



# FORMATE@RTE

Cuadernillo de informática

**PROFESORADO EN TIC.  
UNIDAD CURRICULAR-INFORMÁTICA.  
SEGUNDO AÑO.  
ESTUDIANTES:  
MARCELA TOBARES. DNI 29690751  
JORGE NARVAEZ. DNI 27041553  
DOCENTE: NADIA MARTÍNEZ  
CICLO LECTIVO 2024**

## INTRODUCCIÓN

La computación es la ciencia que se ocupa de los procesos que describen y transforman información. A través de su estudio se sientan las bases para el diseño, la programación y el uso de computadoras digitales. Desde hace varios años, son muchos los ámbitos que aprovechan elementos de este campo de conocimiento e incorporan computadoras. Solo por citar algunos, se puede mencionar la agricultura (que utiliza la computación para realizar análisis de suelos, plagas y hacer un control automático de maquinaria, entre otras cosas), la arquitectura (para diseñar, planificar y hacer seguimientos de obras y procesos arquitectónicos) y la medicina (en la realización de diagnósticos, intervenciones quirúrgicas, etc.). Además, muchas herramientas de uso cotidiano como el correo electrónico, las redes sociales, las plataformas de comercio electrónico y los servicios de mensajería instantánea existen únicamente como consecuencia del desarrollo de esta ciencia. La ubicuidad de las soluciones computacionales hace imprescindible comprender los principios de la computación para tener una visión crítica del mundo que nos rodea.

## OBJETIVOS

El objetivo de este cuadernillo de informática es proporcionar a los estudiantes del ciclo básico del nivel secundario los conocimientos y habilidades necesarios para comprender los fundamentos de la tecnología informática, familiarizarse con el funcionamiento de un ordenador, aprender a utilizar adecuadamente el software y las herramientas informáticas básicas, y desarrollar una actitud crítica y responsable en el uso de la tecnología digital en su vida cotidiana.



## Contenido

INTRODUCCIÓN .....	1
OBJETIVOS .....	1
Unidad 1 .....	5
¿Qué es Informática? .....	5
¿Qué es una Computadora? .....	5
El computador.....	6
Periféricos de entrada y salida. ....	7
Pantalla de inicio. ....	9
Diferencia entre archivo y carpeta.....	10
Ingresar al Explorador de Windows. ....	10
Unidad 2 .....	13
Sistemas operativos. ....	13
Análisis y manipulación de los sistemas operativos.....	14
Tipos de sistemas operativos libres y propietarios.....	17
Tipos de archivos .....	18
Tipos    de archivos de documentos .....	19
Tipos de archivos de imagen .....	19
Tipos de archivos de vídeo .....	20
Tipos de archivos de audio .....	21
Unidad 3.....	25
¿Qué es un programa informático? .....	25
¿Qué es un lenguaje de programación? .....	26
TIPOS DE LENGUAJES DE PROGRAMACIÓN.....	27
Código binario .....	29
Programa.....	30
Algoritmo .....	32
Pseudocódigo .....	34
Nociones básicas: variables, tipos y expresiones .....	34
Estructura general del pseudocódigo.....	36
Estructuras componentes del pseudocódigo .....	36
Estructura secuencial .....	36
Estructura selectiva.....	36
Estructura iterativa .....	40
Bibliografía: .....	445



# FORMATE@RTE

*Cuadernillo de informática*

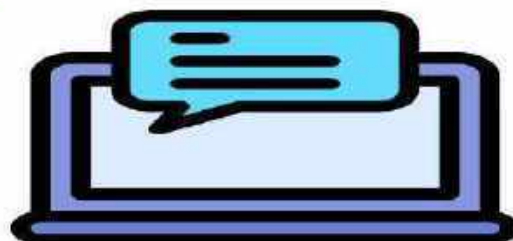
*Conceptos de  
informática y  
componentes  
principales*



*Sistemas  
operativos.*



*Programas y  
comandos  
básicos*





The background of the slide is a light blue gradient with a collage of various computer-related icons in shades of blue, green, and yellow. These icons include a mouse, a smartphone, a USB drive, a Wi-Fi symbol, a monitor, a keyboard, a camera, a folder, a cloud, a paperclip, and a document with a checkmark.

# **UNIDAD 1**

**Conceptos básicos de informática**

**Definición de informática.**

**Tipos de sistemas informáticos.**

**Datos e información.**

**Definición y tipos de computadoras**

**Componentes de las computadoras.**

**Hardware y Software**

**Periféricos.**

.

## Unidad 1

### ¿Qué es Informática?

Informática es un concepto sinónimo de computación, y lo definiremos como conjunto de conocimientos científicos y de técnicas que hacen posible el tratamiento automático de la información por medio de computadoras. La informática combina los aspectos teóricos y prácticos de la ingeniería electrónica, teoría de la información, matemática, lógica y comportamiento humano. Los aspectos de la informática cubren desde la programación y la arquitectura informática hasta la inteligencia artificial y la robótica.



### ¿Qué es una Computadora?

Una computadora es un dispositivo electrónico capaz de recibir un conjunto de instrucciones y ejecutarlas realizando cálculos sobre los datos numéricos, o bien compilando y correlacionando otros tipos de computadoras.



## CONCEPTOS BÁSICOS DE INFORMÁTICA

A través de la historia, el hombre como ser social ha necesitado transmitir y procesar información. En el pasado, para comunicarse utilizaba señales de humo, destellos con espejos, comunicación por medio de silbidos; más adelante, con el desarrollo de la tecnología, lo hizo a través de la clave Morse y el teléfono. Por otra parte, para realizar cálculos y procesar información, el hombre inicialmente comenzó a contar con los dedos, luego con piedritas, trozos de madera y tablas de arcilla, creó el ábaco, después las tablas y estructuras de Naiper, un sinfín de calculadoras, la máquina tabuladora, la máquina de Turing, y así sucesivamente hasta llegar al computador actual.

Se puede decir que el hombre no ha cesado de crear máquinas y métodos con el fin de procesar, transmitir y reproducir la información y para simplificar los trabajos rutinarios y repetitivos, generalmente de cálculo y de gestión; es así como nace la informática, como la ciencia encargada de crear, desarrollar y modificar estas máquinas y métodos.

El término informática viene de la contracción de las palabras información y automática. Existen diversas definiciones del término informática, pero el más extendido es: **Informática es la ciencia que estudia el tratamiento automático y racional de la información.**

# Formate@rte - Cuadernillo de informática



Se habla de tratamiento automático ya que son las máquinas las que realizan los trabajos de captura, proceso y presentación de la información, y se dice que es racional porque todos los procesos se desarrollan a través de programas que se basan en el razonamiento humano.

## El computador

Es una máquina capaz de recibir datos de entrada, realizar operaciones aritméticas y lógicas, y almacenar y reproducir la información resultante; está controlado por un programa o conjunto de instrucciones previamente almacenadas en él. **Las operaciones lógicas** pueden ser: escoger, copiar, mover, comparar y ejecutar. Otras definiciones del computador, tal como menciona Ferreira son: Máquina o dispositivo capaz de recibir información, procesarla y entregar resultados en la forma esperada. Equipo electrónico (**hardware**) que recibe instrucciones en forma de programas (**software**) para resolver diferentes tareas utilizando algoritmos.



**Hardware y software:** Todos los computadores constan de hardware y software. El uno no puede existir sin el otro.

**Hardware:** Es la parte física o tangible del computador; por ejemplo: teclado, mouse, memoria, impresora o parlantes.

**Software:** Es la parte lógica del computador; por ejemplo: los datos y programas almacenados en el computador.





# Formate@rte - Cuadernillo de informática

## Periféricos de entrada y salida.

**El Hardware** es el Esquema básico de los computadores, allí encontramos las unidades de entrada y las de salida.

**Unidades de entrada:** También llamados periféricos o dispositivos de entrada, se encargan de ingresar los datos y los programas desde el exterior a la memoria central, además, convierten esos datos, que son entendibles a los usuarios en un lenguaje máquina. Algunas de estas unidades son: **el teclado, el mouse, el escáner y el lápiz óptico.**

Hay otros que, en algunas circunstancias, funcionan sin intervención humana, como los sensores de distintos tipos –de temperatura, de proximidad, etc.–, las cámaras digitales o los módems –que permiten que la computadora reciba información de Internet–.

**Unidades de salida:** También llamados **periféricos o dispositivos de salida**, convierten los resultados que se encuentran en forma de dígitos a información entendible por el usuario. Por ejemplo: **impresora, monitor o plotter.**



**Unidad central de procesamiento (CPU – Central Process Unit)** es la parte principal o centro neurálgico de un computador y su función es coordinar y realizar todas las operaciones del sistema. Consta de procesador central y memoria principal.



**Procesador Central Consta de la Unidad de Control (U.C.) y la Unidad Aritmético Lógica (ALU)**

**Unidad de Control:** Governa el resto de las unidades, descifra las instrucciones y comandos, controlando que se ejecuten correctamente y en la secuencia solicitada.

**Unidad Aritmético Lógica:** Realiza todas las operaciones aritméticas y lógicas.

**Memoria principal o memoria primaria:** Almacena los datos necesarios para que el sistema realice determinado trabajo.



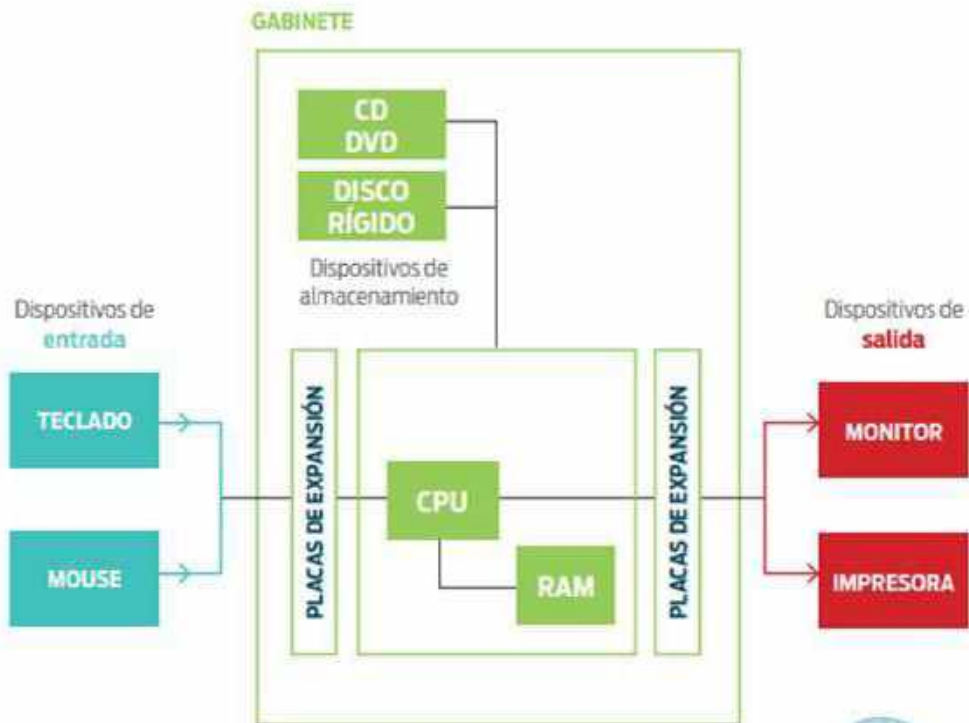
# Formate@rte - Cuadernillo de informática

**RAM** (Random Access Memory- Memoria de Acceso Aleatorio). Es una memoria volátil o fugaz. La información almacenada en ella se mantiene mientras haya suministro de energía eléctrica, su contenido se borra cuando se apaga el computador. En esta memoria el computador almacena los datos y programas que el usuario está usando en un momento determinado. Es una memoria de lectura y escritura en donde el usuario puede almacenar o acceder información (en forma temporal).



**ROM** (Read Only Memory Memoria de lectura). Es permanente. Aquí no se puede escribir información, el computador sólo lee las instrucciones que se encuentran almacenadas, las cuales han sido colocadas allí por el fabricante.

**Memoria auxiliar o almacenamiento secundario:** Son dispositivos de almacenamiento masivo donde la información se almacena por un período de tiempo y se puede recuperar en cualquier momento. Esta información puede estar alojada en el dispositivo sin importar si está o no conectado a la red eléctrica. Ejemplos: disco duro, memorias USB, CD, DVD y tarjetas de memoria.



**Software** es el Sistema operativo encargado de controlar los procesos básicos del PC (del inglés Personal Computer, que significa computador personal), administra los recursos y dispositivos conectados al computador, permite el funcionamiento de las aplicaciones o programas que se encuentran en la computadora.

## Formate@rte - Cuadernillo de informática

El **sistema operativo** se está almacenado en forma permanente en el disco duro, si el sistema operativo está dañado o no hay sistema operativo instalado en el computador, el computador no funciona. **Windows**, según Álvarez, es un sistema operativo con una interfaz gráfica de usuario basada en ventanas, presentando la información mediante íconos, ventanas, botones, barras de desplazamiento, casillas de verificación, etc., los cuales hacen más fácil el diálogo entre el usuario y la máquina. Windows ha evolucionado para dar origen a diversas versiones, de las cuales las más utilizadas en actualidad son: **Windows XP, Windows 7.0 y Windows 8.0. Windows 8:** es un sistema operativo para el manejo de plataformas en formato computador de escritorio (desktop PC), portátiles, tabletas, netbooks, servido de res o centros multimedia. Esta versión de Windows sigue tomando como base un sistema robusto y de gran rendimiento como es Windows 7, pero presenta una pantalla de inicio completamente nueva adaptándose a las distintas plataformas hardware, desde las tabletas táctiles y compactas y los portátiles ligeros hasta los PC de alto rendimiento y equipos “todo en uno” con pantallas de alta definición. **Pantalla de inicio de Windows 8:** aparece una pantalla con aplicaciones en forma de mosaico a las que es posible acceder con un solo clic. El escritorio aparece como otra aplicación y al cual se ingresa de igual forma.

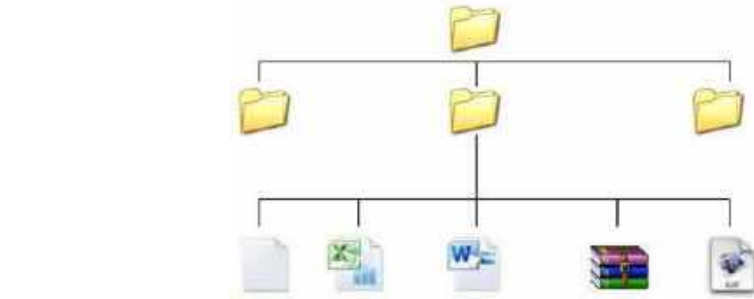


### Pantalla de inicio.

**El escritorio y sus partes:** Se puede observar que en esta versión no aparece el botón inicio en el escritorio, sin embargo, se puede acceder a la pantalla inicio pulsando la tecla Windows.

**Explorador de Windows:** El explorador de archivos es una herramienta de Windows que permite visualizar y organizar la información que se encuentra guardada en las unidades de disco duro, la memoria USB u otras unidades de almacenamiento secundario y en las unidades de red. Para entender este tema hacemos una comparación sobre el modo de guardar nuestros documentos físicos (en papel) y la forma de almacenar la información en el computador. Cuando almacenamos documentos físicos normalmente utilizamos un archivador con gavetas, y dentro de ellas, carpetas o folders que se pueden ordenar alfabéticamente, permitiendo su fácil ubicación. Así mismo, sucede cuando almacenamos archivos en el computador; se crean carpetas para guardarlos, de tal forma que se pueda acceder a ellos fácilmente.

## Diferencia entre archivo y carpeta




**Archivo** es un conjunto de datos del mismo tipo, como por ejemplo un proyecto, un memorando, un informe, una nómina, una factura, una carta, una canción, un video.

Una carpeta es una subdivisión del dispositivo de almacenamiento creada para organizar mejor la información; una carpeta puede contener archivos y carpetas a las que se les llama subcarpetas. En el Explorador de Windows los archivos y carpetas se encuentran en las unidades del equipo, tales como, discos duros, las unidades de red, las unidades de almacenamiento extraíble como memoria USB, CD y tarjetas de memoria. Con el Explorador de Windows se puede copiar, mover, cambiar el nombre, buscar y eliminar archivos o carpetas.



## Ingresar al Explorador de Windows.



Desde Windows 8 se puede ingresar al Explorador de Windows pulsando la tecla Windows y escribiendo Explorador de Windows. Cuando termine la búsqueda, haga clic en el Explorador. Si tiene otra versión de Windows, se pulsa el botón Inicio de la barra de tareas, luego se hace clic en Buscar Programa y archivos y se escribe Explorador de Windows, finalmente se presiona la tecla enter- Inicio, enseguida aparece la siguiente ventana Explorador de Windows. La ventana está dividida en dos secciones: - En el panel de la izquierda aparecen: bibliotecas, favoritos, unidades de disco con carpetas y sub- carpetas en forma jerárquica. - En el panel de la derecha se muestra el contenido de la carpeta activa, es decir, los archivos y carpetas que se encuentran dentro de la carpeta seleccionada.

# Formate@rte - Cuadernillo de informática

## REPASAMOS LOS CONCEPTOS.

Todos los componentes de la computadora se conectan a la placa madre: la CPU y la memoria RAM de manera directa, y los demás componentes a través de placas de expansión, como las placas de video, de sonido, puertos USB, etc. La CPU es la encargada de controlar las acciones de los demás componentes: carga información en la memoria principal, y a sea desde una unidad de almacenamiento o usando dispositivos de entrada, la procesa ejecutando las instrucciones de un programa una por una, y produce nueva información que guarda en un dispositivo de almacenamiento o comunica al exterior mediante dispositivos de salida.



Para más información puedes ver el video en YouTube:

[https://www.youtube.com/watch?v=6fz1Mjv\\_VHc&t=241s](https://www.youtube.com/watch?v=6fz1Mjv_VHc&t=241s)

Completar la ficha luego de terminar la lectura de la unidad 1

característica	Nombre del componente
También se la llama procesador. Controla al resto de los componentes de la computadora.	
Los lectores de código de barras y los micrófonos son ejemplos de este tipo de componentes.	
Es una placa a la que se conectan los componentes principales de una computadora. Tiene circuitos impresos que permiten la comunicación entre ellos.	
Es el componente de hardware que se encarga de ejecutar las instrucciones de los programas. Para hacerlo, realiza operaciones aritméticas y lógicas.	
Tiene una pequeña memoria interna, denominada caché, a la que puede acceder muy rápidamente. Allí mantiene una copia de la porción de la RAM que utiliza con mayor frecuencia y de este modo acelera su velocidad de trabajo.	
Permite la comunicación entre el procesador y la memoria RAM, y entre el procesador y los dispositivos de entrada y salida.	
Componente utilizado para leer, grabar o guardar datos, que no necesita estar encendido ni recibir un suministro permanente de energía para conservar la información.	
Los teclados y los ratones son ejemplos de este tipo de componentes.	
También se la conoce como motherboard.	



# Sistemas operativos: Definición y características.

## Definición y características.

## Funciones y tipos .

## Administración, estructura y almacenamiento de la información.

## Tipos de archivos

## Herramientas y operaciones básicas.



## Unidad 2

### Sistemas operativos.

Un sistema operativo es un software que actúa como intermediario entre el hardware de un dispositivo informático y los programas de aplicación. Se encarga de gestionar los recursos del sistema, como la memoria, el procesador, los dispositivos de entrada y salida, y de proporcionar una interfaz para que los usuarios puedan interactuar con el sistema de manera eficiente



A continuación, se describen algunos de los aspectos más importantes a tener en cuenta en este análisis:



- ❖ **Atributos:** incluyen características como la multitarea, el multiprocesamiento, la multiprogramación, la gestión de memoria, la gestión de archivos, la interfaz de usuario, la seguridad, la portabilidad, entre otros. Estos atributos son fundamentales para entender el funcionamiento y las capacidades de un sistema operativo.
- ❖ **Partes:** Un sistema operativo está compuesto por varias partes o módulos que se encargan de funciones específicas, como el núcleo (kernel), los controladores de dispositivos, el gestor de memoria, el planificador de procesos, el sistema de archivos, entre otros. Cada parte cumple un rol importante en el funcionamiento global del sistema operativo.



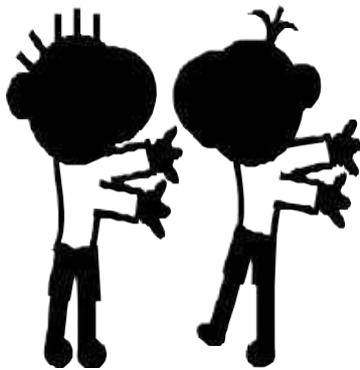
## FORMATE@RTE-Cuadernillo de informática

- ❖ **Elementos:** incluyen los procesos, los hilos, los archivos, los dispositivos de entrada/salida, la memoria, las colas de procesos, entre otros. Estos elementos interactúan entre sí y con el sistema operativo para llevar a cabo las tareas y operaciones del sistema.
- ❖ **Características:** pueden incluir su capacidad de ser multitarea, multiusuario, multiplataforma, su capacidad de gestión de recursos, su eficiencia en el uso de la memoria y del procesador, su capacidad de protección y seguridad, entre otras.
- ❖ **Componentes:** pueden ser hardware, software o una combinación de ambos. Éstos incluyen el núcleo del sistema operativo, los controladores de dispositivos, los servicios del sistema, las utilidades del sistema, las aplicaciones de sistema, entre otros.



### Análisis y manipulación de los sistemas operativos

**El análisis y la manipulación de los sistemas operativos** implica identificar y comprender los atributos, partes, elementos, características y componentes que los componen, para poder entender su funcionamiento, su estructura y sus capacidades, y poder gestionarlos de manera eficaz.



#### Algunas características comunes de los sistemas operativos son:

- **Gestión de recursos:** Los sistemas operativos se encargan de gestionar los recursos del sistema, como la memoria, el procesador, los dispositivos de entrada/salida, entre otros, asignándolos de manera eficiente a los programas y procesos que se ejecutan en la computadora.
- **Interfaz de usuario:** Proporcionan una interfaz gráfica o de línea de comandos que permite a los usuarios interactuar con la computadora, ejecutar programas, gestionar archivos, configurar el sistema, entre otras tareas.
- **Multitarea:** Los sistemas operativos permiten la ejecución de varios programas simultáneamente, alternando entre ellos de manera rápida para ofrecer la ilusión de que se están ejecutando al mismo tiempo.





- **Multiusuario:** Los sistemas operativos pueden admitir múltiples usuarios que acceden al sistema de forma concurrente, cada uno con su propio entorno de trabajo y sus propias configuraciones.
- **Seguridad:** Los sistemas operativos incluyen mecanismos de seguridad para proteger los datos y recursos del sistema, como permisos de acceso, contraseñas, cifrado de datos, entre otros.
- **Gestión de archivos:** Permiten la creación, modificación, organización y eliminación de archivos en el sistema de almacenamiento, así como la gestión de directorios y estructuras de almacenamiento.
- **Gestión de memoria:** Controlan el uso de la memoria del sistema, asignando y liberando espacio de memoria para los programas en ejecución, garantizando que no se produzcan conflictos ni desperdicio de recursos.



Los sistemas operativos son fundamentales para el funcionamiento de cualquier dispositivo informático, ya que facilitan la gestión de recursos, la interacción con el usuario y la ejecución de programas de manera eficiente y segura.

Las funciones de un sistema operativo son variadas y abarcan desde la gestión de recursos del sistema hasta la interacción con el usuario. A continuación, se detallan algunas de las principales funciones





## FORMATE@RTE-Cuadernillo de informática

Gestión de recursos	El sistema operativo se encarga de gestionar los recursos del sistema, como la memoria, el procesador, los dispositivos de entrada/salida, el almacenamiento, entre otros, asignándolos de manera eficiente a los programas y procesos que se ejecutan en la computadora.
Interfaz de usuario	Proporciona una interfaz gráfica o de línea de comandos que permite a los usuarios interactuar con la computadora, ejecutar programas, gestionar archivos, configurar el sistema, entre otras tareas.
Gestión de procesos	Controla la ejecución de procesos y programas en el sistema, asignando recursos, planificando la ejecución de tareas, gestionando la concurrencia y la comunicación entre procesos, y garantizando la estabilidad y eficiencia del sistema.
Gestión de memoria	Controla el uso de la memoria del sistema, asignando y liberando espacio de memoria para los programas en ejecución, gestionando la memoria virtual, y evitando conflictos y pérdida de recursos.
Gestión de archivos	Permite la creación, modificación, organización y eliminación de archivos en el sistema de almacenamiento, gestiona los directorios y estructuras de almacenamiento, y proporciona mecanismos de protección y seguridad de los datos.
Gestión de dispositivos	Controla la interacción entre el sistema operativo y los dispositivos de hardware, gestionando los controladores de dispositivos, administrando la comunicación y el acceso a los dispositivos, y garantizando su correcto funcionamiento.
Seguridad	Incluye mecanismos de seguridad como permisos de acceso, control de usuarios, cifrado de datos, autenticación, auditoría de eventos, entre otros, para proteger los datos y recursos del sistema frente a amenazas y accesos no autorizados.



Estas son solo algunas de las funciones principales de un sistema operativo, que son fundamentales para el correcto funcionamiento y la eficiencia de cualquier dispositivo informático.



## Tipos de sistemas operativos libres y propietarios



Sistema operativo o software libre:	Se distribuye bajo licencia que permite las llamadas 4 libertades: libertad de uso, distribución, copia y modificación a todos los usuarios que lo han adquirido. Es decir, una vez obtenido el mismo, puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas.
Software propietario	La persona física o jurídica (compañía, empresa,) que posee los derechos de explotación sobre el software, tiene la posibilidad de controlar y restringir los derechos del usuario sobre el programa limitando una o varias de las cuatro libertades. Las condiciones en las que el usuario podrá utilizar el programa, se establecen en las Licencias de Usuario Final (End User License Agreement -EULA según sus siglas en inglés).

**Administración, estructura y almacenamiento de la información: archivos y carpetas.**



Una de las principales funciones de un Sistema Operativo es la administración del almacenamiento de información, para lo cual es necesario contar con un “Sistema de

**Archivos.**

**Archivo:** es una colección de datos que se almacena en un medio físico y a la cual se le asigna un nombre.

**Carpeta:** es un contenedor donde se pueden almacenar archivos o también otras carpetas.

**Subcarpeta:** Una carpeta dentro de una carpeta

### **Tipos de archivos**

Los diferentes tipos de archivos tienen diferentes fines; algunos tipos de archivos son más adecuados para la web, mientras que otros son más adecuados para la impresión u otros medios.





## Tipos de archivos de documentos

<b>PDF:</b>	Los archivos de formato de documento portátil (portable document format, PDF) crean documentos presentables, incluyendo formularios y otros documentos que podrías querer compartir o imprimir sin permitir que la gente los modifique.
<b>DOC y DOCX:</b>	<i>Te permiten agregar imágenes, tablas y otros elementos directamente a los documentos, además de personalizar el tamaño, color, tipo, alineación de la tipografía, y más cosas.</i>
<b>TXT:</b>	<i>Documentos de texto sin formato, se utilizan para contenido digital que solo contiene texto</i>
<b>HTML y HTM:</b>	<i>Los archivos de lenguaje de marcado de hipertexto, crean la estructura de una página web</i>
<b>XLS y XLSX:</b>	<i>Son utilizados por programas como Excel para hojas de cálculo.</i>

## Tipos de archivos de imagen

<b>GIF:</b>	Las imágenes de formato de intercambio de gráficos (graphics interchange format, GIF) pueden ser estáticas o animadas.
<b>JPEG o JPG:</b>	Formato de archivo de imagen utilizado con más frecuencia, y su pequeño tamaño lo hace ideal para uso en web suelen no tener buena calidad de imagen.
<b>PNG:</b>	Archivo de gráficos de red portátiles (portable network graphics, PNG) puede ser hasta 10 veces más grande que un JPEG similar, las imágenes PNG también son buenas para su uso en web., permite una gama más amplia de colores, compresión sin pérdida y transparencia.





<b>SVG:</b>	Archivos de gráficos vectoriales escalables (scalable vector graphics, SVG) son imágenes que puede escalar a diferentes tamaños sin poner en riesgo la calidad de la imagen
<b>TIFF o TIF:</b>	Formato de archivo de imagen etiquetado (tagged image file format, TIFF) son compatibles con los colores RGB y CMYK, lo que los hace perfectos para imprimir. Se recomiendan en encabezados, folletos o invitaciones, los archivos TIFF que utilizan el formato CMYK garantizarán que tus colores sean iguales en tu ordenador y en papel.

## Tipos de archivos de vídeo



<b>MP4</b>	Los archivos MP4 son perfectos si estás cargando un vídeo en la web, ya sea tu sitio web o una cuenta de redes sociales.
<b>AVI:</b>	Los archivos de audio/vídeo (audio video interleave, AVI) no son los más pequeños, pero pueden proporcionar y mantener una mejor calidad que los archivos MP4
<b>MOV</b>	Se utilizan en la etapa de edición de vídeo más que nada. Apple creó el tipo de archivo MOV para el reproductor de vídeo QuickTime, pero los sistemas operativos Windows también admiten este tipo.
<b>FLV:</b>	El vídeo flash (flash video, FLV) es un formato desarrollado por Adobe y diseñado para utilizarse con Adobe Flash Player. Los archivos FLV son ampliamente compatibles y de tamaño pequeño,
<b>AVCHD:</b>	Estos archivos son usados principalmente por dispositivos de grabación, con la ventaja de que son fáciles de ver en un televisor. Los archivos AVCHD pueden crear discos de grabación desde dispositivos compatibles



## Tipos de archivos de audio



<b>M4A</b>	Pueden ofrecer compresión sin pérdida, aunque estos archivos M4A son de mayor tamaño, se utiliza sobre todo para canciones en las plataformas de Apple e iTunes. Fue creado por Apple.
<b>MP3</b>	Funcionan con casi todos los dispositivos del mundo, aunque la compresión con pérdida significa que ofrecen menor calidad que otros formatos.
<b>WAV</b>	Son los de mejor calidad. Estos archivos de audio no comprimen el audio de origen en absoluto, lo que significa que suena exactamente según lo previsto. Estos archivos se pueden utilizar para producciones de TV, DVD y rad
<b>FLAC</b>	Formato de audio sin pérdida, pero son mucho más grandes que los tipos de archivos de audio comunes, como los MP3

## Las operaciones básicas sobre archivos y carpetas



**Crear:** Crear nuevos archivos o carpetas.

(Con el documento abierto, haga clic en Archivo > Guardar como. En Guardar como, seleccione dónde desea crear la nueva carpeta. Es posible que necesite hacer clic en Examinar o en Equipo y desplazarse hasta la ubicación de la nueva carpeta. En el cuadro de diálogo Guardar como que se abre, haga clic en Nueva

carpeta).



## FORMATE@RTE-Cuadernillo de informática

**Duplicar:** Crear copias de archivos o carpetas existentes.

1. Seleccione el **archivo o carpeta** que desea **duplicar** en el panel de vista.
2. Elija Editar -> **Duplicar**. También, haga clic con el botón derecho del ratón sobre el **archivo o carpeta** en el panel de vista y, a continuación, elija **Duplicar**.

**Renombrar:** Cambiar el nombre de archivos o carpetas.

**Copiar:** Crear copias de archivos o carpetas en otro lugar.

(haga clic con el botón derecho y seleccione Copiar. Para copiar el elemento: haga clic en Ctrl+C. Vaya a la carpeta a la que desea mover o copiar el elemento y, a continuación, haga clic en Ctrl+V.

**Pegar:** Mover archivos o carpetas copiados previamente a una ubicación deseada. (haga clic en Ctrl+V.)

**Propiedades:** Ver y modificar las propiedades de archivos o carpetas, como tamaño, tipo, fecha de creación, etc.



¡Escanéame!



## FORMATE@RTE-Cuadernillo de informática

Para más información puedes visitar: <https://www.youtube.com/watch?v=CJ-2LjCee00>

<https://create.kahoot.it/share/sistemas-operativos/609497e4-30da-48b4-a401-65c959f61bee>

Ficha de comprensión de la unidad 2-Completar en base a lo abordado en la unidad.

Característica que Proporciona una interfaz gráfica o de línea de comandos que permite a los usuarios interactuar con la computadora, ejecutar programas, gestionar archivos, configurar el sistema, entre otras tareas.	
<i>Los archivos de lenguaje de marcado hipertexto, crean la estructura de una página web</i>	
Funcionan con casi todos los dispositivos del mundo, aunque la compresión con pérdida significa que ofrecen menor calidad que otros formatos	
Es un contenedor donde se pueden almacenar archivos o también otras carpetas.	
Se distribuye bajo licencia que permite las llamadas 4 libertades: libertad de uso, distribución, copia y modificación a todos los usuarios que lo han adquirido. Es decir, una vez obtenido el mismo, puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas.	
Es un software que actúa como intermediario entre el hardware de un dispositivo informático y los programas de aplicación. Se encarga de gestionar los recursos del sistema, como la memoria, el procesador, los dispositivos de entrada y salida, y de proporcionar una interfaz para que los usuarios puedan interactuar con el sistema de manera eficiente	



The background is a light blue surface covered with various stylized, colorful icons representing technology and computing. These include a computer mouse, a smartphone, a Wi-Fi symbol, a USB drive, a laptop, a monitor, a keyboard, a camera, a printer, and various cables and connectors. The icons are scattered around the central text, creating a tech-themed border.

## **UNIDAD 3**

**Programa informático.  
Tipos de programas.  
Lenguaje de programación.  
Tipos de lenguaje de  
Programación**



## Unidad 3

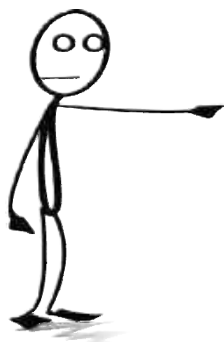
### ¿Qué es un programa informático?



Un programa informático o programa de computador **es una pieza de software**, es decir, una secuencia compleja de instrucciones y procesos orquestados para cumplir una tarea específica en un computador o sistema de computadores. Estos programas pueden ser programas preinstalados en el computador, como el Sistema Operativo que controla todo el funcionamiento del mismo, o pueden ser añadidos adicionalmente por el usuario.

Por lo general, los programas de computador disponen de cierto margen de recursos del sistema informático mientras se ejecutan, y cumplen roles de todo tipo en el mismo, desde **controlar los recursos y las operaciones internas del computador**, hasta mediar con el usuario y permitirle trabajar, recrearse, explorar Internet, etc.

Estos programas son obra de programadores, profesionales de la informática que componen el código matriz (código fuente) de cada pieza de software y que son de alguna manera sus “autores”.



Dicho código consiste en instrucciones (declarativas o imperativas) que luego son convertidas en archivos ejecutables por el usuario o por el sistema y que se ejecutan en la Unidad Central de Procesamiento (CPU).

**Para su creación, los programadores emplean, a su vez, programas informáticos que sirven para redactar estos códigos, es decir, software que sirve para hacer software y se conoce como *lenguaje de programación*.**

Por otro lado, muchos programas informáticos **son capaces de actualizarse y modificarse a través de la descarga de datos de la Internet**, manteniéndose al día o redefiniendo sus componentes, como ocurre con los Programas Antivirus o con los Sistemas Operativos respectivamente.



## Tipos de programas informáticos



**Software del sistema:** Aquellos programas básicos que permiten el uso del computador, vinculando al usuario con los distintos aspectos de hardware y/o redes del sistema, así como con las aplicaciones instaladas en el mismo. Este tipo de programas dan soporte a las aplicaciones y en muchos casos vienen ya instalados e incluso empotrados en el computador. Por ejemplo: controladores de hardware, Sistemas Operativos, líneas de comandos, etc.

**Software de aplicación:** Toda la diversidad de programas secundarios que podemos descargar o instalar en nuestro computador y que permiten llevar a cabo numerosas funciones: procesadores de texto, hojas de cálculo, exploradores Web, videojuegos, software de telecomunicaciones, reproductores multimedia, etc.

### ¿Qué es un lenguaje de programación?



En informática, se conoce como lenguaje de programación a **un programa destinado a la construcción de otros programas informáticos**. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano.

Dicho lenguaje **está compuesto por símbolos y reglas sintácticas y semánticas**, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una aplicación o pieza de software determinado. Así, puede llamarse también lenguaje de programación al resultado final de estos procesos creativos.

La implementación de lenguajes de programación **permite el trabajo conjunto y coordinado**, a través de un conjunto afín y finito de instrucciones posibles, de diversos programadores o arquitectos de software, para lo cual estos lenguajes imitan, al menos formalmente, la lógica de los lenguajes humanos o naturales

No deben confundirse, sin embargo, con los distintos tipos de lenguaje informático. Estos últimos representan una categoría mucho más amplia, en donde están contenidos los lenguajes de programación y muchos



otros protocolos informáticos, como el HTML de las páginas web.

## TIPOS DE LENGUAJES DE PROGRAMACIÓN



<b>Lenguajes de bajo nivel.</b>	Se trata de lenguajes de programación que están diseñados para un hardware específico y que por lo tanto no pueden migrar o exportarse a otros computadores. Sacan el mayor provecho posible al sistema para el que fueron diseñados, pero no aplican para ningún otro.
<b>Lenguajes de alto nivel.</b>	Se trata de lenguajes de programación que aspiran a ser un lenguaje más universal, por lo que pueden emplearse indistintamente de la arquitectura del hardware, es decir, en diversos tipos de sistemas. Los hay de propósito general y de propósito específico.
<b>Lenguajes de nivel medio.</b>	Este término no siempre es aceptado, que propone lenguajes de programación que se ubican en un punto medio entre los dos anteriores: pues permite operaciones de alto nivel y a la vez la gestión local de la arquitectura del sistema.



Otra forma de clasificación a menudo es la siguiente:

<b>Lenguajes imperativos.</b>	Menos flexibles, dada la secuencialidad en que construyen sus instrucciones, estos lenguajes programan mediante órdenes condicionales y un bloque de comandos al que retornan una vez llevada a cabo la función.
<b>Lenguajes funcionales.</b>	También llamados procedimentales, estos lenguajes programan mediante funciones que son invocadas conforme a la entrada recibida, que a su vez son resultado de otras funciones.





Algunos de los lenguajes de programación más conocidos son:



<b>BASIC.</b>	Su nombre proviene de las siglas de Beginner's All- purpose Symbolic Instruction Code (Código simbólico de instrucciones de propósito general para principiantes), y es una familia de lenguajes imperativos de alto nivel, aparecidos por primera vez en 1964. Su versión más actual es Visual Basic .NET
<b>COBOL.</b>	Su nombre es un acrónimo para <i>Common Business-Oriented Language</i> (Lenguaje común orientado a los negocios) y se trata de un lenguaje de programación universal creado en 1959, orientado principalmente a la informática de gestión, es decir, empresarial.
<b>FORTRAN.</b>	Su nombre proviene de <i>The IBM Mathematical Formula Translating System</i> (El sistema de traducción de fórmulas matemáticas de IBM), y es un lenguaje de programación de alto nivel, propósito general y de tipo imperativo, diseñado para aplicaciones científicas y de ingeniería.
<b>JAVA</b>	Un lenguaje de programación de propósito general, orientado a objetos, cuyo espíritu se resume en las siglas WORA: <i>Written Once, Run Anywhere</i> , es decir: Escrito una vez, funciona en cualquier parte. La idea era diseñar un lenguaje universal empleando sintaxis derivada de los lenguajes C y C++, pero empleando menos utilidades de bajo nivel que cualquiera de ambos.



Un lenguaje de programación es un idioma artificial diseñado para expresar  
cálculos  
que pueden ser llevados a cabo por las computadoras. Puede usarse  
para crear programas que controlen el comportamiento físico y lógico de una  
máquina,  
para expresar algoritmos con precisión o como modo de comunicación humana.  
Estos lenguajes permiten especificar de manera precisa sobre qué datos debe  
operar una computadora, cómo deben ser almacenados o transmitidos  
y qué acciones debe tomar  
bajo una gran cantidad de opciones posibles. Todo esto, a través de un lenguaje  
que intenta ser relativamente próximo al lenguaje humano o natural.

Veremos más adelante que los lenguajes de programación están formados por un conjunto de símbolos y reglas sintácticas y semánticas específicas que definen su estructura y el significado de sus elementos y expresiones.

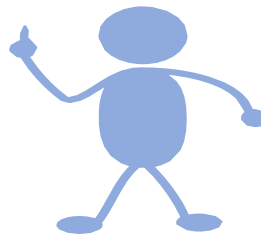


Una computadora funciona bajo el control de un programa, que ha sido desarrollado en un determinado lenguaje que puede ser reducido al código binario, y que debe estar almacenado en la unidad de memoria o disco duro.

## Código binario

Para que la computadora entienda nuestras instrucciones debe usarse un lenguaje específico conocido como código máquina o código binario, que la máquina comprende fácilmente, pero que resulta excesivamente complicado para las personas. Se dice que un sistema es binario cuando solo caben dos posibles resultados o respuestas a un planteo determinado. El código binario es la base de la informática al reducir todas las posibles instrucciones interpretadas por la máquina a un código de unos y ceros (encendido/apagado; sí/no) por el que el microprocesador funciona y ejecuta las órdenes introducidas en él.

La unidad mínima de información en el código binario es el bit –el término proviene del inglés binary digit–. Un bit puede ser un 1 o un 0.



El concepto de bit está presente en las computadoras desde al menos 1936. En aquella época se utilizaban tarjetas perforadas: unas tarjetas de cartulina que servían para ingresar información e instrucciones en una computadora, que simplemente leía la presencia o ausencia de perforación y establecía un patrón. Pero su utilización con el nombre de bit se llevó a cabo por primera vez en 1948, en un trabajo académico realizado por Claude E. Shannon, ingeniero y matemático estadounidense, creador de la Teoría Matemática de la Comunicación. Explicamos que el bit es la unidad mínima de este código y como su nombre lo indica es binario, permite dos combinaciones: 0 o 1. Pero esta unidad mínima permite una secuencia de 8 bits contiguos, a la que se denomina byte. Aunque este concepto aplicado a la tecnología parece muy limitado, cuando se usa de forma combinada permite representar un sinnúmero de datos expresados en valores. Por ejemplo: 00000000, 01010101, 00110011, 00010001, y así en miles de combinaciones posibles. Es decir que, si asignáramos a cada byte un dato, podríamos representar cosas como verde, amarillo, azul y naranja.

Sobre estos principios básicos se desarrollan todos los lenguajes de programación: PHP; JavaScript, Python y Ruby, entre muchos otros.



## Programa

Un programa generalmente implementa –traduce a un lenguaje de programación concreto– un algoritmo.



Un algoritmo es una secuencia no ambigua, finita y ordenada de instrucciones que han de seguirse para resolver un problema.

Los programas suelen subdividirse en partes menores denominadas módulos, de forma que la complejidad algorítmica de cada una de las partes sea menor que la del programa completo, lo cual ayuda al desarrollo del problema. Pero antes de avanzar en cuestiones más complejas, es necesario que conozcan el término código fuente. El código fuente de un programa es un conjunto de líneas de texto en las que están expresadas las instrucciones que debe seguir la computadora para ejecutar dicho programa. En el código fuente de un programa está descrito por completo su funcionamiento. Este código es escrito por un programador en algún lenguaje de programación, pero en este primer estado no es directamente ejecutable por la computadora, sino que debe ser traducido a otro lenguaje –el lenguaje máquina o código objeto– que sí pueda ser ejecutado por el hardware de la computadora. Para esta traducción se usan los llamados compiladores, ensambladores, intérpretes y otros sistemas de traducción.



El área de la informática que se dedica a la creación de programas y por tanto a la creación de su código fuente es la de programación.

El término código fuente también se usa para hacer referencia al código fuente de otros elementos del software. Por ejemplo, el código fuente de una página web – que está escrito en el lenguaje de marcado HTML, en JavaScript u otros lenguajes de programación web– es posteriormente ejecutado por el navegador o browser para hacer visible la página cuando es visitada. Para la implementación de un programa ejecutable –un típico.exe para Microsoft– se deben llevar adelante dos pasos.

**1. Compilación** En este paso se traduce un programa escrito en un lenguaje de programación a otro, generando un programa equivalente que la máquina será capaz de interpretar.

**2. Enlazado** Es el proceso que une el código de los módulos y bibliotecas que forman un programa para generar el ejecutable final. Estos dos pasos se pueden hacer por separado, almacenando el resultado de la fase de compilación en un



## FORMATE@RTE- Cuadernillo de informática

archivo objeto –un .obj para Microsoft o para Unix– para enlazarlo en fases posteriores o crear directamente el ejecutable.

Un programa podría tener partes escritas en varios lenguajes distintos que se podrían compilar de forma independiente y luego enlazar para formar un único módulo ejecutable. Como a esta altura pueden imaginar, el proceso de creación de un programa, también llamado software, es complejo y en general representa mucho tiempo de trabajo de equipos de especialistas



El proceso de creación de un programa, desde el punto de vista de la ingeniería, podría sintetizarse en los siguientes pasos:

1. Reconocer la necesidad de un programa para solucionar un problema o identificar la posibilidad de automatización de una tarea.
2. Recopilar los requisitos del programa para definir qué es lo que debe hacer y para qué se necesita.
3. Realizar el análisis de los requisitos del programa. Debe quedar claro cómo debe realizar el programa las cosas que debe hacer; las pruebas que demuestren la validez del programa se pueden especificar en esta fase.
4. Diseñar la arquitectura del programa para descomponerlo en partes de complejidad abordable.
5. Implementar el programa, lo que consiste en realizar un diseño detallado, especificando completamente todo su funcionamiento.
6. Implantar o instalar el programa es ponerlo en funcionamiento junto con los componentes que pueda necesitar (bases de datos, redes de comunicaciones, etc.).



Como hemos visto hasta aquí, las computadoras no hablan nuestro idioma, necesitan un lenguaje específico pensado por el hombre para ellas. Además, requieren constantemente interpretar todas las instrucciones que reciben.

El programador desarrolla, en distintos lenguajes de programación, instrucciones que hacen posible la comunicación con el microprocesador de la computadora, utilizando términos y símbolos relacionados con el tipo de problema que se debe resolver.





## Algoritmo

Un algoritmo constituye una lista bien definida, ordenada y finita de operaciones, que permite encontrar la solución a un problema determinado. Dado un *estado inicial* y una *entrada*, es a través de *pasos sucesivos* y bien definidos que se llega a un *estado final*, en el que se obtiene una solución (si hay varias) o la solución (si es única).

### Ejemplo

*Problema: Gestionar la lista de compras que una empresa realiza durante un mes.*

### Solución

*Para resolver este problema de gestión muy general, se cuenta con las herramientas que se utilizan en otros problemas que ya tienen una resolución en la empresa (por ejemplo, un programa en Java con los datos en una base de datos). Así, para la solución del problema planteado, se proponen dos opciones:*

- ✓ Usar la base de datos de trabajo para guardar, también en esta, la lista de productos que se requiere comprar.
- ✓ Guardar una lista en entradas que se actualiza cada vez que se captura o se incluye un nuevo producto que la empresa necesita, y que se borra o elimina al momento que el producto ya está abastecido, y en salidas, cada vez que algún empleado necesite una impresión de dicha lista.

*En este ejemplo, el algoritmo global de resolución se compone de diversos pasos sucesivos de diálogo con el usuario (un empleado de la empresa), para mantener actualizada la lista de productos necesarios; asimismo, en pasos siguientes se precisa hacer una inserción y/o una eliminación o borrado de los productos (elementos) de la lista o una impresión en una forma legible.*

Un algoritmo puede ser expresado en:

- ❖ Lenguaje natural (a veces, este no resulta muy claro, pero es muy útil para problemas simples)
- ❖ Pseudocódigo
- ❖ Diagramas de flujo
- ❖ Programas

El uso de algún elemento de la lista anterior para la expresión de un algoritmo, se hace según el nivel de descripción de dicho algoritmo. Es evidente que el lenguaje natural es de mayor utilidad para transmitir las ideas del algoritmo. Al contrario, un programa es difícil de entender por simple lectura, aun por una



persona que conoce el lenguaje del programa, e imposible para aquellas que no lo conocen.

El pseudocódigo y los diagramas de flujo, en cambio, se sitúan en un punto intermedio de comprensión, entre el lenguaje natural y un programa. Estas dos herramientas poseen un poder de expresión equivalente; no obstante, los diagramas de flujo tienen la ventaja de ser más gráficos y visuales.

Con base en el ejemplo anterior, se puede afirmar que la parte de solución expresada en lenguaje natural tiene algunas ambigüedades para el usuario que no es el programador; por ejemplo, ¿qué significa la expresión “de pasos sucesivos de diálogo con el usuario”? Aunque, en ocasiones, también presenta ambigüedades hasta para el propio programador; por ejemplo, ¿cuáles son “los datos en una base de datos” ?, ¿una base de datos es relacional o de otro modelo?, ¿cuál interfaz?, ¿cómo se manejan las lecturas/escritura en dicha base de datos?

Las respuestas a las interrogantes anteriores se expresan de la siguiente forma:

- ✓ La primera ambigüedad (“pasos sucesivos”) se debe expresar lo más detallada posible por el destinatario del programa (el usuario).
- ✓ Los otros cuestionamientos son de detalles técnicos.

La descripción de un algoritmo usualmente se realiza en tres niveles:

1. Descripción de alto nivel. El primer paso consiste en la descripción del problema; luego, se selecciona un modelo matemático y se explica el algoritmo de manera verbal, posiblemente con ilustraciones, pero omitiendo detalles.
2. Descripción formal. En este nivel se usa un pseudocódigo o diagrama de flujo para describir la secuencia de pasos que conducen a la solución.
3. Implementación. Por último, en este nivel se muestra el algoritmo expresado en un lenguaje de programación específico, o algún objeto capaz de llevar a cabo instrucciones.

Para llegar a la implementación, primero se deben tener descripciones de alto nivel o formalmente explícitas, sobre todo cuando el trabajo de desarrollo de un algoritmo se hace en grupo.



## Pseudocódigo

---

Un pseudocódigo (falso lenguaje) está formado por una serie de palabras con un formalismo muy sencillo, que permite describir el funcionamiento de un programa. Se usa tanto en la fase de diseño como en la fase de análisis.

El pseudocódigo describe un algoritmo utilizando una mezcla de frases en lenguaje común, instrucciones de programación y palabras clave que definen las estructuras básicas. Su objetivo es permitir que el programador se centre en los aspectos lógicos de la solución de un problema.

El pseudocódigo utiliza expresiones matemáticas, expresiones lógicas y la noción de variable (sencilla, arreglo, pila, cola, conjunto, etcétera). El pseudocódigo se puede extender para expresar tipos complejos y operaciones entre variables y constantes de este nuevo tipo.

### Estructura de los programas

El lenguaje utilizado en Turbo Pascal es estructurado, lo que significa que cada programa requiere una forma específica de escritura para que sea entendido por el compilador.

Todo programa cuenta con algunas partes o módulos los cuales son:

<b>Cabecera</b>
<b>Declaraciones</b>
<b>Programa</b>

La cabecera del programa únicamente lleva el nombre del programa. En la sección de declaraciones se le indica al compilador todos los identificadores y unidades que se utilizarán durante la ejecución del programa. En la sección del programa se escribe el código de instrucciones que se llevarán a cabo.

### Nociones básicas: variables, tipos y expresiones

Una variable es un contenido de memoria que contiene un valor que podemos cambiar; es decir, que varía. Una variable tiene un nombre (fijo y único) y un valor (variable durante la ejecución del algoritmo).



Las expresiones matemáticas contienen los operadores conocidos, constantes y funciones matemáticas. Por ejemplo:

$$\sqrt{x-1}, 2+16+18, \sin(2)*\cos(x)$$

Una expresión lógica contiene expresiones matemáticas, operadores de comparación y operadores lógicos.

- Los operadores de comparación son: =, ≠, >, <, ≤, ≥.
- Los operadores lógicos son: AND, OR, NOT.

Ejemplo

$$x-1=4, \sqrt{x+16} \neq 5, \sin(x)*\cos(x) \geq 0.2, x < y \text{ AND } y < z$$

*En este caso, las primeras tres expresiones contienen únicamente un operador de comparación, mientras que la última expresión es la traducción de la expresión matemática:  $x < y < z$ .*

Una variable contiene valor de entrada o de salida (resultados) o cálculos intermedios.

Para cambiar o dar un valor a una variable, se utiliza una lectura o una asignación. La lectura de una variable se realiza de la siguiente forma:

Lectura (variable)

El funcionamiento de una variable es conforme al siguiente orden:

1. El usuario del código entrega un valor de buen tipo y este valor se guarda en *variable*.

La escritura de un contenido se hace de una manera muy simple:

Escritura(variable)

Escritura(constante)

El funcionamiento es evidente; el valor de la variable o de la constante se pone en la salida del pseudocódigo.

La principal función del pseudocódigo consiste en decir qué debe hacer el código y cómo; por esta razón, la escritura es muy simple, sin indicar el formato de salida, y con un cierto número de cifras después del punto decimal; por ejemplo, con una política de caracteres o determinando en qué lugar de la pantalla se escribe.





## Estructura general del pseudocódigo

Un pseudocódigo se escribe para dar las grandes líneas del cálculo; su objetivo es compartir con los demás programadores su visión de la resolución del problema. Hay dos principios en la escritura de un pseudocódigo:

- Al inicio se escriben todas las variables que se usan en pseudocódigo; cada una con su nombre y su tipo.

Las líneas del pseudocódigo que siguen son órdenes (instrucciones o estructuras) que se ejecutan de arriba hacia abajo; primero una orden y después otra, así sucesivamente.

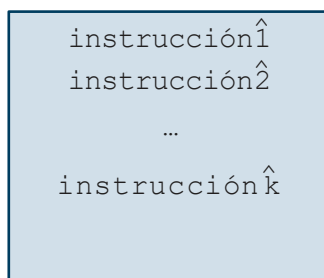
El pseudocódigo que se utiliza para la descripción de un algoritmo o para indicar los pasos de resolución de un problema contiene estructuras de control, las cuales se utilizan para describir las instrucciones de los algoritmos. Hay cuatro tipos de estructuras:

- Secuencial
- Selectiva
- Iterativa
- Anidamiento

## Estructuras componentes del pseudocódigo

### Estructura secuencial

La estructura de control secuencial tiene la siguiente forma:



### Estructura selectiva

Las estructuras selectivas permiten expresar las elecciones que se hacen durante la resolución del problema. Hay varios tipos de estructuras selectivas:



- Selectiva simple.
- Selectiva doble (alternativa).
- Selectiva múltiple.
- Selectiva casos (múltiple).

La estructura selectiva simple es de la siguiente forma:

```
si expresión lógica entonces  
instrucciones fin si
```

En esta estructura, primero se hace el cálculo de la expresión lógica; si el valor de esta expresión es cierto (no falso) se ejecutan las *instrucciones* (puede ser una sola o más de una). Si el valor de la expresión lógica es falso, no se ejecuta nada.

Las palabras *si*, *entonces* y *fin si*, son palabras clave que permiten estructurar y dar un sentido a las instrucciones. Por otro lado, es posible escribir la estructura anterior como: *si expresión lógica entonces instrucciones*

*fin si*

Esta escritura no es tan clara. ¿Dónde inicia y dónde termina la estructura *si*? ¿dónde empiezan las instrucciones que se ejecutan cuando la expresión es cierta? Por estas razones, es mejor que las partes que componen una estructura (en nuestro caso selectiva, aunque también aplica para todas las estructuras selectivas e iterativas) se escriban con algunos espacios y que la parte *si* se alinee con la parte *fin si*.

La estructura selectiva alternativa es de esta forma:

Primero, se hace el cálculo de la expresión lógica. Si el valor de esta expresión es cierto (no falso) se ejecutan las *instrucciones*<sub>1</sub>. Si no, se ejecutan las *instrucciones*.

```
si expresión lógica entonces  
    instrucciones1  
    si no  
        instrucciones2  
    fin si
```



La estructura selectiva múltiple es usada para anidar condiciones lógicas mutuamente excluyentes. Su forma es la siguiente:

```
si expresión lógica1 entonces
    instrucciones1
sino si expresión lógica2 entonces
    instrucciones2
sino si expresión lógica3 entonces
    instrucciones3
...
sino
    instruccionesn
fin si
```

Esta estructura se ejecuta de la siguiente manera:

Se hace el cálculo de la expresión lógica<sub>1</sub>, si el resultado es *cierto* se ejecutan instrucciones y la instrucción selectiva se termina. Si no, se hace el cálculo de la *expresión* lógica; si el resultado es *cierto* se ejecuten *instrucciones*<sub>2</sub> y la instrucción selectiva se termina... Si todas las expresiones lógicas son *falso*, entonces se ejecutan *instrucciones*.

Ejemplo

*Resolver la ecuación de primer grado que tiene su forma matemática más general:  $ax + b = 0$*

*La entrada está formada por los dos parámetros de la ecuación, a y b, que serán guardados en dos variables de tipo punto flotante. La salida será un mensaje sobre la raíz de la ecuación y, en algún caso, su valor.*

*Los casos que pueden aparecer y que deben tratarse de manera diferente serán:*

- *a = 0 y b = 0: cualquier número real es una solución.*
- *A = 0 y b ≠ 0: no hay ninguna solución.*
- *a ≠ 0: la raíz es única y de valor  $-a/b$ .*



*Una proposición de pseudocódigo es la siguiente:*

Real a, b

```
Real x
Lectura (a)
Lectura (b)
si a ≠ 0 entonces
    Escritura ("Hay una única raíz")
     $x \leftarrow b/a$ 
    Escritura (x)
sino si b ≠ 0 entonces
    Escritura ("No hay ninguna raíz")
sino
    Escritura ("Hay una infinidad de raíces")
fin si
```

*Este pseudocódigo contiene una estructura selectiva múltiple con condiciones lógicas exclusivas. La resolución se puede hacer con un pseudocódigo que contiene dos estructuras selectivas alternativas, una añadida a la otra:*

```
Real a, b, x
Lectura (a)
Lectura (b)
si a = 0 entonces
    si b = 0 entonces
        Escritura ("Hay una infinidad de raíces")
    sino
        Escritura ("No hay ninguna raíz")
    fin si
sino
    Escritura ("Hay una única raíz")
     $x \leftarrow b/a$ 
    Escritura (x)
fin si
```

*Las dos soluciones propuestas son semánticamente equivalentes, por lo que se realiza el mismo tratamiento; depende únicamente de la manera de escribir y leer el código y de que los participantes en el desarrollo de la solución del problema prefieran una u otra.*

*En la segunda versión, en lugar de dos declaraciones de variables, también tenemos una sola para las tres variables del código; sin embargo, la primera versión permite nada más una separación "lógica" entre las variables de entrada y la única variable de salida.*





La estructura selectiva múltiple casos se usa cuando un mismo valor se compara con varios valores. Su forma es la siguiente:

```
seleccionar expresión  
caso valor1  
instrucciones1  
  
caso valor2  
instrucciones2  
  
...  
  
en otro caso
```

La *expresión* puede ser una sola variable. Primero, se obtiene el valor de esta expresión y se compara con cada *valor<sub>i</sub>*; si hay expresión = *valor<sub>i</sub>*, se ejecutan las *instrucciones<sub>i</sub>*. Si ningún valor corresponde, se ejecuta la parte '*en otro caso*', *instrucciones*.

### Estructura iterativa

Las estructuras iterativas abren la posibilidad de ejecutar un grupo de instrucciones más de una vez; es decir, sirven para ejecutar varias veces una misma parte de un código. Hay varios tipos de estas:

- Bucle mientras
- Bucle repetir
- Bucle para (cada)

La estructura iterativa mientras (while) tiene la siguiente forma:

```
mientras expresión lógica hacer  
    instrucciones  
fin mientras
```

Su ejecución es la siguiente:

Se calcula la *expresión lógica* y, si su valor es cierto, se ejecutan las *instrucciones* y se hace un nuevo cálculo de la *expresión lógica*. Entonces, en total, las *instrucciones* se ejecutan 0 o varias veces, dependiendo del valor de la *expresión lógica*.

cuidado! Al interior de las instrucciones hay que hacer modificaciones de las



variables que componen la *expresión lógica*, porque la expresión siempre sería cierta y la estructura no se terminaría nunca.

La estructura iterativa repetir (repeat) tiene la siguiente forma:

Su ejecución es la siguiente:

```
repetir
instrucciones
hasta que expresión lógica
```

Se ejecutan las *instrucciones* y se hace el cálculo de la *expresión lógica*. Si su valor es falso, se ejecutan de nuevo las

*instrucciones* y se hace un nuevo cálculo de la *expresión lógica*.

En resumen, las *instrucciones* se ejecutan una o más veces, dependiendo del valor de la *expresión lógica*.

Una estructura *repetir* es equivalente a:

```
instrucciones
mientras NOT (expresión
lógica) instrucciones
```

*El algoritmo de transformación de un número en base 10 en otra base se aborda en el apéndice 1; la idea es hacer multiplicaciones con la base  $b$ , tomando después la parte entera de cada multiplicación. Así, el pseudocódigo:*

```
Real x
Integer b, k
Real y, p
Integer i, c
Lectura (x)
Lectura (b)
Lectura (k)
y ← x
i ← 0
repetir
    p ← y*b
    c ← parte_entera(p)
    y ← parte_fraccionaria(p)
    Escritura (c)
    i ← i + 1
hasta que i ≥ k
```



*Este pseudocódigo reviste interés por varias razones:*

- *Hasta ahora las salidas de los algoritmos se han colocado al final de todos los cálculos; desde que se obtiene una cifra de la representación fraccionaria, esta cifra contenida en la variable  $c$ , se escribe y a cada paso se calcula un nuevo valor en la iteración siguiente.*
- *Se ejecutan exactamente  $k$  iteraciones; la variable  $i$  indica, de manera muy precisa, cuántas iteraciones se ejecutaron.*
- *La manera de trabajar con la variable  $i$  no es única, también se puede escribir así:*

...

```
Real x
Integer b, k
Real y, p
Integer i, c
Lectura (x)
Lectura (b)
Lectura (k)
y ← x
i ← 0
repetir
    p ← y*b
    c ← parte_entera(p)
    y ← parte_fraccionaria(p)
    Escritura (c)
    i ← i + 1
hasta que i ≥ k
```

*El sentido de la variable  $i$  es ahora: se está ejecutando la iteración número  $i$ .*

- *Se requiere el cálculo de las partes entera y fraccionaria; entonces, se indica con claridad la “llamada” de funciones, que normalmente están implementadas en casi todos los lenguajes de programación.*
- *La variable  $p$  es intermediaria y sirve para hacer una sola vez la multiplicación de  $y$  por  $b$ . El uso de esta variable no es obligatorio, se puede escribir directamente y en el siguiente orden:*



$c \leftarrow \text{parte entera}(y*b)$

$y \leftarrow \text{parte fraccionaria}(y*b)$

*Nota: Nada más que la misma multiplicación se hace dos veces.*

- Otra variable intermediaria es  $y$ , que guarda los valores intermedios por el cálculo de la representación de  $x$ ; al inicio, esta variable toma el valor de  $x$ . Es posible usar directamente  $x$  en los cálculos; pero, al final de la estructura repetitiva, el valor de  $x$  se desnaturaliza y el valor inicial se pierde.

Una estructura iterativa que toma en cuenta la noción de variable-contador es la estructura iterativa para (for), la cual tiene la siguiente forma:

```
para i de inicio hasta fin [paso p] hacer
    instrucciones
fin para
```

Donde  $i$  es una variable (simple) e *inicio*, *fin*,  $p$ , son valores numéricos. Si el paso no es declarado, su valor es 1.

Su ejecución es repetitiva y su funcionamiento es el siguiente:

La  $i$  recibe el valor *inicio* y se ejecutan las *instrucciones*; luego,  $i$  se incrementa el valor de  $p$  (el paso) y se re ejecutan las *instrucciones*, si el valor de  $i$  es menor que *fin*.

En resumen, las *instrucciones* se ejecutan 0 o varias veces, dependiendo de los valores de *inicio*, *fin* y el paso. El valor de la variable  $i$  se puede usar al interior de las *instrucciones*, pero no puede ser modificado.

Esta estructura es equivalente a:

```
i ← inicio
mientras i ≤ fin hacer
    instrucciones
    i ← i + p
fin mientras
```



## Bibliografía:

Autor	Fecha	Titulo	Fuente	
			Editor	URL
Dr. Francisco José Villazán Olivarez	2010	Manual de Informática	UMSNH- FCCA Informática 2009	
	2020			<a href="https://www.upv.es/entidades/I2T/info/890613normalc.html">https://www.upv.es/entidades/I2T/info/890613normalc.html</a>
				<a href="https://www.investigacion.frc.utn.edu.ar/labsis/publicaciones/apunte_linux/ma.html">https://www.investigacion.frc.utn.edu.ar/labsis/publicaciones/apunte_linux/ma.html</a>
				<a href="https://mailchimp.com/es/resources/different-types-of-files/">https://mailchimp.com/es/resources/different-types-of-files/</a>
				<a href="https://support.microsoft.com/es-es/office/mover-o-copiar-un-elemento-a-otra-carpeta-19768dfe-86c4-40bf-b82c-1c084b624492#:~:text=Para%20copiar%2C%20haga%20clic%20con,haga%20clic%20en%20Ctrl%2BC.&amp;text=Vaya%20a%20la%20carpeta%20a,haga%20clic%20en%20Ctrl%2BV">https://support.microsoft.com/es-es/office/mover-o-copiar-un-elemento-a-otra-carpeta-19768dfe-86c4-40bf-b82c-1c084b624492#:~:text=Para%20copiar%2C%20haga%20clic%20con,haga%20clic%20en%20Ctrl%2BC.&amp;text=Vaya%20a%20la%20carpeta%20a,haga%20clic%20en%20Ctrl%2BV</a>
	2023		Editorial Etecé. . Sistema Operativo. Recuperado de	
				<a href="https://concepto.de/sistema-operativo/lenguajes_de_programacion_1(1).pdf">https://concepto.de/sistema-operativo/lenguajes_de_programacion_1(1).pdf</a>
				<a href="https://moisesrbb.tripod.com/unidad1p.html">https://moisesrbb.tripod.com/unidad1p.html</a>
				<a href="#">Introducción a la programación Mihaela Juganaru Mathieu (PRIMERA EDICIÓN EBOOK México, 2014)</a>