

COMPUTACIÓN I

CLASE DE TEORÍA 1

(CI-2125, CI-2127)

Profesor

Darwin Rocha
(DarwinRocha@usb.ve)

VALORES ÉTICOS Y MORALES

(ESPECIALMENTE A LA HORA DE PRESENTAR EVALUACIONES)

- **Honestidad**: se refiere a los atributos positivos y virtuosos tales como la integridad, veracidad y sinceridad
- **Laboriosidad**: Es el gusto por trabajar y esforzarse en conseguir objetivos sin rendirse.
- **Responsabilidad**: Un deber u obligación de realizar satisfactoriamente o completar una actividad que hay que cumplir
- **Paz**: Es un estado de tranquilidad que se caracteriza por la no permanencia de conflictos violentos y la facilidad de no tener temor a la violencia.

“Amaras a tu prójimo como a ti mismo”

Jesús de Nazaret

ACLARATORIA SOBRE EL CURSO

Este curso es de PROGRAMACIÓN por tanto se espera que el estudiante tenga conocimientos sobre el uso del computador.

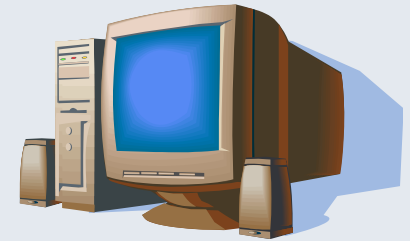
NO veremos cosas como:

- Manejo de Office, LibreOffice u openOffice (ofimática)
- Manejo de cables de redes o conexión a internet (redes de computadores)
- Uso de redes sociales u otras aplicaciones (Uso del internet)

Entonces,

¿QUÉ VEREMOS?

¿QUÉ ES UN COMPUTADOR?



¿QUÉ ES UN COMPUTADOR?

Es una maquina electrónica que se **PROGRAMA** para que ejecute tareas.

Programar correctamente, implica saber como funciona un computador, como desarrollar algoritmos y como codificarlo en un lenguaje cualquiera de programación.

¿CÓMO FUNCIONA UN COMPUTADOR?

(ARQUITECTURA DEL COMPUTADOR)

- Recibir entradas de información del mundo exterior (teclado, mouse, micrófono, internet, etc.)
- Producir salidas de información al mundo exterior (monitor, impresora, internet, etc.)
- Procesar información: aplica operaciones aritméticas o lógicas.
- Almacenar información: los datos reposan en algún dispositivo (pendrive, disco duro, cd, etc)

MODELO ABSTRACTO DE OPERACIONES

En este ejemplo:

Entradas: varios datos y un tipo de operación

Salidas: varios datos.

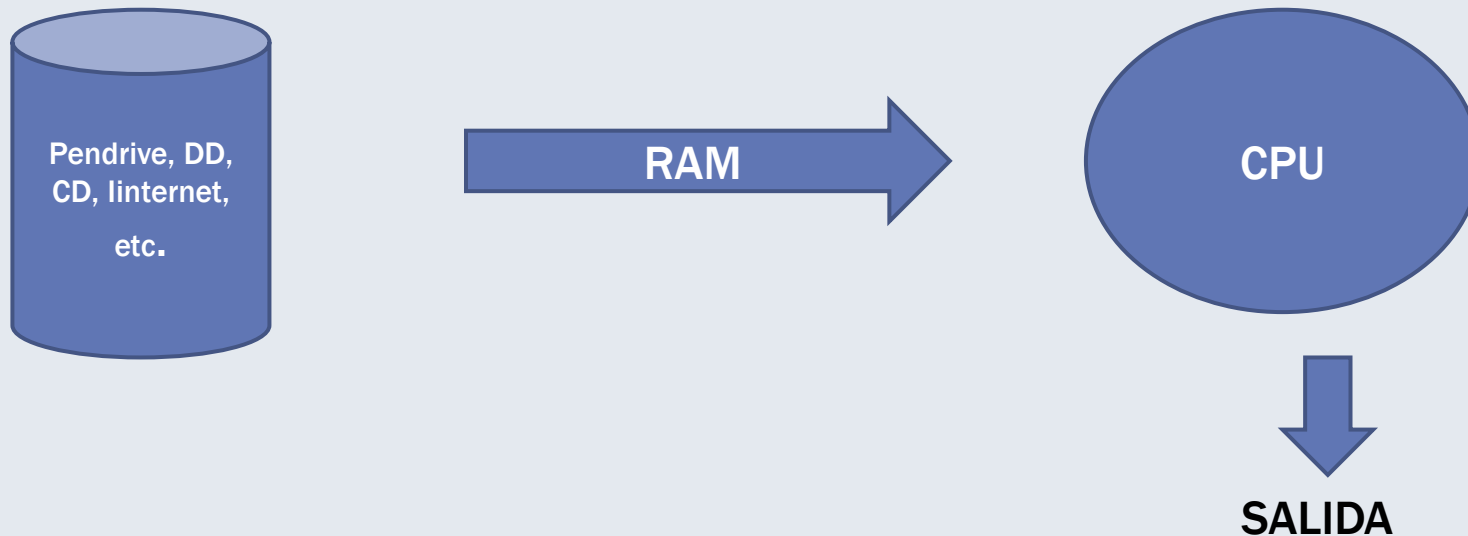


¿CÓMO FUNCIONA UN COMPUTADOR?

(ARQUITECTURA DEL COMPUTADOR)

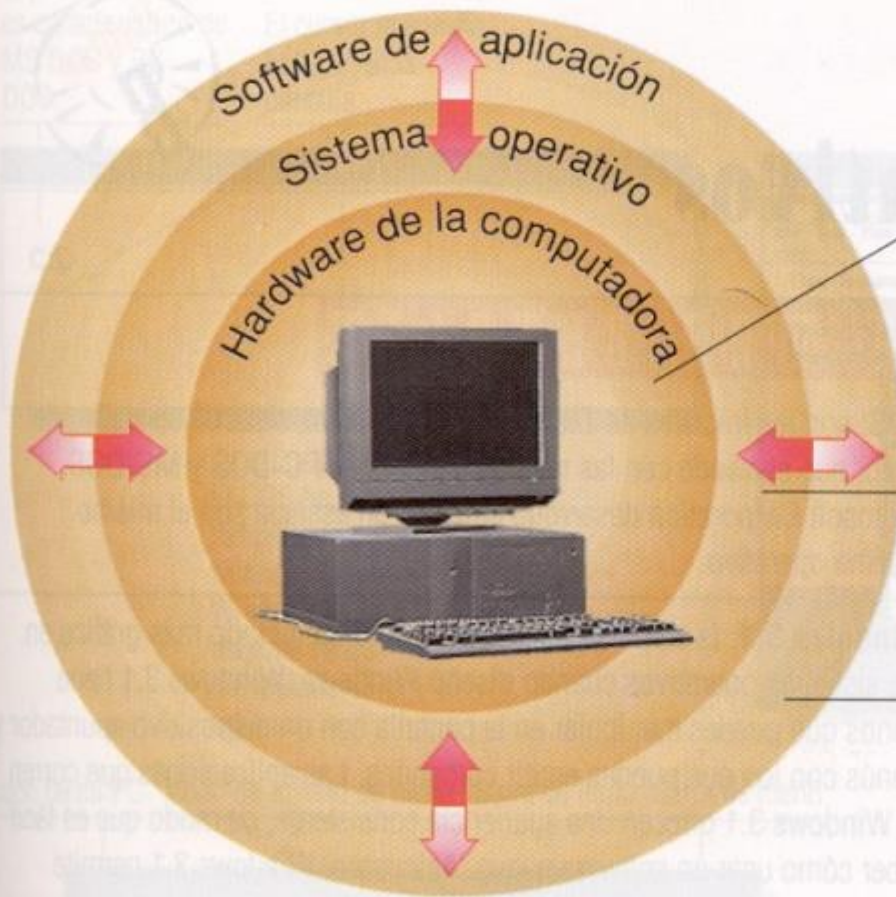
Todo dato a ejecutarse debe estar almacenado en memoria principal (RAM)

Toda ejecución se realiza en la unidad de procesamiento (CPU)



¿CÓMO FUNCIONA UN COMPUTADOR?

(ARQUITECTURA DEL COMPUTADOR)



El **hardware de la computadora** es el núcleo del sistema, pero no puede funcionar sin un sistema operativo.

El **sistema operativo** sirve de enlace entre el hardware de la computadora y el software de aplicación.

El **software de aplicación** necesita que el sistema operativo realice tareas relacionadas con el hardware, como impresión de informes y almacenamiento de datos en discos.

CÓMO DESARROLLAR ALGORITMOS

¿Qué es un algoritmo?

Informalmente:

Serie de pasos finitos y lógicos que me permiten realizar una tarea.

Por ejemplo para bañarse:

1. Quitarse la ropa.
2. Entrar a la ducha
3. Enjabonarse

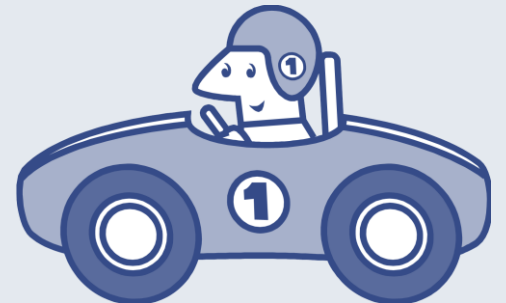


4. quitarse el jabón.
5. Salir de la ducha
6. Secarse y vestirse

CÓMO DESARROLLAR ALGORITMOS

¿Cómo cambiar un neumático o caucho de un automóvil?

(en su forma general)



CÓMO DESARROLLAR ALGORITMOS

¿Cómo cambiar un neumático o caucho de un automóvil?

(en su forma general)

1. Buscar las herramientas
2. Colocar el gato
3. Aflojar las tuercas
4. Retirar el neumático
5. Colocar un nuevo neumático
6. Apretar las tuercas
7. Quitar el gato
8. Guardar las herramientas



CÓMO DESARROLLAR ALGORITMOS

Definición formal:

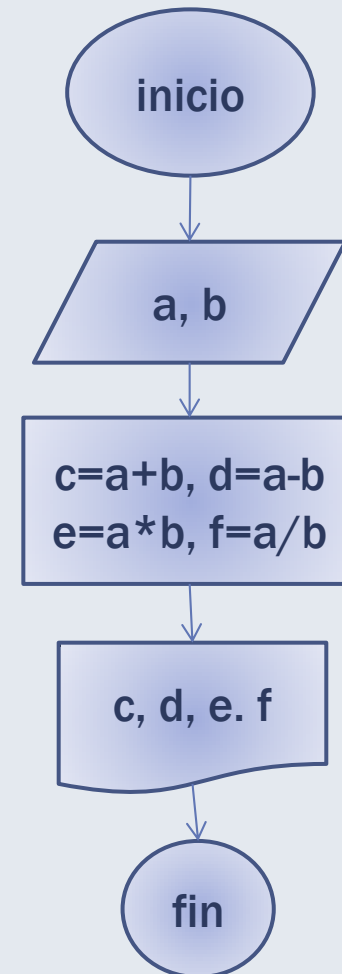
Secuencia ordenada de instrucciones muy bien definidas, precisas y efectivas, que al ser ejecutadas, siempre generan un resultado en tiempo finito.

CÓMO DESARROLLAR ALGORITMOS

EJEMPLO 1.A

Leer dos números (a, b) y aplicar sobre ellos las operaciones básicas.

- a) Iniciar
- b) $a = 7$ y $b = 9$
- c) $c = 7 + 9$
- d) $d = 7 - 9$
- e) $e = 7 * 9$
- f) $f = 7 / 9$
- g) 16.0, -2.0, 63.0, 0.779
- h) Finalizar



CÓMO DESARROLLAR ALGORITMOS

Precondición: indica las condiciones que deben satisfacer los datos de entrada para que el programa pueda cumplir su tarea.

Si X es el argumento de una raíz, X debe positivo

Postcondición: indican las condiciones de salida que son aceptables como soluciones correctas del problema en cuestión.

Si Y es igual a X al cuadrado, Y siempre será positivo

CÓMO DESARROLLAR ALGORITMOS

- **Precondición:** describe todas las posibles entradas del algoritmo o un segmento de este.

En el ejemplo 1.A: los números a y b son naturales (o enteros positivos sin incluir el cero)

- **Postcondición:** describe todas las posibles salidas

En el ejemplo 1.A *son todas números reales*

CÓMO DESARROLLAR ALGORITMOS

NOTA: si cambiamos las precondiciones o las postcondiciones, sería equivalente a cambiar el ejercicio.

- Precondición: describe todas las posibles entradas del algoritmo o un segmento de este.

En el ejemplo 1.B: los números a y b son enteros,

- Postcondición: describe todas las posibles salidas

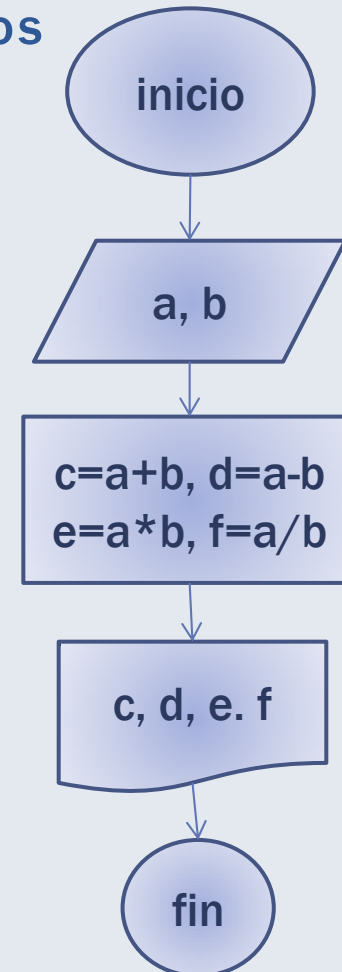
En el ejemplo 1.B son *todas* números enteros

CÓMO DESARROLLAR ALGORITMOS

EJEMPLO 1.B

Tomar dos números (a, b) y aplicar sobre ellos las operaciones básicas.

- a) Iniciar
- b) $a = 7$ y $b = -9$
- c) $c = 7 + (-9)$
- d) $d = 7 - (-9)$
- e) $e = 7 * (-9)$
- f) $f = 7 / -9$
- g) -2, 16, -63, 0
- h) Finalizar



FORMAS DE REPRESENTACIÓN DE ALGORITMOS

- ✓ Lenguaje Natural
- ✓ Diagramas de flujo
- ✓ Lenguaje Algorítmico o pseudocódigo

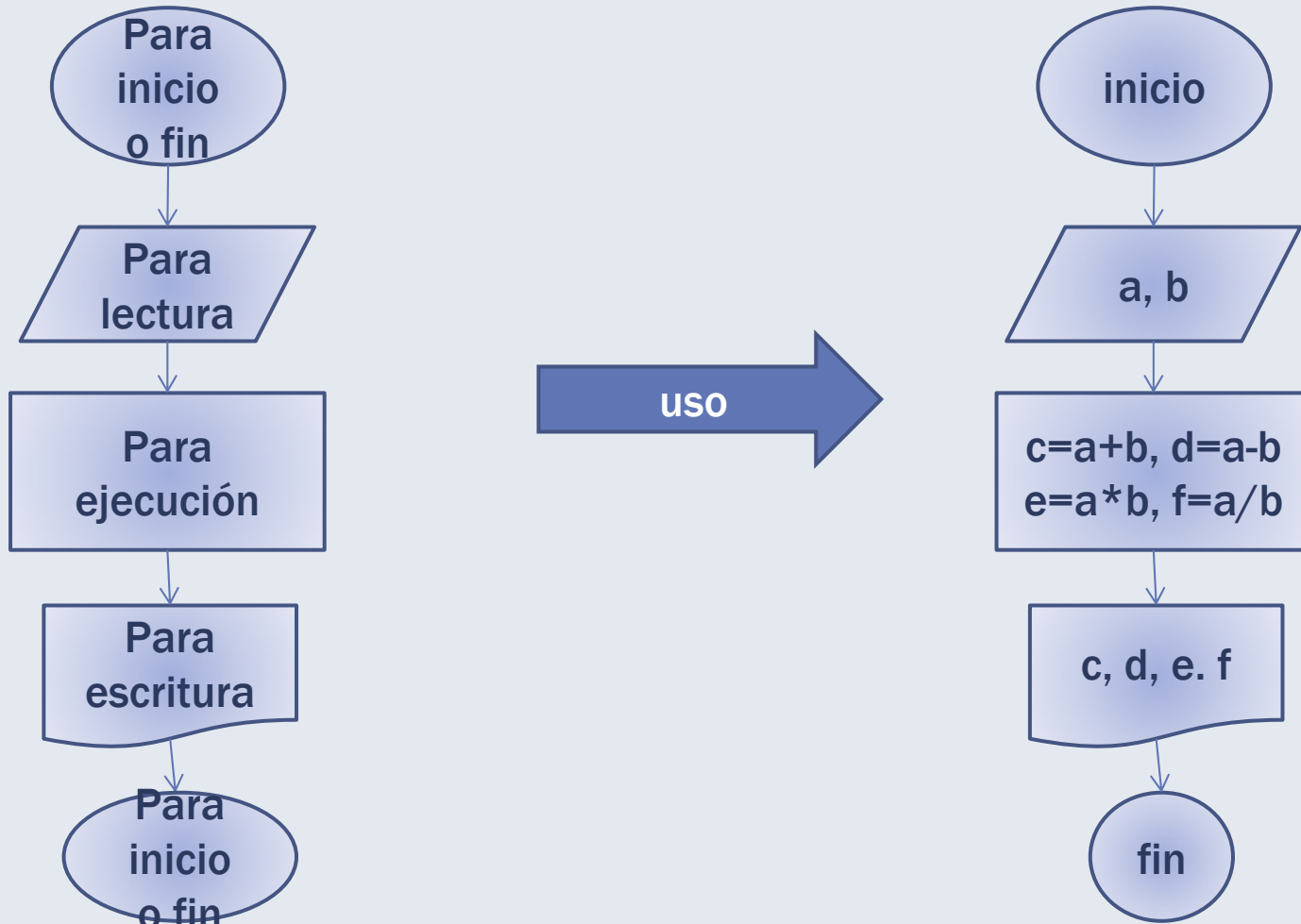
```
Inicio
{
  Enteros: a,b,c,d,e,f;
  Leer a
  Leer b
  c ← a + b
  d ← a - b
  e ← a * b
  f ← a / b
  Escribir c, d, e, f;
}
Fin
```



DIAGRAMAS DE FLUJO

- Es una forma de representación algorítmica que usa una esquematización gráfica.
- Muestra gráficamente los pasos a seguir para alcanzar la solución al problema.
- Todo esquema de representación algorítmica es y debe ser independiente del lenguaje.
- El diagrama de flujo representa la solución del problema. El programa representa la implementación.

DIAGRAMAS DE FLUJO



DIAGRAMAS DE FLUJO

(REGLAS DE CONSTRUCCIÓN)

- Todo diagrama de flujo debe tener inicio y fin.
- Las líneas utilizadas para indicar la dirección del flujo del diagrama, deben ser rectas: verticales u horizontales.
- Todas las líneas utilizadas para indicar dirección del diagrama, deben estar conectadas.
- Debe construirse de arriba hacia abajo y de izquierda a derecha.
- La notación utilizada es y debe ser independiente del lenguaje de programación.

DIAGRAMAS DE FLUJO

(REGLAS DE CONSTRUCCIÓN)

- Si la construcción del diagrama de flujo requiere más de una hoja se deben utilizar los conectores adecuados y enumerar las páginas correspondientes.
- Al resolver un problema complejo es conveniente colocar comentarios que expresen o ayuden a entender la solución.
- No puede llegar más de una línea a un símbolo determinado.

PROGRAMACIÓN VS CODIFICACIÓN

❖ **Programación:** Se refiere a todos los pasos involucrados en resolver un problema de computación.



❖ **Codificación:** Se define como el proceso de escribir instrucciones en algún lenguaje de programación existente.



PROGRAMACIÓN VS CODIFICACIÓN

La Programación Involucra:

- ✓ Definir lo que se quiere hacer.
- ✓ Clarificar las ambigüedades del enunciado del problema.
- ✓ Decidir como resolver el problema
- ✓ Esquematizar la solución



En la Codificación se debe aplicar:

- ✓ La detección de errores.
- ✓ Documentación, pruebas y mantenimiento.



¿PREGUNTAS?

Cubículo: MYS-212C (previo acuerdo por correo)

Correo: darwinrocha@usb.ve

También pueden hacer consulta a través de los foros de moodle.

«... y esta es la vida eterna que te conozcan a ti, el único Dios verdadero y Jesucristo a quien tu has enviado » juan 17:3

BIBLIOLOGÍA

- Notas de clases de los profesores: Mireya Morales, Carmen R. Rodríguez y Miguel Torrealba
- Fundamentos de programación, piensa en C. Profesor Osvaldo Cairo, Pearson Educación, Primera edición 2006.