

INTRODUCCIÓN A LA INFORMÁTICA

Guía resumen Módulo II

Tabla de contenido

1.	Estructura funcional de un computador	4
1.1	Definición	4
1.2	Unidad central de procesamiento (CPU)	4
1.3	Unidad Aritmética-Lógica (ALU)	5
1.4	Unidad de Control (UC)	5
1.5	Memoria principal (RAM)	5
1.6	Memoria principal (ROM)	5
1.7	Acumulador o pila del ordenador	6
1.8	CMOS BIOS	6
1.9	Modelo básico de la arquitectura de Von Neumann	6
2.	Dispositivos de entrada y salida	7
2.1	Unidades periféricas	7
2.2	Unidades de entrada	7
3	Hardware y Software	8
3.1	hardware	8
3.2	Software	8
4	Dato e Información	8
4.1	Dato cuantitativo	8
4.2	Dato cualitativo	8
4.3	Información	9
5	Algoritmo	10
5.1	Algoritmos cualitativos	10
5.2	Algoritmos cuantitativos	10
5.3	Algoritmos de ordenamiento	10
5.4	Algoritmos de búsqueda	10
6	Pseudocódigo	11
6.1	Análisis	11
6.2	Diseño	11
6.3	Implementación	11
7	Diseño Top – Down	12
8	Diagrama de flujo	13

ANALISTA DESARROLLADOR DE APLICACION DE SOFTWARE

9	Estructuras algorítmicas.....	14
9.1	La estructura secuencial.....	14
9.2	Estructuras de selección	16
9.2.1	Si (selección).....	16
9.2.2	Si...Sino (selección doble).....	16
9.2.3	Según (selección múltiple)	16
9.3	Estructuras iterativas	17
9.3.1	Mientras	17
9.3.2	Hacer	17

INTRODUCCIÓN A LA INFORMÁTICA

Módulo II

1. Estructura funcional de un computador

Un computador es una máquina que está diseñada para facilitarnos la vida. En muchos países se le conoce como computadora u ordenador, pero todas estas palabras se refieren a lo mismo.

Esta máquina electrónica nos permite desarrollar fácilmente múltiples tareas que ahora hacen parte de nuestra vida cotidiana, como elaborar cartas o una hoja de vida, hablar con personas de otros países, hacer presupuestos, jugar y hasta navegar en internet.

Nuestro computador hace esto procesando datos para convertirlos en información útil para nosotros.

1.1 Definición

El funcionamiento estructural de una computadora se basa en:

- La **captura de datos** que se van a procesar por medio de alguna unidad de entrada.
- Su **almacenamiento** en la unidad central de procesamiento.
- En la **ejecución** de un programa que transforma los datos de entrada en resultados.
- En la **comunicación** de esos resultados (información) al exterior, por medio de una unidad de salida.

1.2 Unidad central de procesamiento (CPU)

CPU por sus siglas del inglés:

Central Process Unit es la encargada de realizar todas las operaciones de proceso de datos y control del ordenador.

Está formada por dos unidades;

- Unidad Aritmetica Lógica (ALU)
- Unidad de Control. (Unit Control)

1.3 Unidad Aritmética-Lógica (ALU)

Es la encargada de realizar los cálculos aritméticos y lógicos tales como sumar, restar, multiplicar etc y operaciones lógicas como si, y, o, no entre dos valores.

1.4 Unidad de Control (UC)

Es el cerebro del ordenador, que dirige todas las operaciones de procesado, sus funciones son:

- Localizar cada instrucción a ejecutar e interpretarla.
- Distribuir las órdenes oportunas al resto de dispositivos para ejecutar la instrucción.
- Controlar la secuencia de ejecución de las instrucciones.
- Establecer la comunicación entre el resto de dispositivos

1.5 Memoria principal (RAM)

Conocida como memoria de acceso aleatorio **RAM**, por sus siglas del inglés: *Random-Access Memory*.

Se utiliza para almacenar los datos y los programas que están en uso, es uno de los principales factores que marca el rendimiento del PC.

Se distinguen dos tipos de memoria

- **RAM** (volátil)
- **ROM** (permanente)

1.6 Memoria principal (ROM)

Conocida como (**Read Only Memory**) o memoria de sólo lectura, es la que almacena información básica del equipo, sin ella, no es posible determinar qué elementos básicos integra al momento del arranque.

Su contenido no puede ser modificado, contiene “rutinas” o mejor dicho, pequeños programas del sistema, es un medio de almacenamiento utilizado en ordenadores y dispositivos electrónicos, que permite solamente la lectura de la información.

Se encuentran tres pequeños programas:

- El **POST** es realizar una prueba inicial del Hardware crítico del sistema.
- El **SETUP** es donde podemos entrar en su configuración y realizar cambios en sus parámetros

1.7 Acumulador o pila del ordenador

Los datos que hay en esta memoria nunca se perderán aun cuando se apague el ordenador.

¿Cómo se consigue que no se pierdan al apagarse?

Mediante el acumulador o pila del ordenador.

1.8 CMOS BIOS

El BIOS ha sustituido a la antigua memoria ROM, en este caso algunos datos se pueden modificar por si el usuario quiere ampliar su ordenador

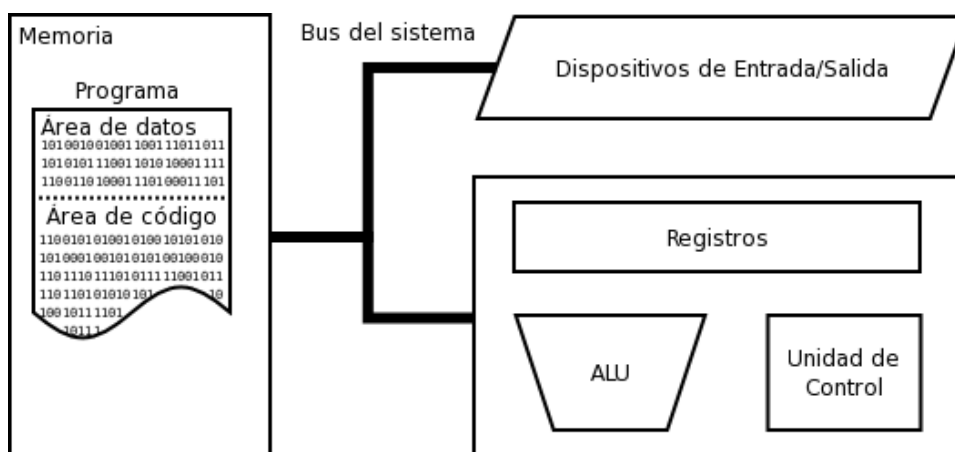
El BIOS (o la bios) de una PC es una **memoria ROM**, pero con la facultad de configurarse según las características particulares de cada máquina (hay datos que se pueden modificar, por ejemplo añadirle un disco duro).

El BIOS ha sustituido a la antigua memoria ROM, en este caso algunos datos se pueden modificar por si el usuario quiere ampliar su ordenador

El BIOS (o la bios) de una PC es una **memoria ROM**, pero con la facultad de configurarse según las características particulares de cada máquina (hay datos que se pueden modificar, por ejemplo añadirle un disco duro).

1.9 Modelo básico de la arquitectura de Von Neumann

Unidad central de procesamiento



2. Dispositivos de entrada y salida

2.1 Unidades periféricas

En informática, **periférico** es la denominación genérica para designar al aparato o dispositivo auxiliar e independiente conectado a la unidad central de procesamiento de una computadora.

Se consideran periféricos a las unidades o dispositivos de hardware a través de los cuales la computadora se comunica con el exterior, y también a los sistemas que almacenan o archivan la información.

Las unidades periféricas se clasifican en:

- Unidades de entrada
- Unidades de salida
- Unidades de almacenamiento

2.2 Unidades de entrada

Son los que permiten introducir datos externos a la computadora para su posterior tratamiento por parte de la CPU. Estos datos pueden provenir de distintas fuentes, siendo la principal un ser humano.

Los periféricos de entrada más habituales son:

- Teclado (*keyboard*)
- Ratón (*mouse*)
- Panel táctil (*touchpad*)
- Escáner (*scanner*)

3 Hardware y Software

3.1 hardware

El **hardware**, es toda la parte física del computador que se puede ver, es decir todos los componentes de su estructura física, la pantalla, el teclado, la torre y el mouse hacen parte del hardware del equipo.

3.2 Software

El **software**, son todos los programas informáticos que hacen posible la realización de tareas específicas dentro de un computador. Por ejemplo Word, Excel, PowerPoint, los navegadores web, los juegos, sistemas operativos, etc.

4 Dato e Información

Si bien ambos conceptos se relacionan deben distinguirse conceptualmente uno del otro. Es el componente **mínimo** de una información mayor necesaria para llegar al conocimiento exacto de una cosa o hecho que se describe en objetos, situaciones, etc. Para ello dos tipos de datos:

4.1 Dato cuantitativo

Son aquellos que se pueden contar o medir. Expresan mediante números las propiedades de un objeto, hecho o persona.

Ejemplo: Edad, peso, fecha de transacción.

4.2 Dato cualitativo

No se pueden contar, ni medir. Expresan nominalmente las características o propiedades de un objeto, hecho o persona.

Ejemplo: sexo, descripción de un artículo, nombre.

“un dato se define como una representación simbólica que por si solo no transmite ningún mensaje ni tampoco permiten tomar decisiones”.

4.3 Información

Es el conjunto de datos procesados en forma significativa, ordenados y con una secuencia lógica sobre algún suceso o hecho de importancia, este tiene un valor real para tomar decisiones.

A medida que tenemos más información, más fácil nos resulta tomar decisiones correctas, incrementando la probabilidad del éxito.

- Conjunto de Datos procesados
- Conjunto de Datos organizados
- Tienen un significado
- Transmiten un mensaje
- Permite la toma de decisiones
- Favorece a la resolución de problemas
- Incrementa el conocimiento

5 Algoritmo

Es una serie de normas o leyes específicas que hacen posible la ejecución de actividades, cumpliendo una serie de pasos continuos que no de origen a dudas a la persona que realice dicha actividad.

Los algoritmos se pueden expresar de diversas formas tales como el lenguaje natural, el lenguaje de programación, pseudocódigo y diagramas de flujo.

5.1 Algoritmos cualitativos

Son aquellos que se realizan por medio de las palabras, lo que quiere decir que las órdenes vienen dadas en forma verbal, por ejemplo una receta de cocina

5.2 Algoritmos cuantitativos

Son aquellos que se realizan por medio de cálculos matemáticos, por ejemplo si se desea saber cuál es la raíz cuadrada de un número, se pueden aplicar algoritmos para ello.

5.3 Algoritmos de ordenamiento

Son aquellos que llevan en orden los elementos que ingresan, dependiendo del orden numérico o léxico.

5.4 Algoritmos de búsqueda

Son aquellos que tratan de encontrar dentro de la lista que ingresa, algún elemento en especial que cumpla con las órdenes dadas.

6 Pseudocódigo

Es un **lenguaje artificial e informal** que ayuda a los programadores a desarrollar algoritmo.

El **pseudocódigo** es similar al lenguaje cotidiano, es cómodo y amable con el usuario, aunque no es realmente un verdadero lenguaje de programación. Un programa escrito en pseudocódigo puede ser fácilmente convertido en un programa Java, si es que está bien elaborado.

6.1 Análisis

- Formalizar el entorno del problema
- Definir lo que debe hacer el programación
- Estudio de posibilidades y casos
- Determinar los actores de intervienen
-

6.2 Diseño

- Definir los elementos que intervienen
- Identificación de los datos de entrada y salida
- Identificación de recursos a utilizar
- Definición de las acciones para resolver el problema
- Se utilizan técnicas de diseño descendente (top-down), programación modular y estructurada

6.3 Implementación

- Transformar los elementos y acciones en una secuencia ordenada de instrucciones y sentencias que definen el algoritmo.

7 Diseño Top – Down

Consiste en establecer una serie de niveles de **mayor a menor complejidad** que den solución al problema.

Consiste en efectuar una relación entre las etapas de la estructuración de forma que una etapa jerárquica y su inmediato inferior se relacionen mediante entradas y salidas de información.

Este diseño consiste en una serie de descomposiciones sucesivas del problema inicial, que recibe el refinamiento progresivo del repertorio de instrucciones que van a formar parte del programa.

La utilización de la técnica de diseño top -down tiene los siguientes objetivos básicos:




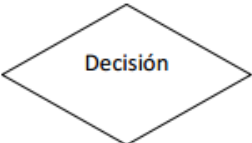
- Simplificación del problema y de los **subprogramas** de cada descomposición.
- Las diferentes partes del problema pueden ser **programadas** de modo independiente e incluso por diferentes personas.
- El programa final queda estructurado en forma de **bloque o módulos** lo que hace mas sencilla su lectura y mantenimiento.




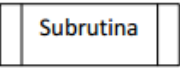
8 Diagrama de flujo

Un diagrama de flujo es la **representación gráfica de un algoritmo**.

También se puede decir que es la representación detallada en forma gráfica de cómo deben realizarse los pasos en la computadora para producir resultados.

Esta representación gráfica se da cuando varios símbolos (que indican diferentes procesos en la computadora), se relacionan entre sí mediante líneas que indican el orden en que se deben ejecutar los procesos. Los símbolos utilizados han sido normalizados por el instituto norteamericano de normalización (**ANSI**).

Símbolo	Descripción
	Terminal. Representa el inicio y el final de un algoritmo.
	Entrada y Salida (E/S). Representa la lectura de datos desde el dispositivo de entrada estándar, así como la impresión de datos en el dispositivo de salida estándar.
	Proceso. Representa cualquier tipo de operación que pueda originar un cambio de la información almacenada en memoria, asignaciones u operaciones aritméticas.
	Decisión. Nos permite analizar una situación, con base en los valores verdadero y falso. Toma una decisión de las instrucciones que a continuación ejecuta el algoritmo.

	Conector. Sirve para enlazar dos partes cualesquiera del diagrama que están en la misma página.
	Línea de flujo. Indica el orden de la ejecución de las operaciones. La flecha indica cuál es la siguiente instrucción que se debe realizar.
	Conector. Conecta a dos puntos del diagrama cuando éstos se encuentran en páginas diferentes. Representa el inicio y el final de un programa.
	Llamada a subrutina. Llama a un proceso determinado o subrutina. Una subrutina es un módulo independiente del módulo principal, que realiza una tarea determinada y al finalizar regresa el control de flujo al módulo principal.

9 Estructuras algorítmicas

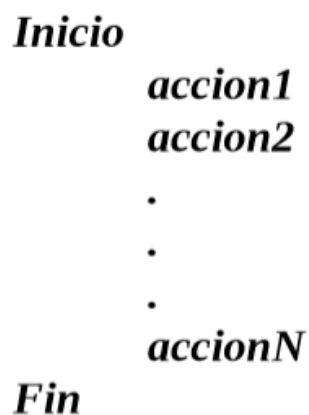
Las estructuras de operación de programas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas.

Estas estructuras se clasifican de acuerdo con su complejidad en:

ESTRUCTURAS ALGORITMICAS		
SECUENCIALES	CONDICIONALES	CICLICAS
ASIGNACION	SIMPLES	HACER PARA
ENTRADA	MULTIPLES	HACER MIENTRAS
SALIDA		REPETIR HASTA

9.1 La estructura secuencial

Es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso. Una estructura secuencial se representa de la siguiente forma:



ANALISTA DESARROLLADOR DE APLICACION DE SOFTWARE

ASIGNACION	Consiste en el paso de valores o resultados a una zona de memoria. Dicha zona será reconocida con el nombre de variable que recibe el valor. La asignación se puede clasificar de la siguiente forma	
SIMPLES	Consiste en pasar un valor constante a una variable.	$a = 15$
CONTADOR	Consiste en usarla como un verificador del número de veces que se realiza un proceso.	$a = a + 1$
ACUMULADOR	Consiste en usarla como un sumador en un proceso.	$a = a + b$
DE TRABAJO	Donde se puede recibir el resultado de una operación matemática que involucre muchas variables	
LECTURA	La lectura consiste en recibir desde un dispositivo de entrada. (p/e el teclado) un valor. Esta operación se representa en un pseudocódigo como sigue: Leer a, b Donde "a" y "b" son las variables que recibirán los valores	
ESCRITURA	Consiste en mandar por un dispositivo de salida. (p/e pantalla) un resultado o mensaje. Esta proceso se representa en un pseudocódigo como sigue: Escribe "El resultado es: ", R Donde "El resultado es: " es un mensaje que se desea aparezca y R es una variable que contiene un valor.	

9.2 Estructuras de selección

9.2.1 Si (selección)

En esta estructura de selección, las instrucciones dependen de una condición para ejecutarse. El algoritmo analiza el resultado de dicha condición booleana (se cumple o no) y realiza la acción correspondiente a ese resultado. Si no se cumple la condición no hace nada.

**SI (condición) ENTONCES
INSTRUCCIONES
FIN DEL SI**

9.2.2 Si...Sino (selección doble)

Esta estructura funciona de la misma manera que la selección tradicional, solo que en caso de no cumplirse la condición realiza otra acción, correspondiente a la instrucción2.

**SI (condición) ENTONCES
INSTRUCCION1
SINO
INSTRUCCION2
FIN DEL SI**

9.2.3 Según (selección múltiple)

En la selección múltiple, se evalúa el valor de una variable (valor) y dependiendo del mismo, s corresponde a uno de los casos de nuestro algoritmo ejecuta la acción correspondiente al mismo. En caso de no verificarse ningún caso, la estructura no hace nada, a menos que se agregue un caso para cuando esto ocurre.

**SEGUN <valor>
Valor1 INSTRUCCIÓN 1
Valor2 INSTRUCCIÓN 1
....
Valorn INSTRUCCIÓN n
FIN DEL SEGUN**

9.3 Estructuras iterativas

Este tipo de estructuras nos permiten realizar iteraciones (repeticiones) de nuestras instrucciones, de manera que cada acción se realizara varias veces.

9.3.1 Mientras

Esta estructura es un bucle que realizara iteraciones de las instrucciones hasta que deje de cumplirse una cierta condición.

**MIENTRAS (condicion) HACER
INSTRUCCIONES
FIN DEL MIENTRAS**

9.3.2 Hacer

Este bucle realizara las instrucciones un numero n de veces, especificado por nosotros.

**HACER(n veces)
INSTRUCCIONES
FIN DEL HACER**

9.3.2 Para

Este bucle también realizara un numero determinado de iteraciones, pero definidos en un intervalo (desde valor inicial hasta un valor final) y se indica como es el incremento de dicho valor.

**PARA (i=valor inicial; i < valor final ; i++) HACER
INSTRUCCIONES
FIN DEL PARA**