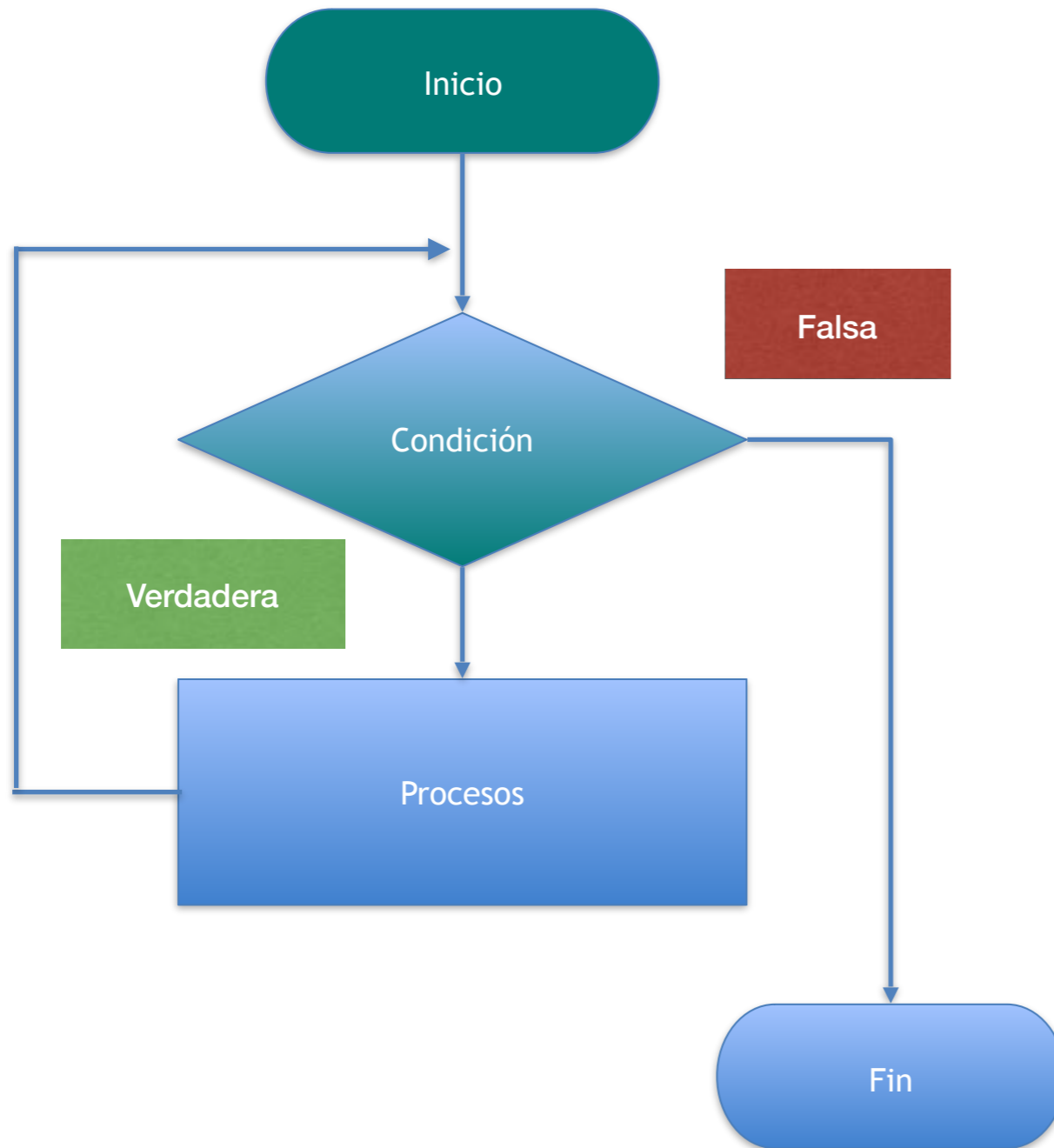


# Diagrama de Flujo y Pseudo código

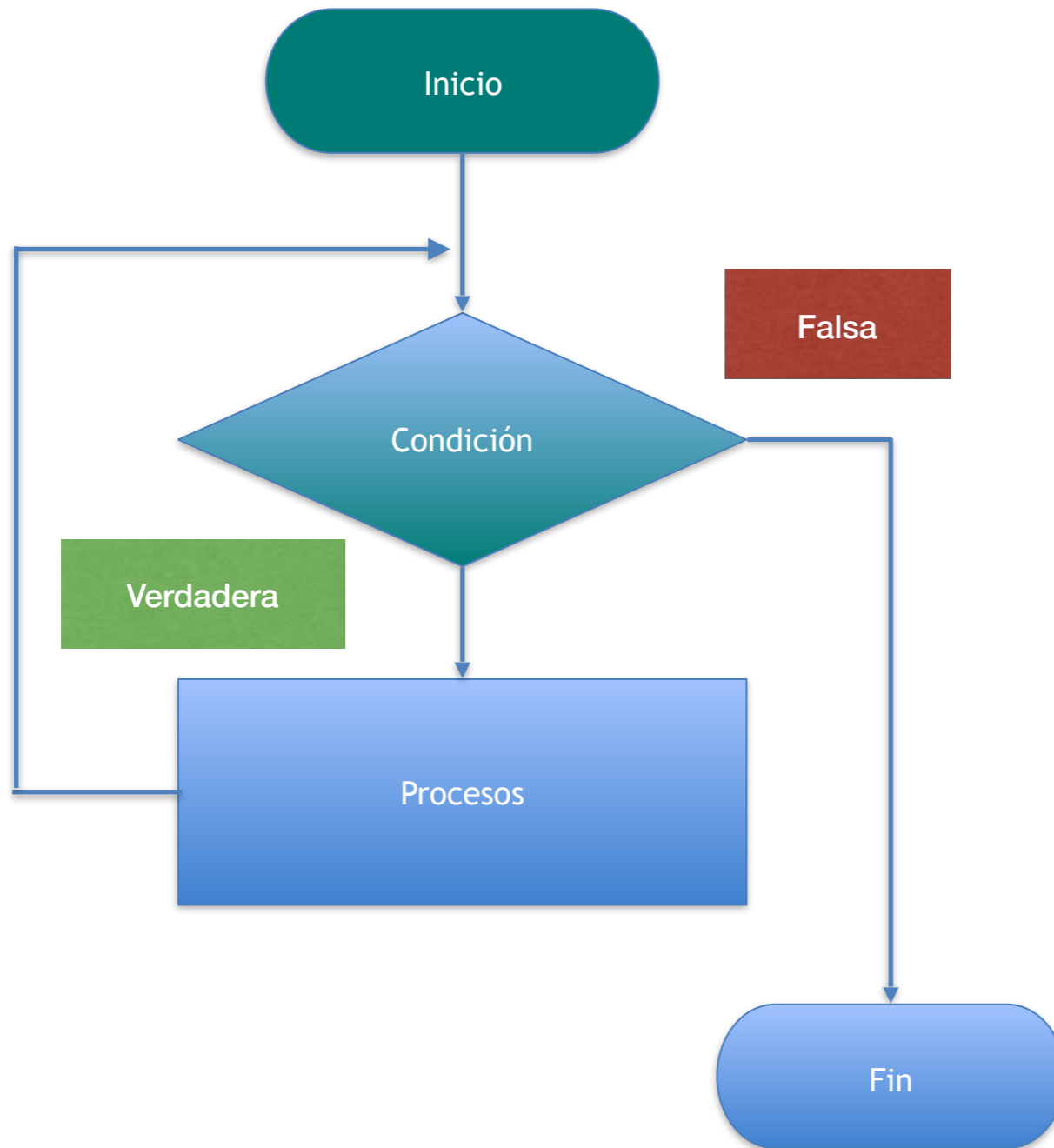




**Mientras( condición)**



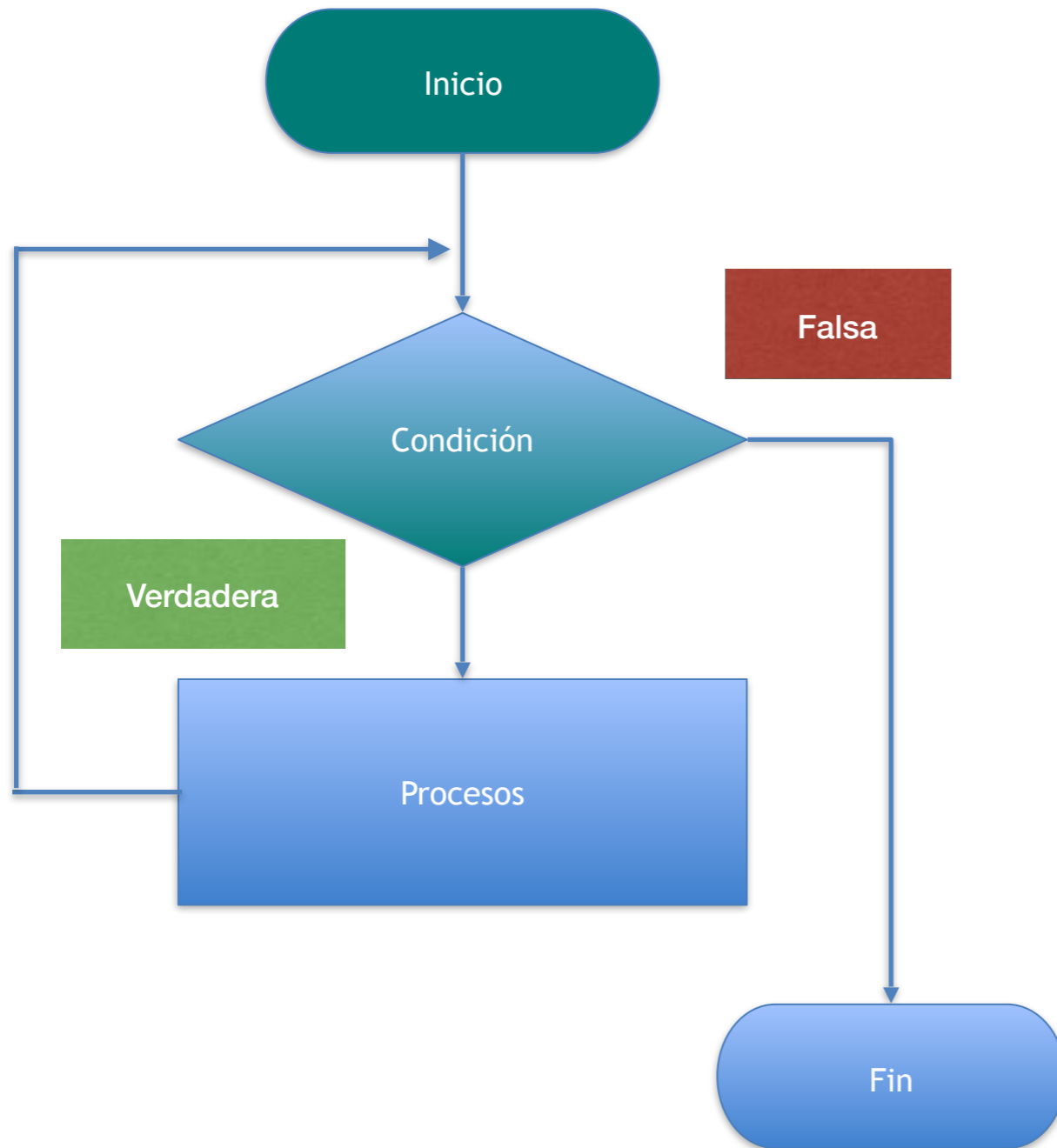
**Mientras( condición)**  
{



**Mientras( condición)**

{

Bloque de instrucciones



**Mientras( condición)**

{

Bloque de instrucciones

}



Universidad Autónoma del Estado de México

---

# Componentes





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



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



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



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



- De forma general un ciclo repetitivo mientras 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.





Universidad Autónoma del Estado de México

# Validación



- La validación de datos introducidos por el usuario es una tarea crucial al desarrollar algoritmos más complejos
- Ayuda a prevenir errores.
- Se requiere conocer los rangos



- Identificar los valores que son validos
- Identificar que valores no son válidos
- Puede utilizarse una recta para idenficarlos gráficamente

Valores inválidos

Rango de valores válidos

Valores inválidos



- Crear condición que satisfaga el rango de la caja naranja

Valores inválidos

Rango de valores válidos

Valores inválidos

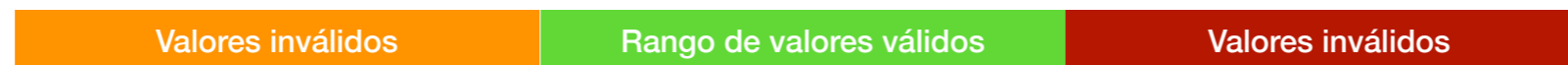
**Variable1**

>

**Variable2**



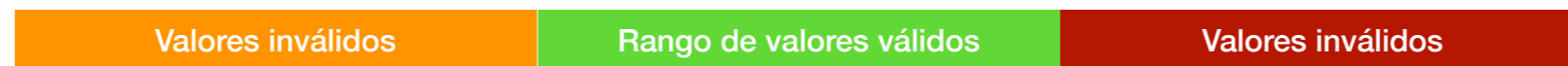
- Crear condición que satisfaga el rango de la caja roja



Variable1  Variable2

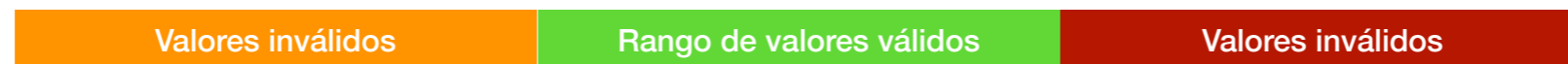


- Establecer la relación lógica para la validación



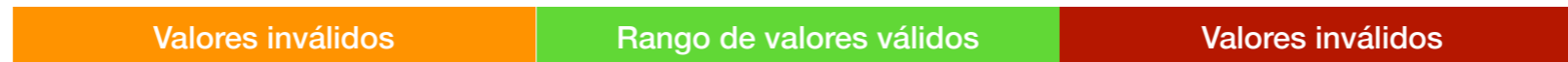


- El lado naranja verifica si el valor es menor al rango
- El lado rojo verifica si es mayor





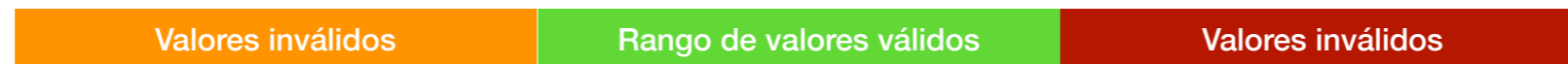
- El operador || (o) lo ocupamos debido a que no existe un numero que cumpla con ambas condiciones es decir
- 
- No existe número menor al rango y a la vez mayor al rango valido







- Es posible hacer esta operación lógico relacionala con valores fijos
- En el ejemplo podemos ver que se limita a valores menores a 10 y mayores a cero
- Los valores correctos 1 a 9





Universidad Autónoma del Estado de México

---

**Ejemplo**



- entero numero < -0
- Imprimir “introduce un numero ”
- Leer numero
- Mientras (numero < 0 || numero > 10) {
  - Imprimir “el número es erróneo ”
  - Imprimir “introduce otro número ”
  - Leer numero
- Fin mientras



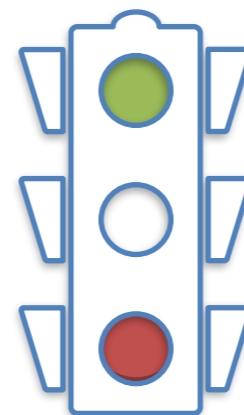
- Crear un algoritmo que valide que la edad de un niño oscila entre 1 a 17 años.



- Establecer la entrada de datos
- Se requiere leer la edad



Edad	$edad < 0 \    \    \ edad > 17$	Evaluación
-1	$-1 \leq 0$	VERDADERO
0	$0 \leq 0$	VERDADERO
10	$10 \leq 0$	FALSO
4	$4 \leq 0$	FALSO



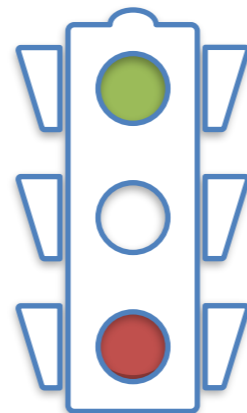
Se repite

Se detiene



- Establecer la condición
- $(edad < 0 \parallel edad > 17)$

Edad	$edad < 0 \parallel edad > 17$	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



Universidad Autónoma del Estado de México

**Ejemplo**





- Establecer el bloque de instrucciones que se debe repetir
  - Imprimir “el número es erróneo ”
  - Imprimir “introduce otro número ”
  - Leer numero
-



Universidad Autónoma del Estado de México

# Pseudocódigo



- entero edad ← -0
- Imprimir “introduce la edad”
- Leer edad
- Mientras (edad < 0 || numero > 18) {
  - Imprimir “ya es un adulto ”
  - Imprimir “introduce la edad ”
  - Leer edad
- Fin mientras

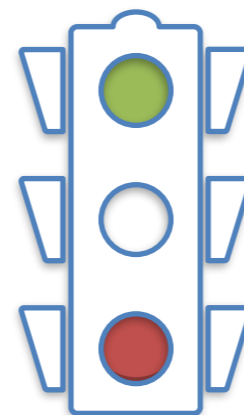


- Establecer la entrada de datos
- Se requiere leer la contraseña



Hola1

Pasword	Password! ="hola1"	Evaluación
Casa	Casa!="Hola1"	VERDADERO
Vaca	Vaca!="Hola1"	VERDADERO
Hola1	Hola1!="Hola1"	FALSO
Hola1	Hola1!="Hola1"	FALSO



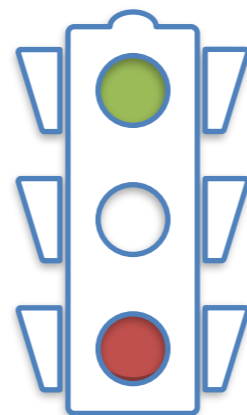
Se repite

Se detiene



- Establecer la condición
- (contraseña != "Hola1")

Pasword	Password! ="hola1"	Evaluación
Casa	Casa!="Hola1"	VERDADERO
Vaca	Vaca!="Hola1"	VERDADERO
Hola1	Hola1!="Hola1"	FALSO
Hola1	Hola1!="Hola1"	FALSO



Se repite

Se detiene



Universidad Autónoma del Estado de México

**Ejemplo**



- Establecer el bloque de instrucciones que se debe repetir
  - Imprimir “el número es erróneo ”
  - Imprimir “introduce otro número ”
  - Leer numero
-





Universidad Autónoma del Estado de México

# Pseudocódigo



- cadena password ← “Hola1”
- Imprimir “la contraseña ”
- Leer password
- Mientras (password ≠ “hola1”)
  - Imprimir “contraseña invalida ”
  - Imprimir “introduce la contraseña ”
  - Leer password
- Fin mientras



Universidad Autónoma del Estado de México

**Gracias !!!**



Cairo, O., (2008), Metodología de la programación: Algoritmos, Diagramas de Flujo y programas, México, Alfaomega.

Norton, P., (2006), Introducción a la computación, México, McGraw-Hill.

Joyanes, A. L., (2008), Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos, Madrid, McGraw-Hill.

Ramírez, F., (2007), Introducción a la programación: Algoritmos y su implementación en VB.NET, C#, JAVA Y C++, México, Alfaomega.

Rodríguez, B. L., Fernández A. M. y Joyanes A. L., (2003), Fundamentos de programación. Libro de problemas. España, McGrawHill.

Criado, Ma. A., (2006), Programación en lenguajes estructurados, México, Alfaomega.

López, R. L., (2004), Programación estructurada. Un enfoque algorítmico, México, Alfaomega.

# **ALGORITMOS Y PROGRAMACIÓN BÁSICA**

## **Diapositivas**

Material Didáctico sólo visión

Nombre del material:

**Iteración o Repetitivas – mientras validación.**

## **Contenido**

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

Juan Pablo Cobá Juárez Pegueros  
Facultad de Ingeniería  
Bioingeniería Médica

## **Presentación**

Hoy en día es de gran importancia en el mundo de la Bioingeniería Médica formar profesionales con un alto sentido de responsabilidad, de ética y vocación de servicio, y con las competencias y aprendizajes para diseñar y construir tecnología para solucionar problemas y necesidades específicas en los campos de intervención e investigación médica, libre de riesgos sanitarios

Por lo anterior los alumnos deben conocer, analizar y resolver mediante algoritmos y programación problemas asociados con los sistemas de la bioingeniería.

## **Objetivo general de la unidad de aprendizaje**

Solucionar problemas mediante el desarrollo de algoritmos en pseudocódigo, verificando su funcionalidad mediante la prueba de escritorio.

## Guion explicativo para el uso del material didáctico

Las unidades del programa son las siguientes:

UNIDAD 1. ARQUITECTURA DE COMPUTADORAS Y RESOLUCIÓN DE PROBLEMAS MEDIANTE ALGORITMOS  
UNIDAD 2. TIPOS, OPERADORES Y EXPRESIONES  
UNIDAD 3. ESTRUCTURAS DE CONTROL EN DIAGRAMA DE FLUJO Y PSEUDOCÓDIGO  
UNIDAD 4. ARREGLOS  
UNIDAD 5. INTRODUCCIÓN A LA PROGRAMACIÓN MODULAR

El material se encuentra enfocado exclusivamente a la Unidad 3, específicamente en el tema 3 Iteración o Repetitivas mientras.

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

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. Introducción
2. ¿Cómo se ejecuta?
3. ¿Cómo se establece una condición? 1
4. ¿Cómo se establece una condición? 2
5. ¿Cómo se establece una condición? 3
6. Diagrama de Flujo y Pseudo código
7. Componentes
8. Variable de control
9. Incremento
10. Validación
11. Identificación de rangos 1
12. Identificación de rangos 2
13. Identificación de rangos 3
14. Identificación de rangos 4
15. Interpretación
16. Ejemplo 1
17. Pseudocódigo
18. Ejemplo 2
19. Pseudocódigo
20. Bibliografía