



Universidad Autónoma del Estado de México
UAEM

Unidad 3. Estructuras de control en diagrama de Flujo y pseudocódigo

Estructuras de control

Tema. Estructuras de control selección simple

Juan Pablo Cobá Juárez Pegueros
Algoritmos y programación básica
Bioingeniería Médica
Facultad de Medicina

05/09/2019



- ¿Qué son las estructuras de control ?
- ¿Cuáles son las estructuras de control?
- Estructura de control selección simple SI
- Características de la estructura de control simple
- Pseudocódigo
- Diagrama de flujo
- Ejercicios
 - Ejercicio 1
 - Ejercicio 2
 - Ejercicio 3
- Bibliografía



Objetivo: 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

Tema. Estructuras de control
selección simple

Unidad 3. Estructuras de control en
diagrama de Flujo y pseudocódigo
Estructuras de control



- Son estructuras que permiten decidir que instrucción se ejecutará a continuación.
- Por lo tanto, a veces se denominan expresiones de selección
- Las declaraciones condicionales nos dan el poder de tomar decisiones básicas



- Las sentencias condicionales en algoritmos son :
- Instrucción SI
- Instrucción SI-NO-
- Instrucción SI - multiple



Universidad Autónoma del Estado de México
UAEM

Estructura de control selección simple SI

05/09/2019



- Es la instrucción que permite tomar decisiones simples.
- Son la base de decisiones más complejas
- Permite que el algoritmo cambie su flujo de control.



Universidad Autónoma del Estado de México
UAEM

Pseudocódigo

05/09/2019



SI (condición) entonces

Instrucción-1

Instrucción-2

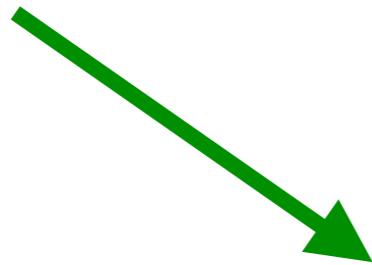
Instrucción-3

Instrucción-n

Fin SI



**SI es una
palabra
reservada**



SI (condición) entonces
Instrucción-1
Instrucción-2
Instrucción-3
Instrucción-n
Fin SI



SI es una
palabra
reservada

La condición debe ser una expresión
relacional, por lo que debe ser
evaluada como **verdadera** o **falsa**

SI (**condición**) entonces
Instrucción-1
Instrucción-2
Instrucción-3
Instrucción-n
Fin SI



SI es una
palabra
reservada

La condición debe ser una expresión
relacional, por lo que debe ser
evaluada como **verdadera** o **falsa**

SI (**condición**) entonces

Instrucción-1

Instrucción-2

Instrucción-3

Instrucción-n

Fin SI

Indica el final de
la estructura



SI es una
palabra
reservada

La condición debe ser una expresión
relacional, por lo que debe ser
evaluada como **verdadera** o **falsa**

SI (**condición**) entonces

Instrucción-1
Instrucción-2
Instrucción-3
Instrucción-n

Fin SI

Indica el final de
la estructura

Si la condición es **verdadera** las
instrucciones se ejecutan



SI es una
palabra
reservada

La condición debe ser una expresión
relacional, por lo que debe ser
evaluada como **verdadera** o **falsa**

SI (**condición**) entonces

Instrucción-1

Instrucción-2

Instrucción-3

Instrucción-n

Fin SI

Indica el final de
la estructura

Si la condición es **verdadera** las
instrucciones se ejecutan
Si es evaluada como **falsa**, las
instrucciones se omiten



Universidad Autónoma del Estado de México
UAEM

Diagrama de flujo

05/09/2019

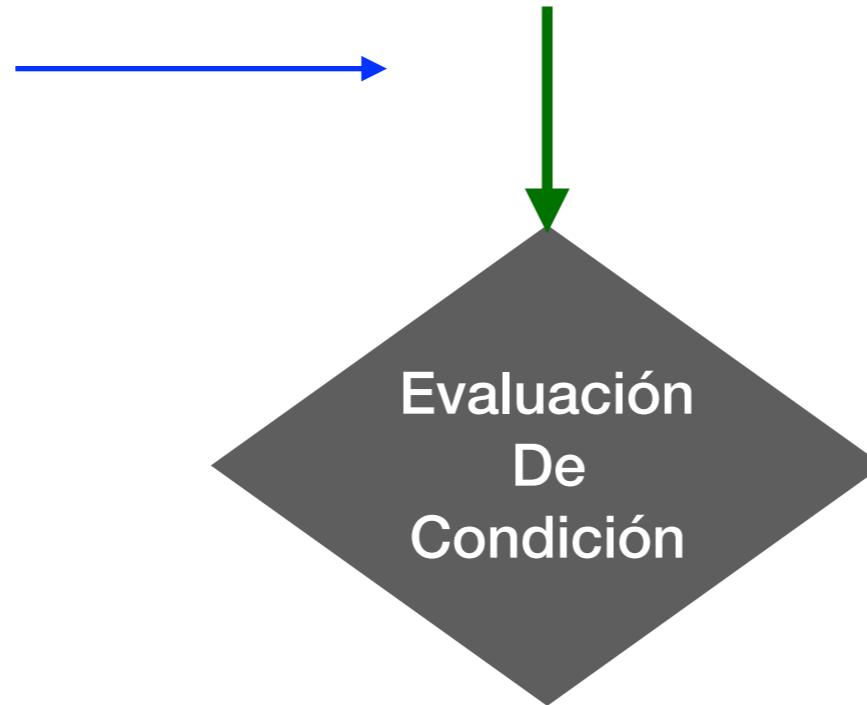


**Inicio de la estructura de
control selección simple**



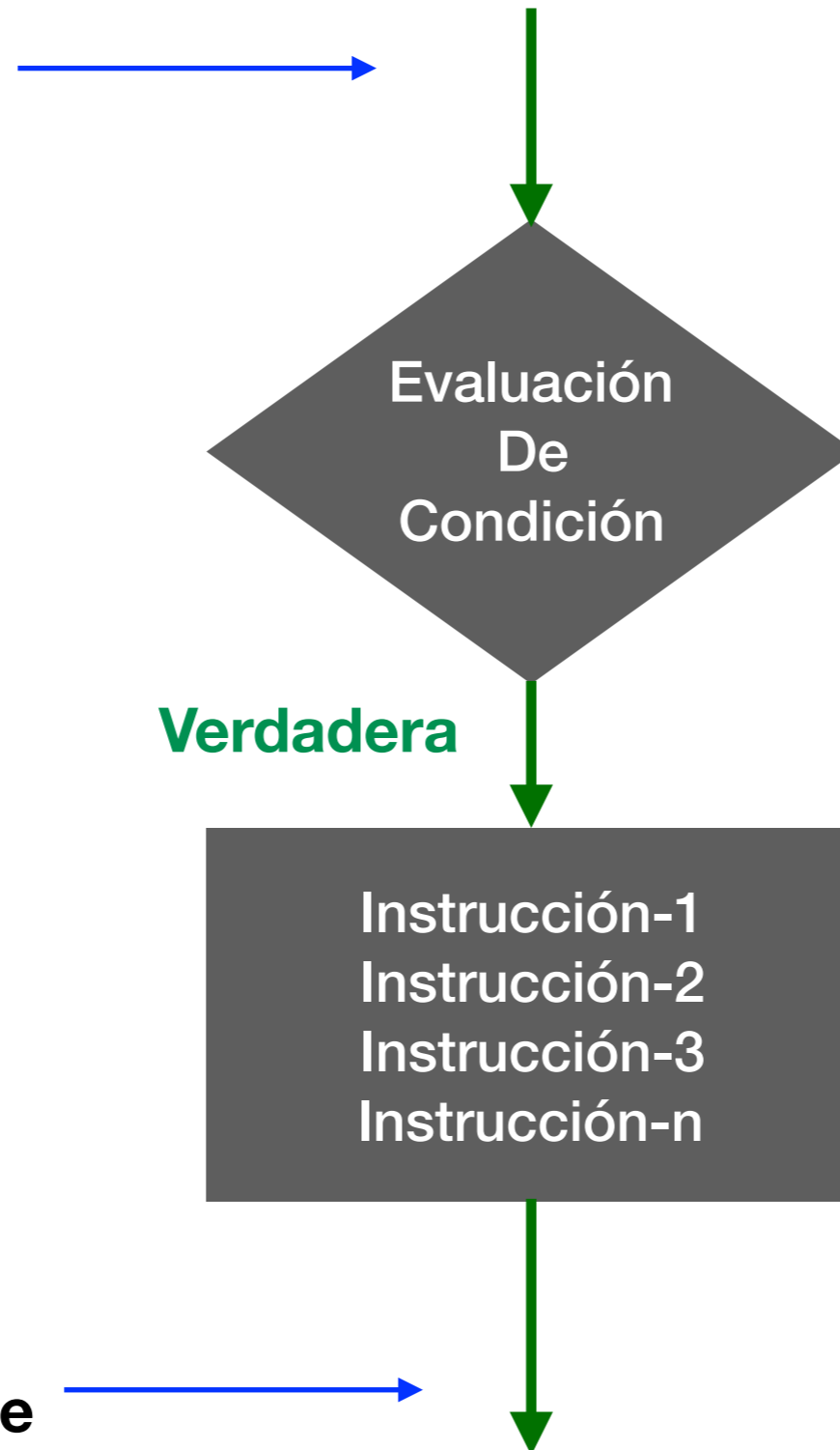


**Inicio de la estructura de
control selección simple**





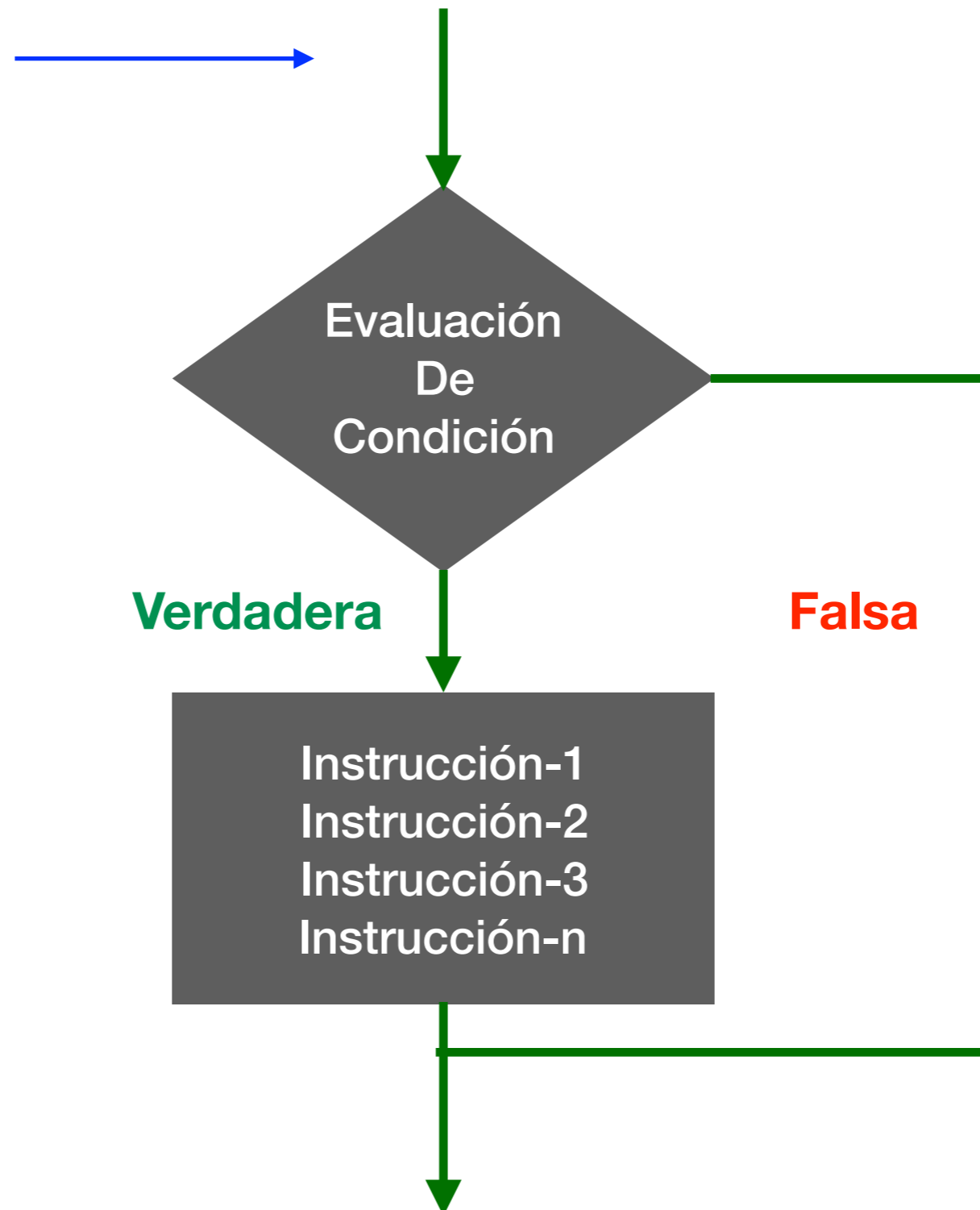
Inicio de la estructura de
control selección simple



Fin de la estructura de
control selección simple

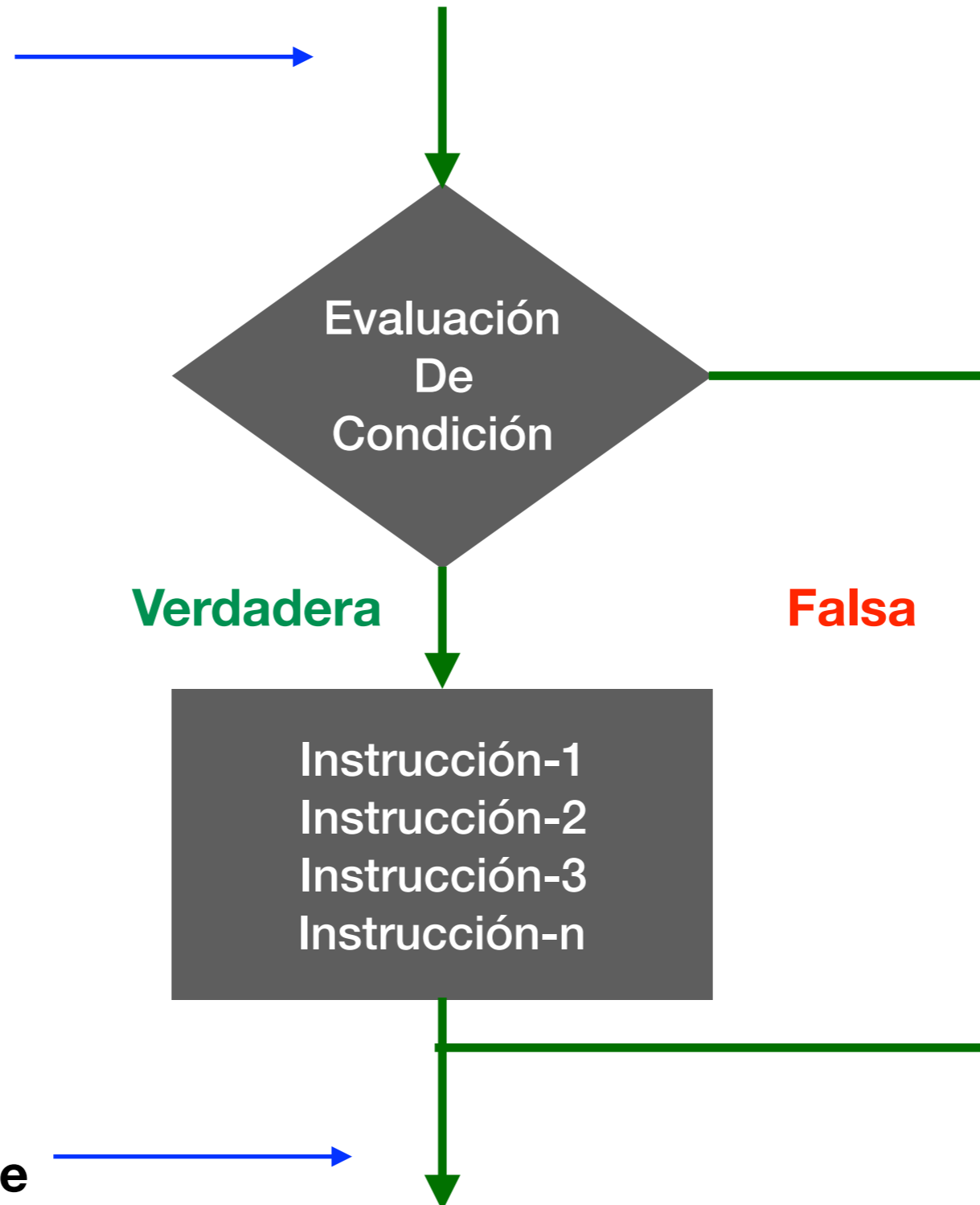


Inicio de la estructura de
control selección simple





Inicio de la estructura de
control selección simple



Fin de la estructura de
control selección simple



Universidad Autónoma del Estado de México
UAEM

Ejercicios



**Construir un algoritmo tal, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que esa calificación fuese mayor que 8.
Salidas: mensaje de aprobado si se cumple la condición.**



Construir un algoritmo tal, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que esa calificación fuese mayor que 8.

Salidas: mensaje de aprobado si se cumple la condición.

Entrada: calificación

Salida: imprimir "Aprobado"



**Construir un algoritmo tal, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que esa calificación fuese mayor que 8.
Salidas: mensaje de aprobado si se cumple la condición.**

Entrada: calificación

Salida: imprimir "Aprobado"

Inicio

Inicialización del algoritmo

Inicio





**Construir un algoritmo tal, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que esa calificación fuese mayor que 8.
Salidas: mensaje de aprobado si se cumple la condición.**

Entrada: calificación

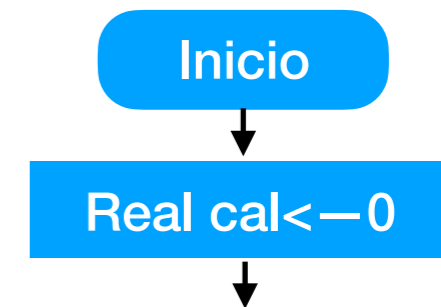
Salida: imprimir "Aprobado"

Inicio

Inicialización del algoritmo

Entero $cal \leftarrow 0$

Declaración de variables





Construir un algoritmo tal, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que esa calificación fuese mayor que 8.
Salidas: mensaje de aprobado si se cumple la condición.

Entrada: calificación

Salida: imprimir "Aprobado"

Inicio

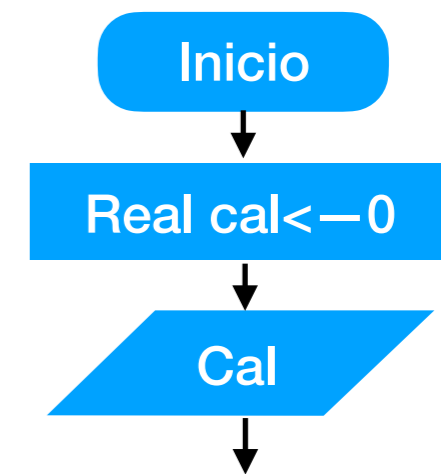
Inicialización del algoritmo

Entero $cal \leftarrow 0$

Declaración de variables

Leer cal

Lectura de datos





Construir un algoritmo tal, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que esa calificación fuese mayor que 8.
Salidas: mensaje de aprobado si se cumple la condición.

Entrada: calificación

Salida: imprimir "Aprobado"

Inicio

Entero $cal \leftarrow 0$

Leer cal

Si $(cal > 8)$ entonces
Imprimir "Aprobado"

Fin SI

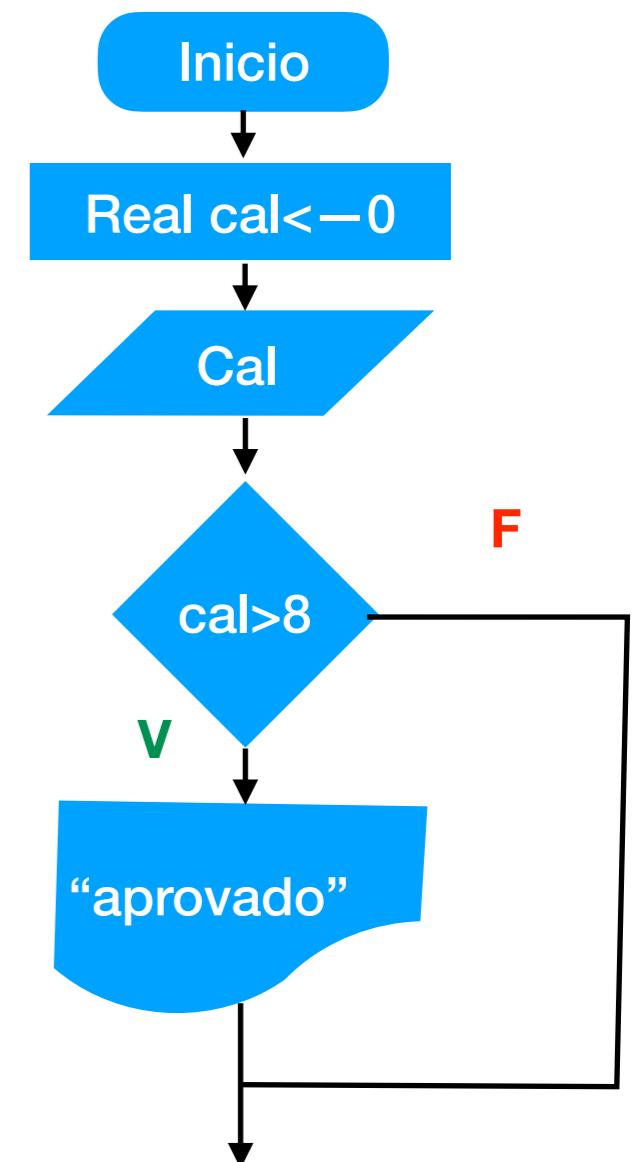
Inicialización del algoritmo

Declaración de variables

Lectura de datos

Inicio y evaluación
de la selección

Fin de la selección





Construir un algoritmo tal, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que esa calificación fuese mayor que 8.
Salidas: mensaje de aprobado si se cumple la condición.

Entrada: calificación

Salida: imprimir "Aprobado"

Inicio

Entero $cal \leftarrow 0$

Leer cal

Si $(cal > 8)$ entonces
Imprimir "Aprobado"

Fin SI

Fin

Inicialización del algoritmo

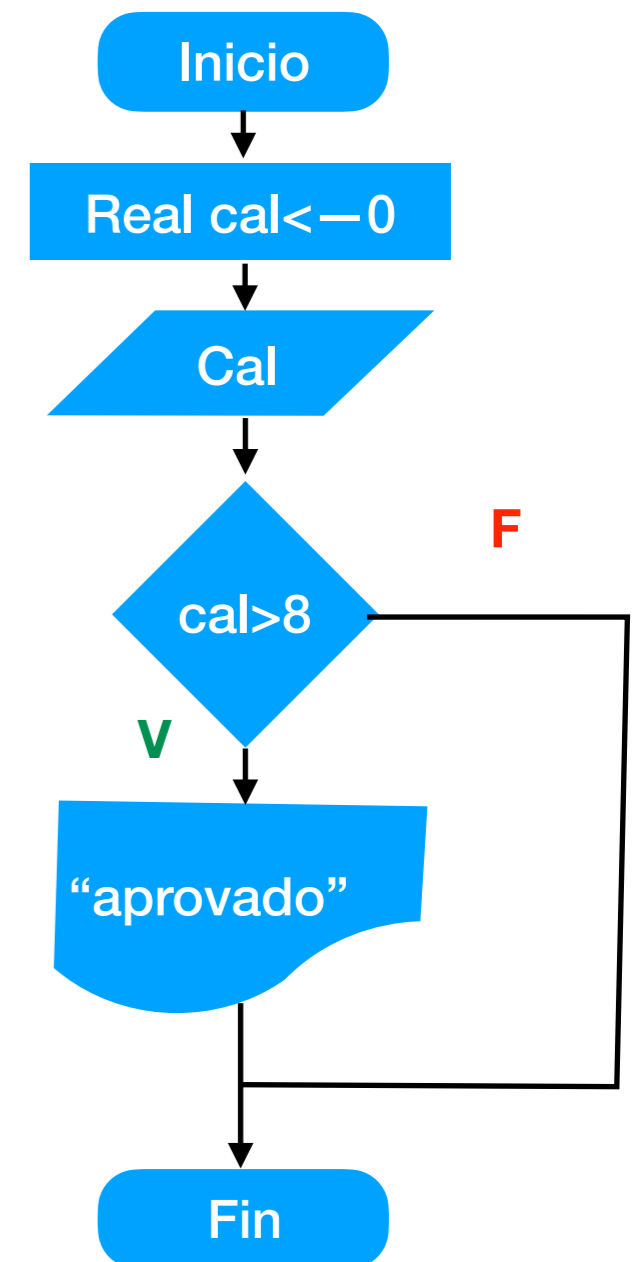
Declaración de variables

Lectura de datos

Inicio y evaluación
de la selección

Fin de la selección

Fin del algoritmo





Construir un algoritmo tal, que dado como dato un número real muestre en pantalla el numero introducido pero cambiando su signo solo en caso de que el numero no sea mayor o igual a cero



Construir un algoritmo tal, que dado como dato un número real muestre en pantalla el numero introducido pero cambiando su signo solo en caso de que el numero no sea mayor o igual a cero

Entrada: número real

Salida: imprimir - número real



Construir un algoritmo tal, que dado como dato un número real muestre en pantalla el numero introducido pero cambiando su signo solo en caso de que el numero no sea mayor o igual a cero

Entrada: número real

Salida: imprimir - número real

Inicio

Inicialización del algoritmo

Inicio





Construir un algoritmo tal, que dado como dato un número real muestre en pantalla el número introducido pero cambiando su signo solo en caso de que el número no sea mayor o igual a cero

Entrada: número real

Salida: imprimir - número real

Inicio

Real num < -0

Inicialización del algoritmo

Declaración de variables

Inicio

Real num < -0





Construir un algoritmo tal, que dado como dato un número real muestre en pantalla el numero introducido pero cambiando su signo solo en caso de que el numero no sea mayor o igual a cero

Entrada: número real

Salida: imprimir - número real

Inicio

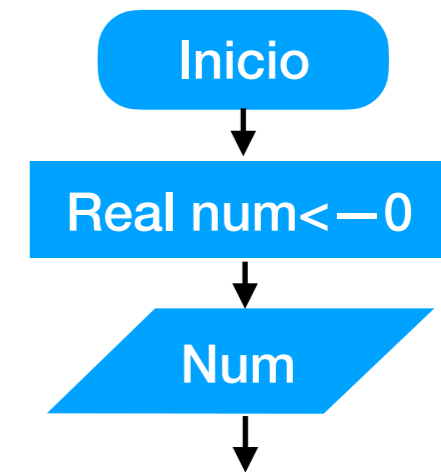
Real num < -0

Leer num

Inicialización del algoritmo

Declaración de variables

Lectura de datos





Construir un algoritmo tal, que dado como dato un número real muestre en pantalla el número introducido pero cambiando su signo solo en caso de que el número no sea mayor o igual a cero

Entrada: número real

Salida: imprimir - número real

Inicio

Real $num < -0$

Leer num

Si ($num < 0$) entonces

Imprimir - num

Fin SI

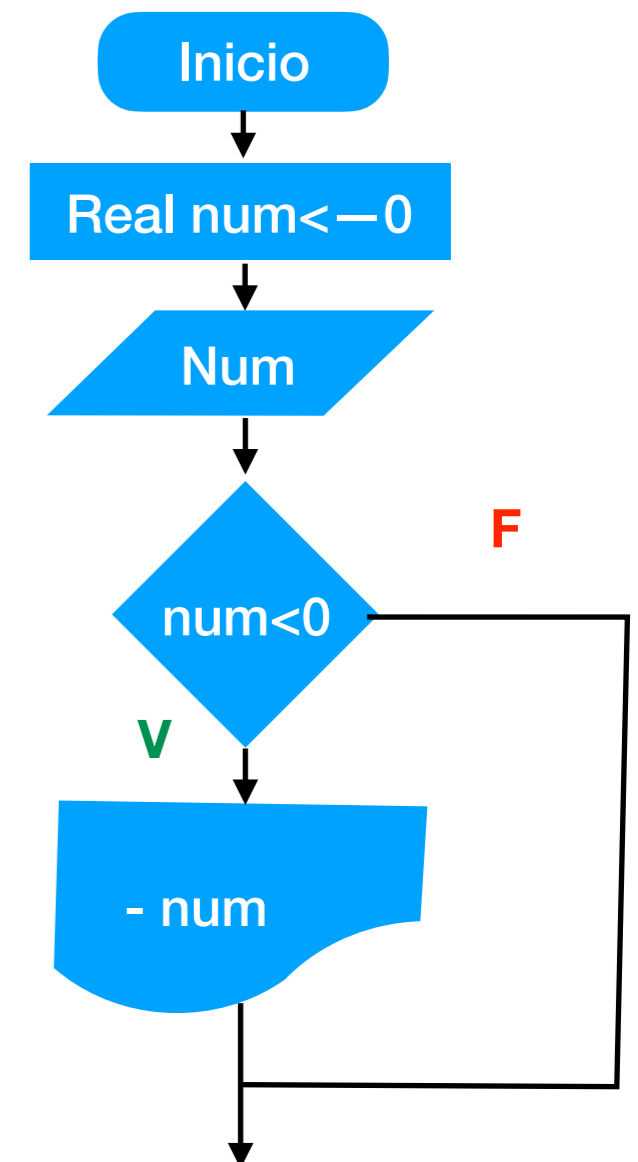
Inicialización del algoritmo

Declaración de variables

Lectura de datos

Inicio y evaluación
de la selección

Fin de la selección





Construir un algoritmo tal, que dado como dato un número real muestre en pantalla el número introducido pero cambiando su signo solo en caso de que el número no sea mayor o igual a cero

Entrada: número real

Salida: imprimir - número real

Inicio

Real $num < -0$

Leer num

Si ($num < 0$) entonces

Imprimir - num

Fin SI

Fin

Inicialización del algoritmo

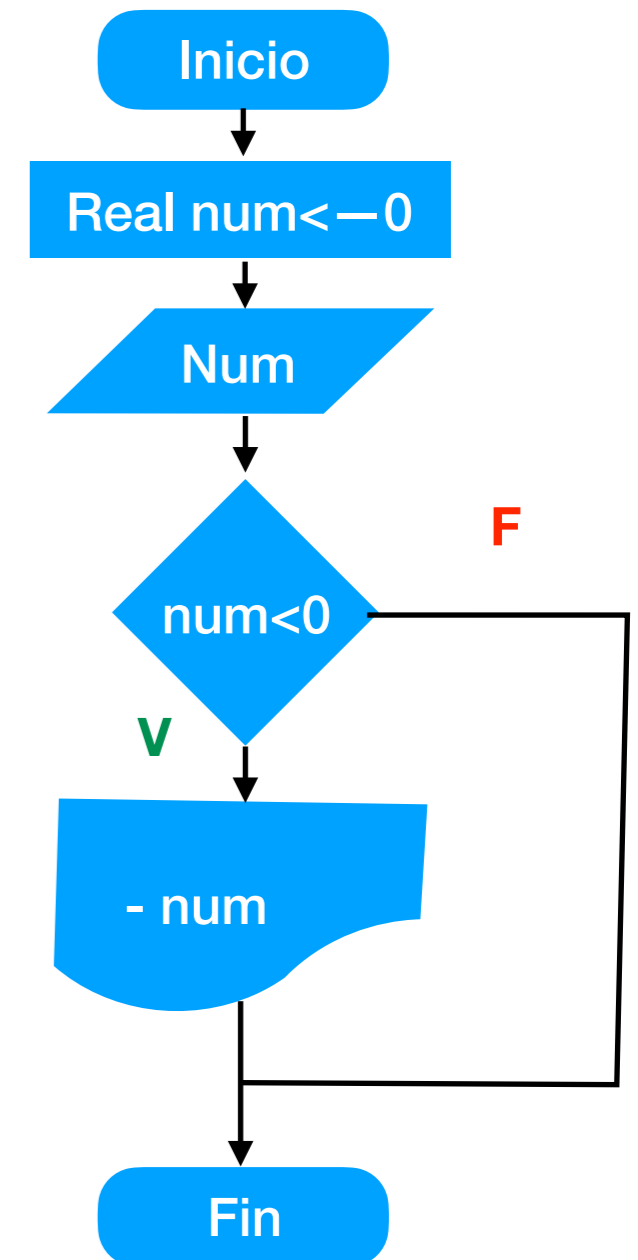
Declaración de variables

Lectura de datos

Inicio y evaluación
de la selección

Fin de la selección

Fin del algoritmo





Construir un algoritmo tal, que indique si un número se encuentra dentro de un rango de dos números.



Construir un algoritmo tal, que indique si un número se encuentra dentro de un rango de dos números.

Entrada: numero

Salida: imprimir en rango



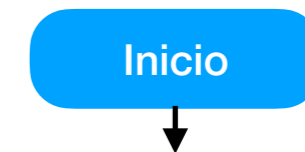
Construir un algoritmo tal, que indique si un número se encuentra dentro de un rango de dos números.

Entrada: numero

Salida: imprimir en rango

Inicio

Inicialización del algoritmo





Construir un algoritmo tal, que indique si un número se encuentra dentro de un rango de dos números.

Entrada: numero

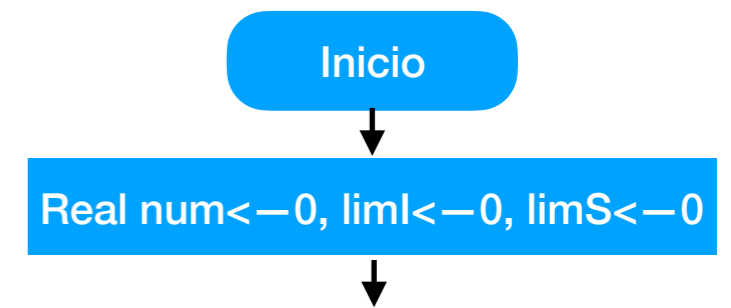
Salida: imprimir en rango

Inicio

Inicialización del algoritmo

Entero $num < -0$, $limI < -0$, $limS < -0$

Declaración de variables





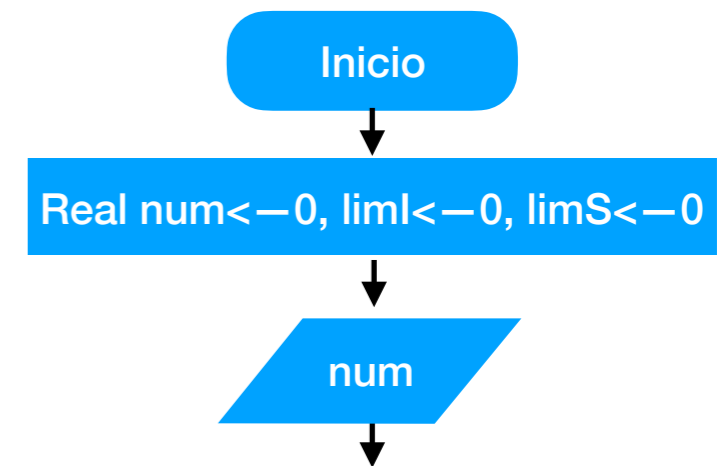
Construir un algoritmo tal, que indique si un número se encuentra dentro de un rango de dos números.

Entrada: numero

Salida: imprimir en rango

Inicio

Inicialización del algoritmo



Entero $num < -0$, $limI < -0$, $limS < -0$

Declaración de variables

Leer num

Lectura de datos



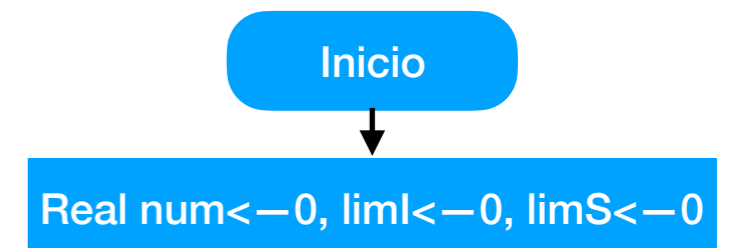
Construir un algoritmo tal, que indique si un número se encuentra dentro de un rango de dos números.

Entrada: numero

Salida: imprimir en rango

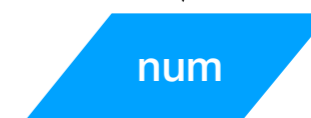
Inicio

Inicialización del algoritmo



Entero num<-0, limI<-0, limS<-0

Declaración de variables

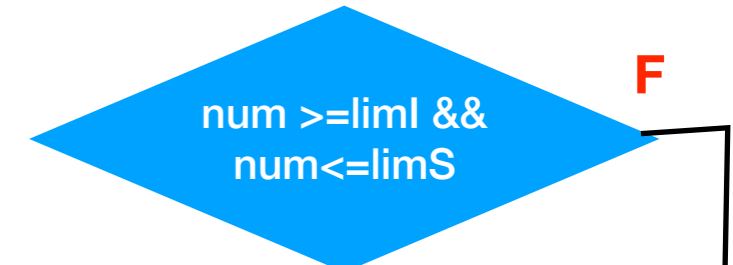


Leer num

Lectura de datos

**Si (num >=limI && num<=limS) entonces
Imprimir "en rango"
Fin Si**

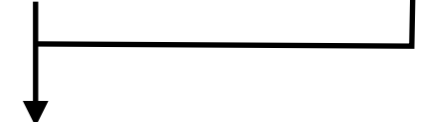
**Inicio y evaluación de la
selección**



Proceso



Fin de la selección





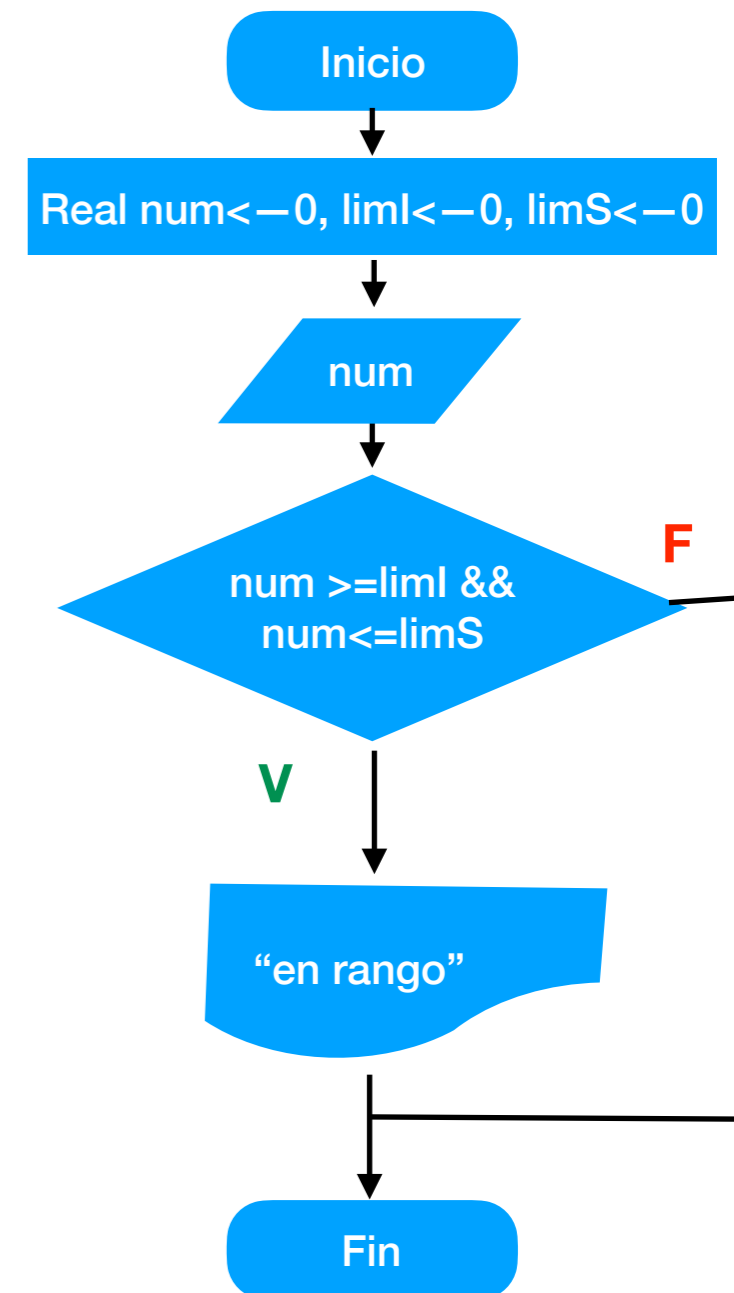
Construir un algoritmo tal, que indique si un número se encuentra dentro de un rango de dos números.

Entrada: numero

Salida: imprimir en rango

Inicio

Inicialización del algoritmo



Entero $num < 0$, $limI < 0$, $limS < 0$

Declaración de variables

Leer num

Lectura de datos

**Si $(num \geq limI \ \&\& \ num \leq limS)$ entonces
Imprimir "en rango"
Fin SI**

Inicio y evaluación de la selección

Proceso

Fin de la selección

Fin

Fin del algoritmo



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

ALGORITMOS Y PROGRAMACIÓN BÁSICA

Diapositivas

Material Didáctico sólo visión

Nombre del material:

Estructuras de control - Selección Simple

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 Estructuras de control selección simple.

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. ¿Qué son las estructuras de control ?
2. ¿Cuáles son las estructuras de control?
3. Estructura de control selección simple SI
4. Características de la estructura de control simple
5. Pseudocódigo
6. Sintaxis y estructura de la estructura de control decisión simple 1
7. Sintaxis y estructura de la estructura de control decisión simple 2
8. Sintaxis y estructura de la estructura de control decisión simple 3
9. Sintaxis y estructura de la estructura de control decisión simple 4
10. Sintaxis y estructura de la estructura de control decisión simple 5
11. Sintaxis y estructura de la estructura de control decisión simple 6
12. Diagrama de flujo 1
13. Diagrama de flujo 2
14. Diagrama de flujo 3
15. Diagrama de flujo 4
16. Diagrama de flujo 5
17. Ejercicios
18. Ejercicio 1
19. Ejercicio 2
20. Ejercicio 3
21. Bibliografía