



**Universidad Nacional de Misiones**  
**Facultad de Ciencias Exactas, Químicas y Naturales**

# Informática Básica

Ingreso 2026

**MÓDULO APÓSTOLES**

Analista en Sistemas de Computación  
Licenciatura en Sistemas de Información  
Profesorado Universitario en Computación



## CONTENIDO

### Contenido

<b>BIENVENIDOS</b>	<b>4</b>
PRESENTACIÓN	5
CONTENIDOS	5
OBJETIVOS	6
METODOLOGÍA	6
<b>AUTORIDADES</b>	<b>8</b>
COORDINADORES DE LAS CARRERAS EN EL MÓDULO APÓSTOLES	8
<b>NOCIONES ELEMENTALES DE INFORMÁTICA</b>	<b>9</b>
INFORMÁTICA	9
DATOS, INFORMACIÓN Y CONOCIMIENTO. CONCEPTOS GENERALES	10
¿EN QUÉ SE DIFERENCIA EL "CONOCIMIENTO" DE LOS "DATOS" Y DE LA "INFORMACIÓN"?	11
INFORMACIÓN VS DATOS	12
¿POR QUÉ SE AUTOMATIZA EL TRATAMIENTO DE LA INFORMACIÓN?	14
¿POR QUÉ LA NECESIDAD DE INFORMACIÓN?	14
LA COMPUTADORA	14
<i>Hardware y software</i>	15
CONCEPTO DE BIT Y BYTE	15
UNIDADES DE MEDIDA PARA EL ALMACENAMIENTO DE INFORMACIÓN	16
<b>SISTEMAS DE NUMERACIÓN</b>	<b>18</b>
CLASIFICACIÓN	18
BASE DE UN SISTEMA DE NUMERACIÓN	18
TEOREMA FUNDAMENTAL DE LA NUMERACIÓN	20
CONVERSIONES ENTRE SISTEMAS	21
<b>ETAPAS DE LA RESOLUCIÓN DE PROBLEMAS</b>	<b>23</b>
ETAPAS DE LA RESOLUCIÓN DE PROBLEMAS	23
<i>Entender el problema</i>	24
<i>Trazar un plan</i>	24
<i>Ejecutar el Plan</i>	25
<i>Revisar</i>	25
<i>Ejemplo: un problema y su resolución en etapas</i>	25
RESOLUCIÓN DE PROBLEMAS CON COMPUTADORA	26
<i>Algoritmo</i>	26
<i>Fases de un algoritmo en computación</i>	28
<i>Fase: Analizar el problema</i>	29
<i>Ejercitación</i>	30
<b>PROGRAMACIÓN, LÓGICA Y ESTRUCTURAS DE CONTROL</b>	<b>31</b>
PROGRAMACIÓN	31
LENGUAJE DE PROGRAMACIÓN Y TIPOS DE LENGUAJE	32
LÓGICA	32
<i>Razonamiento lógico</i>	32
<i>Proposiciones atómicas y proposiciones moleculares</i>	33
<i>Conectivos lógicos</i>	34
ALGORITMO	35



<i>Estructuras de control</i>	36
ESTRUCTURAS BÁSICAS EXPRESADAS TÉCNICAMENTE	38
<i>Las secuencias de órdenes</i>	38
<i>Las decisiones</i>	39
<i>Los ciclos</i>	41
<i>Importancia de la indentación en las estructuras de control</i>	42

## BIENVENIDOS

El presente material fue elaborado para que el **Aspirante a Ingreso en la Facultad de Ciencias Exactas, Químicas y Naturales** de la Universidad Nacional de Misiones pueda acceder al Programa de Contenidos del requisito **INFORMÁTICA BÁSICA**.

El equipo docente a cargo de la elaboración del presente material está integrado por:

- Lic. Gladis M. SEQUEIRA
- Prof. Silvia ZAJACZKOWSKI
- Laura BURZMINSKI

El equipo docente responsable del contenido de este cursillo está conformado por:

- Lic. Gladis M. SEQUEIRA
- Prof. Silvia ZAJACZKOWSKI
- Esp. Lic. Claudio BIALE
- Lic. Viviana ARENHARDT

## PRESENTACIÓN

Dentro de la ciencia informática existen áreas y especializaciones muy diversas. Para el seguimiento y estudio de cualquiera de ellas conviene asentar previamente los conceptos básicos y generales. Este módulo busca acercar al futuro estudiante los temas propios de la disciplina, con un enfoque fundamentalmente práctico que destaca la importancia de la resolución de problemas utilizando computadoras.

Temas a tratar:

- Presentación de los conceptos fundamentales de la Informática, definiciones, alcances y términos propios de la disciplina.
- Unidades de medida de la información almacenada, introducción a las características de los principales sistemas numéricos que están relacionados con la Informática y conversiones entre sistemas, como base para entender los temas propios de la disciplina.
- Introducción a la resolución de problemas utilizando computadoras mediante una aproximación a los conceptos sobre datos y operaciones con los datos, y el concepto de algoritmos. Esto permitirá introducir una metodología sencilla para la construcción de soluciones factibles de ser implementadas, posteriormente, en una computadora.

## CONTENIDOS

Informática: definición, alcances. Información. Carácter y dato. Unidades de medida de la información. La computadora: hardware y software. Programas e instrucciones. Unidades de medida de la información almacenada. Sistemas numéricos. Conversiones entre sistemas. Datos y operaciones. Evaluación de expresiones. Algoritmos: concepto, características. Estructuras de control: secuencia, selección, repetición.

## OBJETIVOS

Que el alumno ingresante a las carreras Informáticas:

- Logre una aproximación al alcance de la disciplina.
- Incorpore los términos usuales relacionados con la Informática y su herramienta principal: la computadora.
- Identifique las características de los sistemas numéricos utilizados en Informática y los mecanismos de conversión entre sistemas.
- Comprenda la importancia de expresar sus soluciones de manera precisa, ordenada y siguiendo reglas formales.
- Modele soluciones algorítmicas para problemas sencillos, combinando representaciones de datos y estructuras de control apropiadas.
- Refuerce su vocación profesional mediante la incorporación temprana de los conceptos fundamentales de la disciplina y la práctica de actividades sencillas pero esenciales como la resolución de problemas.

## METODOLOGÍA

La modalidad de cursado dependerá del momento en que se realice.

### **Dictado octubre-noviembre 2025**

Esta instancia será completamente virtual, asíncrona y autoasistida.

Esto significa que todo el material (explicaciones teóricas, videos de apoyo y guías de ejercicios) estarán publicados en el aula virtual (plataforma Moodle institucional). El estudiante podrá administrar sus horarios de acceso y realización de actividades durante las 24 hs del día, en función de su disponibilidad, dentro del plazo establecido para el cursillo. Es importante que se organice regulando los tiempos, llevando al día las lecturas y realizando las actividades propuestas.

Al ser *autoasistido* deberá leer el material y realizar las actividades previstas de manera autónoma.

La primera mitad de este módulo se acreditará online mediante dos cuestionarios independientes tipo multiple choice dentro del aula virtual. Cuando finalice cada cuestionario, verá su calificación (**Aprobado** si resolvió correctamente al menos el 60% de cada cuestionario).

La segunda mitad del módulo (algoritmia) se acreditará mediante evaluación presencial en la última semana del cursillo. Se realizará en la sede Apóstoles (Pellegrini 269). Avisaremos a través del aula virtual la fecha en que ocurrirá. Previo a la evaluación presencial ofreceremos un cuestionario autoevaluativo tipo multiple choice dentro de la plataforma con la única intención que ayude a revisar conceptos.

No habrá recuperatorios pero, si es necesario, podrá “recursar” los contenidos que corresponda durante el cursillo presencial.

El aula virtual dispone de una **Cartelera de Avisos**, donde el Coordinador/Tutor virtual de la materia podrá publicar información general del curso. Es responsabilidad del estudiante ingresar regularmente al aula y verificar si hay novedades.

### **Dictado febrero-marzo 2026**

Se prevé dictado presencial, en la sede Apóstoles.

Habitualmente se dan dos clases semanales que incluyen teoría y resolución de ejercicios. Días y horarios serán publicados oportunamente en la web de la facultad y la plataforma del aula virtual.

## AUTORIDADES

### **DECANO**

Dr. Dardo Andrea MARTÍ

### **VICE-DECANA**

Dr. Sandra Liliana GRENON

### **SECRETARIA PRIVADA**

LGU. Cecilia Andrea YULAN

### **SECRETARIA ADMINISTRATIVA**

Srta. Miriam Raquel DAVIS

### **SECRETARIA ACADÉMICA**

Mgter Gladys Graciela GARRIDO

### **SECRETARIA ACADÉMICA ADJUNTA**

Esp. Edith Graciela ABILDGAARD

### **SECRETARIO DE BIENESTAR ESTUDIANTIL**

Sr. Carlos Adrián SOTELO

### **SECRETARIA ADJUNTA DE BIENESTAR ESTUDIANTIL**

Srta. Clarisse PITTANA HENGEN

### **SEC. DE EXTENSIÓN Y VINCULACIÓN TECNOLÓGICA**

Mgter. Alice Raquel RAMBO

### **SECRETARIA DE INVESTIGACIÓN**

Dra. Claudia Marcela MENDEZ

### **SECRETARIA DE POSGRADO**

Bqca. Marina Gisel NOVOSAK

## COORDINADORES DE LAS CARRERAS EN EL MÓDULO APÓSTOLES

- **Analista en Sistemas de Computación:** Prof. Roberto Luis SUELDO, Lic. Gladis SEQUEIRA
- **Licenciatura en Sistemas:** Mgter. Sergio CABALLERO, Mgter. Martín REY
- **Profesorado Universitario en Computación:** Mgter. Prof. Fabio LEDESMA, Prof. Daniela Alejandra MARTÍNEZ

## NOCIONES ELEMENTALES DE INFORMÁTICA

### INFORMÁTICA

La palabra informática es de origen francés y está formado por la contracción de los vocablos:

#### INFORmación y autoMÁTICA

Por Informática se entiende al “Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático y racional de la información por medio de computadoras”.

En esta definición se destacan dos términos relevantes:

- Tratamiento automático: se refiere a que son las propias máquinas (computadoras) las que realizan las tareas de captura, proceso y presentación de la información.
- Tratamiento racional: se refiere a que todo el proceso está regulado a través de una secuencia de instrucciones (programa) que siguen el razonamiento humano.

Por lo tanto, la ciencia informática se encarga del tratamiento automático de la información a través de la computadora.

Para entender en qué consiste la Informática, es imprescindible conocer:

- ✓ Las características técnicas de las computadoras y sus componentes asociados (periféricos).
- ✓ Los distintos tipos de información y datos que se manejan (sistemas de representación, archivos y bases de datos).
- ✓ Los procesos y métodos aplicados a la información o datos (programas).
- ✓ Los sistemas de comunicación que permiten tratar la información a distancia y compartirla de manera confiable (redes de comunicación de datos).

Conocer con detalle estos aspectos y la relación que existe entre ellos, permitirá entender cómo se realiza el tratamiento automático de la información utilizando computadoras.

En la definición se hace mención a la utilización de “computadoras”. En realidad, el término es más amplio y debe entenderse como “sistema informático”, siendo este un conjunto de recursos destinados al tratamiento de la información.

Como se ha notado, la informática tiene como objetivo principal proporcionar información en forma automatizada, por eso es necesario tener claro qué se entiende por **información**.

## DATOS, INFORMACIÓN Y CONOCIMIENTO. CONCEPTOS GENERALES

**Datos:** Los datos son la mínima unidad expresada sobre un tema. También se pueden ver como un conjunto de valores que por sí solos son irrelevantes como apoyo a la toma de decisiones, no dicen nada sobre el porqué de las cosas y no son orientativos para la acción.

Un número telefónico o un nombre de una persona, por ejemplo, son datos que, sin un propósito, una utilidad o un contexto no sirven como base para tomar una decisión. Los datos pueden ser una colección de hechos almacenados en algún lugar físico como un papel, un dispositivo electrónico (CD, pendrive, disco duro, disco sólido...), o la mente de una persona. En este sentido las tecnologías de la información han aportado mucho a la recopilación de datos.

**Información:** La información se puede definir como un conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), son de utilidad para quién debe tomar decisiones al disminuir la incertidumbre. Los datos se pueden transformar en información añadiéndoles valor, por ejemplo:

- ubicando el contexto y para qué propósito se generaron,
- procesándolos matemática o estadísticamente,
- corrigiendo sus errores e inconsistencias o
- resumiéndolos.

Por ende, la información es capaz de cambiar la forma en que el receptor percibe algo, impactando sobre sus juicios de valor y sus comportamientos.

**Conocimiento:** El conocimiento es una mezcla de experiencia, valores, información y *know-how* que sirve como marco para la incorporación de nuevas experiencias e información, y es útil para la acción. Se origina y aplica en la mente de los conocedores.

El conocimiento se deriva de la información, así como la información se deriva de los datos. Para que la información se convierta en conocimiento es necesario realizar acciones como:

- Comparar con otros elementos.
- Predecir consecuencias.

- Buscar conexiones.
- Conversar con otros portadores de conocimiento.

### ¿EN QUÉ SE DIFERENCIA EL "CONOCIMIENTO" DE LOS "DATOS" Y DE LA "INFORMACIÓN"?

En una conversación informal, los tres términos suelen utilizarse indistintamente y esto puede llevar a una confusión en la interpretación del concepto "conocimiento". Quizás la forma más sencilla de diferenciar los términos sea pensar que los datos están localizados en el mundo y el conocimiento está localizado en agentes de cualquier tipo (personas, empresas, máquinas...), mientras que la información adopta un papel mediador entre ambos.

Basándonos en las definiciones de Davenport y Prusak (1999):



## INFORMACIÓN VS DATOS

La Informática tiene como objetivo principal proporcionar información en forma automatizada, en consecuencia, es necesario tener claro qué se entiende por información.

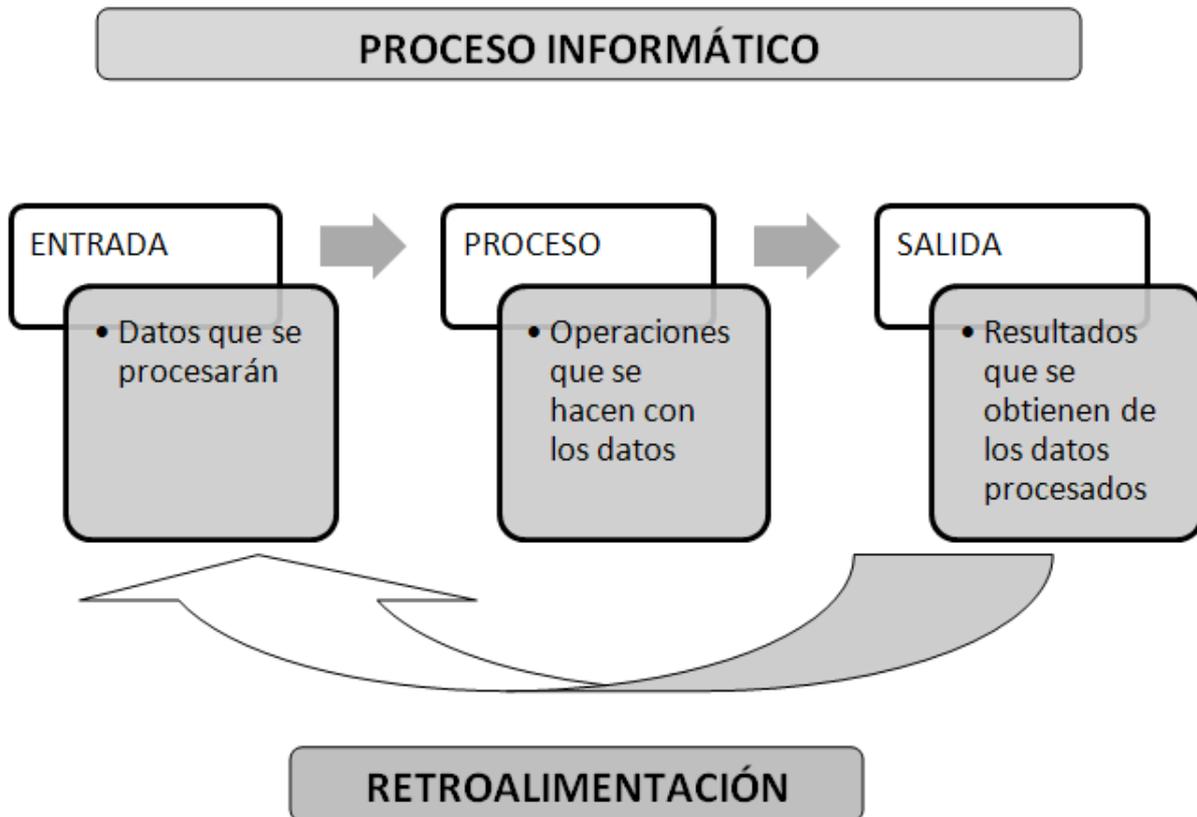
La unidad básica con la que trabajamos es definida como Dato, entendiéndose este concepto como un Carácter, una Representación o un Símbolo que se encuentra en forma aislada, sin un contexto determinado y ni un ordenamiento específico.

Esta asignación, orden o contexto está dado mediante el Conjunto de Datos que son procesados, tarea a cargo de la computadora mediante la cual se obtiene una cantidad variable de Datos Organizados, que tienen un significado y pueden ser interpretados. *Esta interpretación es entonces considerada como una Información.*

El concepto de **información** en informática abarca distintos niveles y formas de representación. En su forma más básica, la información se transmite entre dispositivos electrónicos que forman parte de una computadora. Por ejemplo, hablamos de información cuando nos referimos a los **datos almacenados** en un disco duro, una memoria USB o cualquier otro medio de almacenamiento. También consideramos información a los **datos que se envían o reciben** entre computadoras conectadas en una red. Hoy en día, la informática también incluye como información todo lo que se transmite a través de **Internet**, conocida como la "red de redes". Esto implica estudiar cómo se envía, recibe, analiza y gestiona la información en entornos digitales globales.

Entonces el concepto de Información está aplicado no solo a lo que es el envío y recepción de datos dentro de un sistema cerrado, sino también a lo que respecta a la Comunicación Entre Equipos, como también qué es lo que comparten los usuarios con el mundo, siempre considerándose de que Datos Aislados no constituyen nada por sí mismos, sino que requieren de un debido Procesamiento para poder ser contextualizados y transformados en Información.

De esta manera, se puede entender que las computadoras tratan solo con la parte física de los símbolos: la *representación de la información*. Por ello, cuando se dice que una computadora procesa información se debe entender que manipula las cadenas de símbolos que portan la información. Es decir, la computadora recibe como entrada ciertos datos de un problema y proporcionará como producto cierta secuencia de símbolos, que, convenientemente interpretada, dará la solución (información) del problema.



**Pequeño resumen:**

En el contexto genérico de procesamiento de información, se denomina **dato** a la información de entrada e información a la salida o resultado del procesamiento.

Un **dato** es una representación de un objeto del mundo real mediante la cual se puede modelar aspectos de un problema que se desea resolver con un programa en una computadora.

A su vez, un dato está formado por uno o varios caracteres. Un **carácter** es cualquier símbolo numérico, alfabético o especial que se emplea en la escritura y en el cálculo, tales como:

Numéricos: 0,1,2,... 9

Alfabéticos: a, b, c,..., z, A, B, C,... Z

Especiales: \*, /, +, #,...

De control: salto de línea, fin de archivo

## ¿POR QUÉ SE AUTOMATIZA EL TRATAMIENTO DE LA INFORMACIÓN?

Las principales razones que han llevado a la automatización del tratamiento de la información son:

- La sustitución del hombre para tareas monótonas. Ejemplo: control automático de parámetros de calidad en la fabricación de productos industriales.
- La obtención de seguridad en algunas tareas, como las que implican la repetición de una serie de pasos, en las que el hombre es más propenso a cometer errores.
- La ejecución de funciones que, aunque el hombre pueda llevarlas a cabo por sí mismo, su ejecución tardaría mucho tiempo. Por ejemplo, los cálculos complejos para el seguimiento, control y proyección en el tiempo del crecimiento de la cota de agua en una represa y su producción energética.
- La realización de funciones que el hombre por sí solo no puede llevar a cabo. Por ejemplo, comunicaciones a larga distancia y de manera inmediata.

## ¿POR QUÉ LA NECESIDAD DE INFORMACIÓN?

El conocimiento o información obtenida en el procesamiento se utiliza para tomar decisiones con vistas a un accionar concreto. Esta es la importancia que tiene la Informática en la actualidad: permite obtener información confiable, precisa y oportuna que permitirá tomar mejores decisiones.

Esto posibilita a las empresas y organizaciones el logro eficiente de sus objetivos.

## LA COMPUTADORA

La Informática trata exclusivamente el procesamiento de datos por una computadora.

Una computadora es una máquina digital y sincrónica, con cierta capacidad de cálculo numérico y lógico, controlada por un programa almacenado y con posibilidad de comunicación con el mundo exterior.

Computador, computadora u ordenador (se consideran términos equivalentes), se define como una máquina capaz de aceptar datos de entrada, efectuar con ellos operaciones lógicas y aritméticas, y proporcionar la información resultante a través de un medio de salida; y bajo el control de un programa de instrucciones previamente almacenado en el propio computador.

Las operaciones lógicas permiten comparar valores o determinar condiciones de verdad o falsedad, y las aritméticas son análogas a las usuales en matemática (suma, multiplicación, etc.).

## Hardware y software

En un sistema informático, se encuentran dos partes fundamentales que interactúan entre sí:

- *Hardware*: el origen etimológico del término hardware lo encontramos claramente en el inglés. Y es que aquel está conformado por la unión de dos vocablos de la lengua anglosajona: hard que puede traducirse como “duro” y ware que es sinónimo de “cosas”.

La Real Academia Española define al hardware como el **conjunto de los componentes que conforman la parte material (física) de una computadora**, como el disco duro, la placa madre, y la unidad central de proceso.

- *Software*: El software es una palabra que proviene del idioma inglés, pero que gracias a la masificación de uso, ha sido aceptada por la Real Academia Española. Según la RAE, el software es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.

Se considera que el software es el **equipamiento lógico e intangible de un ordenador**. En otras palabras, el concepto de software abarca a todas las aplicaciones informáticas, como los procesadores de textos, las planillas de cálculo y los editores de imágenes.

## CONCEPTO DE BIT Y BYTE

Una computadora no entiende palabras, cifras numéricas, imágenes, notas musicales o letras del alfabeto. No puede procesar información sin dividirla en unidades más pequeñas, sólo es capaz de digerir la información que se ha dividido en **bits**.

Un bit o dígito binario, es la unidad más pequeña de información que puede procesar una computadora. Puede tener uno de dos valores: 0 ó 1. Representa la información correspondiente a la ocurrencia de un suceso de entre dos posibilidades distintas: prendido o apagado, abierto o cerrado, blanco o negro.

Si podemos imaginar el interior de una computadora como una colección de microscópicos conmutadores on/of, como si fueran lámparas eléctricas, podremos entender por qué la información se procesa bit a bit. Cada conmutador o lámpara, puede estar prendido o apagado, indicando dos posibles estados: 0 o 1, como se muestra en la siguiente figura.

Se utiliza la codificación para asignar valores a los estados: el valor numérico "0" si la lámpara está apagada y "1" cuando está encendida. Este código sencillo permite transmitir información. Por ejemplo, si establecemos que una lámpara prendida en la puerta de una habitación significa "no se puede entrar" estaremos recibiendo un mensaje perfectamente definido. Una sola lámpara permite codificar dos informaciones distintas (por ejemplo: "se puede entrar" y "no se puede entrar").

Bit 1	Bit 0
 0	 0
 0	 1
 1	 0
 1	 1

Pero si tenemos dos lámparas podríamos codificar cuatro ( $4=2^2$ ) informaciones distintas, considerando las posibles combinaciones de prendido y apagado:

- las dos lámparas están apagadas (0,0)
- la primera está apagada y la segunda encendida (0,1)
- la primera está encendida y la segunda apagada (1,0)
- la dos están encendidas (1,1)

De esta forma una computadora puede procesar fragmentos más grandes de información tratando grupos de bits como unidades lógicas. Por ejemplo, una colección de 8 bits, llamada byte, puede representar 256 mensajes diferentes ( $256=2^8$ ). Esta codificación permite representar, por ejemplo, las letras del alfabeto, los dígitos decimales y símbolos especiales.

## UNIDADES DE MEDIDA PARA EL ALMACENAMIENTO DE INFORMACIÓN

Para medir longitudes se utilizan los metros, para medir capacidad se usan los litros, para medir el peso se utilizan los gramos y el tiempo se mide en horas.

Para medir la capacidad de almacenamiento de información se utilizan los bytes.

Dentro de la computadora la información se almacena y se transmite en base a un código que sólo usa dos símbolos, el 0 y el 1, y a este código se le denomina código binario.

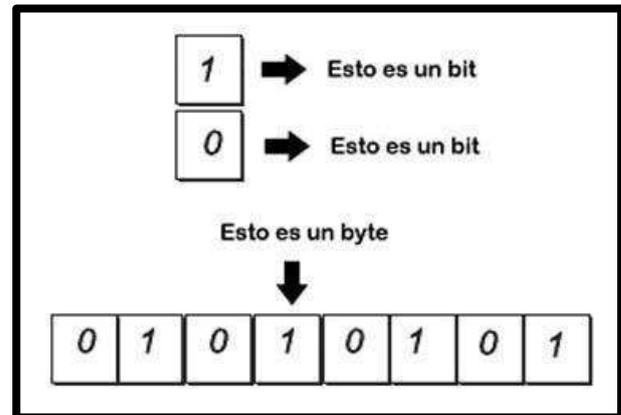
Las computadoras reducen toda la información a ceros y unos, es decir que representan todos los datos, procesos e información con el código binario, un sistema que denota todos los números con combinaciones de 2 dígitos, el 0 y el 1.

Esto significa que el potencial de la computadora se basa en solo dos estados electrónicos: encendido y apagado. Las características físicas de la computadora permiten que se combinen estos dos estados electrónicos para representar todo tipo de información.

La unidad más elemental de información es un valor binario conocido como bit (0 ó 1).

Representa la información correspondiente a la ocurrencia de un suceso de entre dos posibilidades distintas: prendido o apagado, abierto o cerrado.

En informática, cada letra, número o signo de puntuación ocupa un byte (8 bits). Por ejemplo, cuando se dice que un archivo de texto ocupa 5.000 bytes estamos afirmando que éste equivale a 5.000 letras o caracteres.



Un byte es el número de bits necesarios para almacenar un carácter.

Este número depende del código utilizado por la computadora, siendo generalmente 8, por lo que habitualmente byte se utiliza como sinónimo de 8 bits.

La capacidad de almacenamiento de una computadora o de un soporte de información, tal como el disco rígido o un pendrive, se mide en bytes. Pero como el byte es una unidad relativamente pequeña, es usual utilizar múltiplos (observar la siguiente tabla).

Las denominaciones de las unidades derivan del Sistema Internacional de Unidades.

Unidades de información (del byte)			
Sistema Internacional (Decimal)		ISO/IEC 80000-13 (Binario)	
Múltiplo – (Símbolo)	SI	Múltiplo – (Símbolo)	ISO/IEC
Kilobyte (KB)	10 <sup>3</sup>	Kibibyte (KiB)	2 <sup>10</sup>
Megabyte (MB)	10 <sup>6</sup>	Mebibyte (MiB)	2 <sup>20</sup>
Gibabyte (GB)	10 <sup>9</sup>	Gibibyte (GiB)	2 <sup>30</sup>
Terabyte (TB)	10 <sup>12</sup>	Tebibyte (TiB)	2 <sup>40</sup>
Petabyte (PB)	10 <sup>15</sup>	Pebibyte (PiB)	2 <sup>50</sup>
Exabyte (EB)	10 <sup>18</sup>	Exbibyte (EiB)	2 <sup>60</sup>
Zettabyte (ZB)	10 <sup>21</sup>	Zebibyte (ZiB)	2 <sup>70</sup>
Yottabyte (YB)	10 <sup>24</sup>	Yobibyte (YiB)	2 <sup>80</sup>

## SISTEMAS DE NUMERACIÓN

Un sistema de numeración es un conjunto de símbolos y reglas que permiten representar datos numéricos.

### CLASIFICACIÓN

Los sistemas de numeración pueden clasificarse en dos grandes grupos: posicionales y no posicionales:

- En los sistemas no posicionales los dígitos tienen el valor del símbolo utilizado, que no depende de la posición que ocupan en el número. Un ejemplo clásico es el sistema de numeración egipcio.
- En los sistemas de numeración posicionales el valor de un dígito depende tanto del símbolo utilizado, como de la posición que éste símbolo ocupa en el número, es decir que se caracterizan porque un símbolo tiene distinto valor según la posición que ocupa en la cifra.

Desde hace mucho tiempo, el hombre en su vida diaria se expresa, comunica, almacena información, la manipula, etc. mediante letras y números. Para la representación numérica utiliza el sistema de representación decimal, en tanto que, dependiendo del idioma, dispone de un alfabeto que representa estas letras. Siguiendo el mismo principio que guía al hombre, las computadoras tienen su propio sistema de representación. Debido a su construcción basada fundamentalmente en circuitos electrónicos digitales, utiliza un sistema binario. Esto obliga a transformar la representación de nuestra información, tanto numérica como alfanumérica, a una representación binaria para que la máquina sea capaz de procesarlos.

Para poder comprender mejor esto, necesitamos profundizar acerca de algunos sistemas de numeración y manejar el siguiente concepto:

### BASE DE UN SISTEMA DE NUMERACIÓN

Cuando los hombres empezaron a contar usaron los dedos, guijarros, marcas en bastones, nudos en una cuerda y algunas otras formas para ir pasando de un número al siguiente. A medida que la cantidad crece se hace necesario un sistema de representación más práctico.

En diferentes partes del mundo y en distintas épocas se llegó a la misma solución, cuando se alcanza un determinado número se hace una marca distinta que los representa a todos ellos. Este número es la base. Se siguen añadiendo unidades hasta que se vuelve a alcanzar por segunda vez el número

anterior y se añade otra marca de la segunda clase. Cuando se alcanza un número determinado (que puede ser diferente del anterior constituyendo la base auxiliar) de estas unidades de segundo orden, las decenas en caso de base 10, se añade una de tercer orden y así sucesivamente.

La base que más se ha utilizado a lo largo de la Historia es 10 según todas las apariencias por ser ese el número de dedos con los que contamos. Hay alguna excepción notable como son la numeración babilónica que usaba 10 y 60 como bases y la numeración maya que usaba 20 y 5 aunque con alguna irregularidad.

Desde hace 5000 años la gran mayoría de las civilizaciones han contado en unidades, decenas, centenas, millares etc. es decir de la misma forma que seguimos haciéndolo hoy. Sin embargo la forma de escribir los números ha sido muy diversa y muchos pueblos han visto impedido su avance científico por no disponer de un sistema eficaz que permitiese el cálculo.

Casi todos los sistemas utilizados representan con exactitud los números enteros, aunque en algunos pueden confundirse unos números con otros, pero muchos de ellos no son capaces de representar grandes cantidades, y otros requieren tal cantidad de símbolos que los hace poco prácticos.

Pero sobre todo no permiten en general efectuar operaciones tan sencillas como la multiplicación, requiriendo procedimientos muy complicados que sólo estaban al alcance de unos pocos iniciados. De hecho, cuando se empezó a utilizar en Europa el sistema de numeración actual, los abaquistas, los profesionales del cálculo se opusieron con las más peregrinas razones, entre ellas la de que siendo el cálculo algo complicado en sí mismo, tendría que ser un método diabólico aquel que permitiese efectuar las operaciones de forma tan sencilla.

El sistema actual fue inventado por los indios y transmitido a Europa por los árabes. Del origen indio del sistema hay pruebas documentales más que suficientes, entre ellas la opinión de Leonardo de Pisa (Fibonacci) que fue uno de los introductores del nuevo sistema en la Europa de 1200. El gran mérito fue la introducción del concepto y símbolo del cero, lo que permite un sistema en el que sólo diez símbolos puedan representar cualquier número por grande que sea y simplificar la forma de efectuar las operaciones.

*Resumiendo: la base de un sistema de numeración es la cantidad de símbolos que utiliza ese sistema para representar las cantidades.*

Todos los sistemas de numeración que necesitamos comprender tienen una propiedad posicional. Esto quiere decir que el dígito o elemento, además de tener un valor absoluto, tiene un valor relativo a la posición que ocupa dentro del número.

A continuación, detallaremos el sistema decimal, binario y hexadecimal.

- Decimal o base 10: es el sistema que ha usado el hombre para contar desde hace bastante tiempo, adoptado por contar con los diez dedos de la mano. Tiene 10 elementos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Binario o base 2: es el sistema de numeración que utiliza internamente el hardware de las computadoras. Tiene 2 elementos: 0, 1
- Hexadecimal o base 16: al igual que los anteriores, el sistema hexadecimal es un sistema posicional pero que utiliza dieciséis símbolos para la representación de cantidades.

Tiene 16 elementos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A la hora de hacer los cálculos los valores de las letras en este sistema tienen un valor numérico: A=10, B=11, C=12, D=13, E=14, F=15

## TEOREMA FUNDAMENTAL DE LA NUMERACIÓN

Se trata de un teorema que relaciona una cantidad expresada en cualquier sistema de numeración posicional con la misma cantidad expresada en el sistema decimal. Supongamos una cantidad expresada en un sistema cuya base es **B** y representamos por **xi** cada uno de los dígitos que contiene dicha cantidad, donde el subíndice **i** indica la posición del dígito con respecto a la coma fraccionaria, la posición se numera en forma creciente hacia la izquierda y decreciente hacia la derecha de la coma (posición 0), en ambos casos de a 1. El Teorema Fundamental de la Numeración dice que el valor decimal de una cantidad expresada en otro sistema de numeración, está dado por la fórmula:

$$\sum_N^D (\text{Dígito})_i \times (\text{Base})^l$$

Donde:

- Base= 10
- $l$  = Posición respecto a la coma,
- $D$  = n° de dígitos a la derecha de la coma,
- $N$  = n° de dígitos a la izquierda de la coma -1,
- Dígito= cada uno de los que componen el número.

Ejemplo:

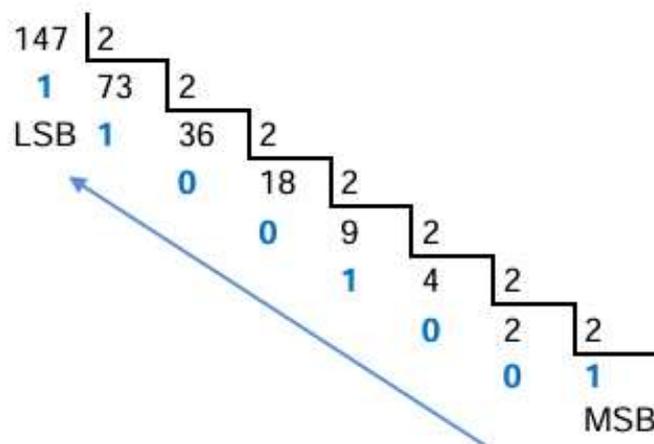
$$17.825 = 1 \times 10^4 + 7 \times 10^3 + 8 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$
$$10.000 + 7.000 + 800 + 20 + 5$$
$$17.825$$

## CONVERSIONES ENTRE SISTEMAS

Por el momento solo trabajaremos con la conversión entre los sistemas decimal y binario.

**Decimal a binario:** Se va dividiendo la cantidad decimal por 2, apuntando los residuos, hasta obtener un cociente menor al divisor. El último cociente obtenido es el bit más significativo (MSB) y el primer residuo es el bit menos significativo (LSB).

Por ejemplo, para convertir el número decimal 147 a binario, utilizando el método de divisiones sucesivas, procedemos de la siguiente manera:



Entonces:

$$153_{10} = 10010011_2$$

¿Te animás a elegir un número decimal cualquiera y convertirlo al sistema binario?

**Binario a decimal:** Se aplica el Teorema Fundamental de la numeración, es sencillo, basta con desarrollar el número teniendo en cuenta el valor de cada dígito en su posición y multiplicándolo por una potencia de 2 cuyo exponente es 0 en el bit situado más a la derecha. El exponente se incrementa en una unidad según vamos avanzando posiciones hacia la izquierda. Luego se suman los productos.

*Por ejemplo, para convertir el número binario  $1010011_2$  a decimal, lo desarrollamos teniendo en cuenta el valor de cada bit:*

$$\begin{aligned} 1010011 &= (1 \cdot 2^6) + (0 \cdot 2^5) + (1 \cdot 2^4) + (0 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0) \\ 1010011 &= (1 \cdot 64) + (0 \cdot 32) + (1 \cdot 16) + (0 \cdot 8) + (0 \cdot 4) + (1 \cdot 2) + (1 \cdot 1) \\ 1010011 &= 64 + 0 + 16 + 0 + 0 + 2 + 1 \\ 1010011 &= \mathbf{83} \end{aligned}$$

Entonces:

$$1010011_2 = 83_{10}$$

¿Y si ahora convertís a decimal el número que obtuviste en la conversión anterior?

## ETAPAS DE LA RESOLUCIÓN DE PROBLEMAS

La resolución de problemas, utilizando como herramienta una computadora, requiere contar con la suficiente capacidad de expresión para indicar a la máquina lo que debe llevarse a cabo.

***La Informática es la ciencia que estudia el análisis y resolución de problemas utilizando computadoras.***

Si se busca en el diccionario la definición de la palabra "problema", podrá hallarse alguna de las siguientes:

- Cuestión o proposición dudosa, que se trata de aclarar o resolver.
- Enunciado encaminado a averiguar el modo de obtener un resultado cuando se conocen ciertos datos.

La resolución de problemas mediante una computadora consiste en dar una adecuada formulación de pasos precisos a seguir.

Si se piensa en la forma en que una persona indica a otra como resolver un problema, se verá que habitualmente se utiliza un lenguaje común y corriente para realizar la explicación. Cuando se trata de una computadora, no pueden utilizarse indicaciones ambiguas. Ante cada orden resulta fundamental tener una única interpretación de lo que hay que realizar. Una máquina no posee la capacidad de decisión del ser humano para resolver situaciones no previstas.

Además, para poder indicar a la computadora las órdenes que debe realizar es necesario previamente entender exactamente lo que se quiere hacer. Es fundamental conocer con qué información se cuenta y qué tipo de transformación se quiere hacer sobre ella.

## ETAPAS DE LA RESOLUCIÓN DE PROBLEMAS

Para entender cómo resolver un problema debemos entender el siguiente esquema, elaborado por George Polya, matemático húngaro, quien invirtió un esfuerzo considerable en intentar caracterizar los métodos generales que usa la gente para resolver problemas, y para describir cómo debería enseñarse y aprender la manera de resolver problemas.



### **Entender el problema**

Básicamente es poner a prueba nuestra comprensión de lectura (también puede ser oral) del problema.

Debemos seguir estos pasos:

1. Leer y re leer el problema.
2. Entender la pregunta, es decir, tener claro cuál es el resultado esperado.
3. Identificar los datos importantes.
4. Organizar y clasificar los datos e información.
5. Realizar un esquema o figura.

### **Trazar un plan**

Esto quiere decir qué acciones debemos hacer con los datos y verificar nuestros datos, por lo que debemos tener presente estas preguntas:

- ✓ ¿Qué operaciones (acciones) necesito?
- ✓ ¿Qué datos que poseo no son importantes?
- ✓ ¿Será mejor descomponer el problema en otros más pequeños?
- ✓ ¿Tengo más alternativas?

## Ejecutar el Plan

Ahora que entendemos el problema y hemos elegido nuestras operaciones debemos ejecutarlo, esto quiere decir seguir paso a paso nuestra traza (configuración) y verificar si vamos llegando al resultado esperado.



Debemos ejecutar las operaciones y preguntarnos ¿vamos por el camino correcto? si es así seguimos con las siguientes operaciones y comprobar si nos acercamos a la solución.

Recuerda apoyarte con dibujos o diagramas

## Revisar

Luego de ejecutar nuestro plan, y al comprobar que hemos llegado al resultado esperado, debemos entregar una respuesta completa.

Podemos preguntarnos si existe otra forma de resolver el problema y comenzamos el ciclo de nuevo. Ver si podemos hacerlo más genérico para casos similares.

*Tener en la mente el problema porque puede servir de ayuda en un caso similar.*

## Ejemplo: un problema y su resolución en etapas

### **Problema:**

En la aprobación de un proyecto de ley presentado ante una comisión de la Cámara de Representantes de Puerto Rico, hubo 7 votos a favor de representantes del Partido Nuevo Progresista más que del Partido Popular, y el número de votos a favor del Partido Popular fue el doble de los votos a favor de los representantes del Partido Independentista. Hubo 2 representantes del partido Independentista que votaron a favor de la aprobación del proyecto. ¿Con cuántos votos a favor se aprobó el proyecto?

### **Resolución:**

- Entender el problema:** Se desea saber el número de votos a favor de la aprobación. Se conoce que del Partido Nuevo Progresista hubo 7 votos más que el del Partido Popular. Además, que el número de votos del partido Popular fue el doble de los votos del Partido Independentista y que hubo 2 votos del Partido Independentista.

- b) **Trazar un plan:** Se aplicará estrategia de trabajar de adelante hacia atrás. Primero, se utilizará el hecho que hubo 2 votos del Partido Independentista para determinar el número de votos del Partido Popular. Luego, se determinará el número de votos del Partido Nuevo Progresista. Por último, se sumará las tres cantidades.
- c) **Ejecutar el plan:** Como hubo 2 votos del partido independentista, hubo el doble o 4 votos del partido Popular. Como del Partido Nuevo Progresista hubo 7 más que del Partido Popular, en este partido hubo 11 votos.  
Por tanto, en total hubo:  $2 + 4 + 11 = 17$  votos.
- d) **Revisar:** La cantidad obtenida parece razonable.
- e) **Solución:** Hubo 17 votos a favor del proyecto.

## RESOLUCIÓN DE PROBLEMAS CON COMPUTADORA

La resolución de problemas utilizando como herramienta una computadora no se resume únicamente en la escritura de un programa, sino que análogamente al diagrama de Pólya, podemos especificar etapas bien definidas. El proceso abarca todos los aspectos que van desde interpretar las necesidades del usuario hasta verificar que la respuesta brindada es correcta.



Informáticamente, el concepto de algoritmo se presenta como un concepto clave y fundamental para la resolución de problemas con computadora.

### Algoritmo

Podemos decir que un algoritmo es un conjunto finito de reglas que dan una secuencia de operaciones para resolver un problema específico.

El término proviene del matemático persa Mohammed Al Khwarizmi que alcanzó gran repercusión por el enunciado de reglas paso a paso para sumar, restar, multiplicar y dividir números decimales.

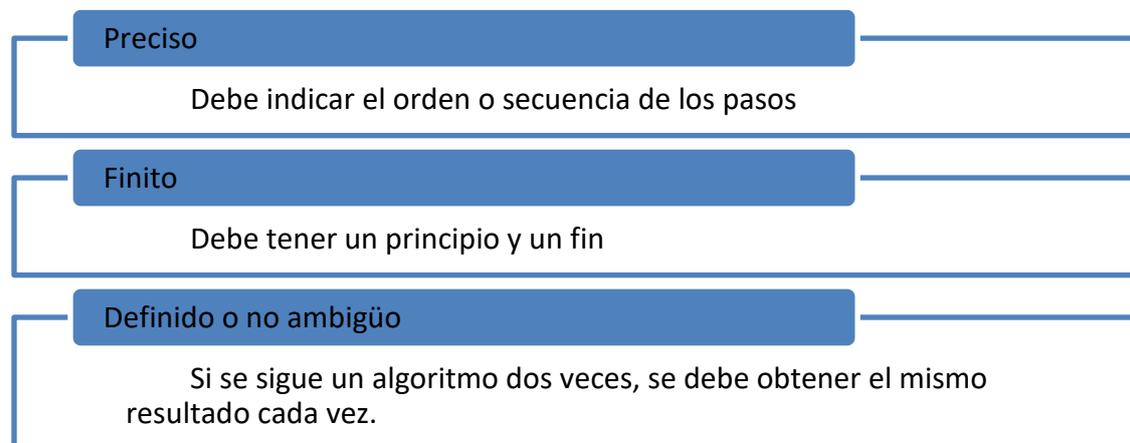
Para llegar a la realización de un programa es necesario el diseño previo de un algoritmo, de modo que sin algoritmo no puede existir un programa.

Los algoritmos son independientes tanto del lenguaje de programación<sup>1</sup> en que se expresan como de la computadora que los ejecuta. En cada problema, el algoritmo se puede expresar en un lenguaje diferente de programación o ejecutarse en una computadora distinta, sin embargo, el algoritmo será siempre el mismo. Así, por ejemplo, en una analogía con la vida diaria, una receta de un plato de cocina se puede expresar en español, inglés, portugués, pero independientemente del idioma los pasos para la elaboración del plato se realizarán sin importar el idioma del cocinero.

En las ciencias de la computación y en la programación, los algoritmos son más importantes que los lenguajes de programación o las computadoras. Un lenguaje de programación es tan solo un medio para expresar un algoritmo y una computadora es solo un procesador para ejecutarlo. Tanto el lenguaje de programación como la computadora son los medios para obtener un fin: conseguir que el algoritmo se ejecute y se efectúe el proceso correspondiente.

Ahora que vimos la importancia de los algoritmos, veamos las **características del diseño de algoritmos**.

Un buen algoritmo debe cumplir con los siguientes requerimientos:



Resumiendo, un algoritmo puede definirse como una secuencia ordenada de pasos elementales, exenta de ambigüedades, que lleva a la solución de un problema dado en un tiempo finito.

<sup>1</sup> Lenguaje de programación: se puede definir como un lenguaje para darle órdenes a la computadora y que ella las entienda y ejecute.

## Fases de un algoritmo en computación

Ahora que sabemos un poco más de cómo resolver un problema, llevemos el mismo teorema para resolver un algoritmo en computación.

Las fases serían, entonces:



**Analizar el problema:** Esta etapa sería **Entender** el problema. Aquí debemos:

- ✓ Formular el problema
- ✓ Conocer el resultado esperado
- ✓ Identificar datos e información
- ✓ Definir las operaciones
- ✓ Restricciones del problema

**Diseñar un algoritmo:** Es la representación de los pasos del algoritmo a través de pseudocódigo, que es una aproximación a un lenguaje de programación. En el mismo los pasos se convierten en instrucciones que se encuentran ordenadas en forma lógica. Además se puede usar un diagrama para ver la secuencia de las operaciones de forma lógica. Esta etapa sería **Trazar** el plan.

**Traducir el algoritmo:** Es **Ejecutar** el plan, es decir, debemos pasar nuestro algoritmo representado en pseudocódigo a un lenguaje de programación, sabiendo que cada lenguaje posee su propia gramática y sintaxis:

- ✓ Comenzar y terminar un programa: INICIO, FIN
- ✓ Declarar los tipos de los datos: entero, decimal, letra, texto.
- ✓ Entrada por teclado: leer
- ✓ Decisión: si - sino
- ✓ Iteración: mientras
- ✓ Mostrar por pantalla: imprimir

**Depurar el programa:** Esta etapa es **Revisar**. Se revisan y corrigen los errores de nuestra traducción mediante el resultado obtenido al ejecutar el programa, que debemos probar y validar. Para depurar nuestro programa

debemos probar con distintos valores posibles y seguir la secuencia de nuestro diseño y nuestra traducción, para saber si se está llegando de forma correcta a los resultados. Nos podemos ayudar haciendo una tabla para seguir la secuencia de nuestro programa y anotar los valores a medida que se vayan modificando.

**Ahora**, concentrémonos en la etapa: *Analizar el problema*. Para ayudarnos a entender esta etapa, supongamos que nos encomiendan juntar la ropa del tendal... ¡veamos!

### **Fase: Analizar el problema**

Formular el problema: juntar una remera que está en el tendal y dejar los broches en el tendal, doblar la remera y dejarla sobre la silla.

Conocer el resultado esperado: descolgar la remera y dejarla doblada en la silla.

Identificar datos e información:

- *Datos*: hay un tendal, está la remera colgada, broches y silla.
- *Información que necesitamos*: ¿Dónde está el tendal? ¿Dónde está la silla?

Definir las operaciones

1. Dirigirme al tendal
2. Elegir uno de los extremos de la remera
3. Levantar las manos
4. Con una mano sostener la remera
5. Con la otra mano quitar los broches de cada extremo de la remera
6. Llevar la mano que sostiene la remera con fuerza hacia arriba
7. Abrir la mano
8. Dejar caer la remera sobre el hombro
9. Con la otra mano colocar los broches en el tendal
10. Dirigirnos a la silla que nos fue indicada
11. Doblar la remera
12. Depositar la remera en la silla



Podemos observar, como definimos anteriormente, que el algoritmo es una secuencia ordenada de pasos elementales, exenta de ambigüedades, que lleva a la solución de un problema dado en un tiempo finito y que cualquier persona puede reproducir esa solución y llevar a cabo las acciones para resolver el problema.

*Restricciones del problema:*

- ✓ El tendal debe ser posible de alcanzar
- ✓ Saber dónde está la silla para depositar la remera

Si alguna de estas restricciones no se cumple, nuestro algoritmo no funciona.

**Desafío:** ¿y si el problema se amplía a modificar o completar el algoritmo para más de una prenda en el tendal?

### **Ejercitación**

Para los siguientes casos formular el problema, plantear un resultado esperado, Identificar datos e información y definir las operaciones



- a) Usar un buscador para acceder a la primera página web que se liste al buscar información sobre restauración de muebles de algarrobo. Vos elegís si haces la búsqueda en una computadora o en el celular.
- b) Preparar una ensalada de frutas naturales que incluya una manzana, una naranja y una pera.

## PROGRAMACIÓN, LÓGICA Y ESTRUCTURAS DE CONTROL

### PROGRAMACIÓN

Podemos empezar diciendo que programación es la elaboración de programas para la resolución de problemas mediante computadoras; y la preparación de los datos necesarios para obtener la solución de un problema a través de una computadora...

¡Pero es mucho más!!!

Es el proceso por el cual una o varias personas desarrollan un programa valiéndose de una herramienta que le permita escribir el *código fuente* (para escribir el código fuente la herramienta es un lenguaje de programación, tal como C++, Java y Python) y de otra herramienta que sea capaz de “traducirlo” a lo que se conoce como *lenguaje de máquina*, que es el lenguaje que entiende el microprocesador.

Este último paso se conoce como *compilación* y es necesario para que el código pueda ser ejecutado por la plataforma para la cual haya sido creado (una computadora, una tableta, un teléfono móvil, etc.). Existe también una forma de traducir el código denominada *interpretación*, que consiste en analizar línea a línea, hasta que se traduzca lo suficiente como para poder realizar una tarea.

Con este criterio, los lenguajes de programación se dividen en dos grandes grupos:

- Los que se compilan
- Los que se interpretan

Los programadores tienen en sus manos el poder de dar vida a un sistema, a una aplicación, a un videojuego. Es importante notar que todos estos productos pueden ser definidos con absoluto detalle en un documento, incluyendo imágenes y gráficos que expliquen cómo funciona cada milímetro de los mismos; sin embargo, hasta que un desarrollador de software no entra en acción, no es posible verlos en movimiento, ¡¡probarlos... pasar de la teoría a la práctica!!

En un plano más técnico, la programación se realiza mediante el uso de algoritmos.

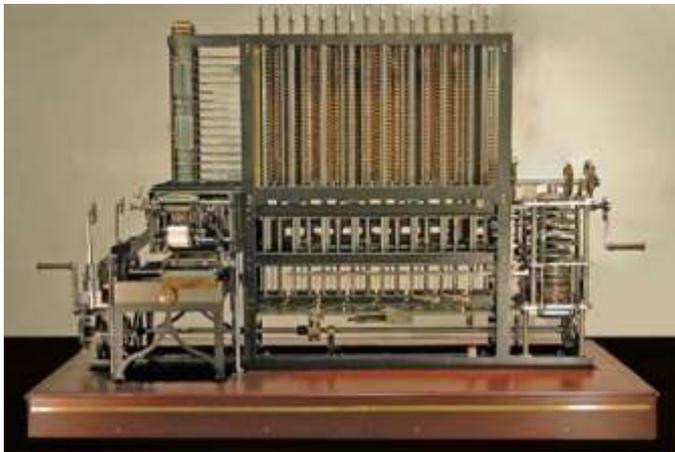
*“Un programa es una serie de instrucciones ordenadas, codificadas en lenguaje de programación que expresa un algoritmo y que puede ser ejecutado en un computador.”*

## LENGUAJE DE PROGRAMACIÓN Y TIPOS DE LENGUAJE

Ahora que tenemos un poco más claro el concepto de programación hablemos de los lenguajes de programación:

“...un lenguaje de programación es aquella estructura que, con una cierta base sintáctica y semántica, imparte distintas instrucciones a un programa de computadora.”

A la hora de establecer el origen del lenguaje de programación tenemos que hacer referencia, sin lugar a dudas, a Ada Lovelace que está considerada como la primera programadora de computadoras conocida en todo el mundo.



**“Esta máquina puede hacer cualquier cosa que sepamos cómo ordenarle que la ejecute”**

Palabras de Ada refiriéndose a la máquina analítica de Charles Babbage

*¿Cómo habrá sido esta historia?*

*¿Qué habrá en internet sobre esto?*

Un lenguaje de programación tiene la capacidad de especificar, de forma precisa, cuáles son los datos con los que debe trabajar un equipo informático, de qué modo deben ser conservados o transferidos dichos datos y qué instrucciones debe poner en marcha la computadora ante ciertas circunstancias.

## LÓGICA

### Razonamiento lógico

Razonar es la actividad mental que permite lograr la estructuración y la organización de las ideas para llegar a una conclusión.

En informática, la lógica tiene un papel muy importante ya que es una ciencia llena de estructuras, métodos y ramas, de las cuales hay una en especial que a los informáticos nos ayuda mucho: la lógica proposicional. Esta rama de las ciencias lógicas permite el razonamiento a través de un

mecanismo que primero evalúa sentencias simples y luego sentencias complejas, formadas mediante el uso de conectivos proposicionales, por ejemplo Y (AND), O (OR). Este mecanismo nos ayuda a determinar la veracidad de las cuestiones que nos planteamos.

*¿Lo decimos de una manera más simple?*

Mucho de nuestro trabajo como informáticos está relacionado con la toma de decisiones, y para tomar las decisiones correctas o decidir qué hacer ante determinados casos, utilizamos razonamientos fundamentados en la ciencia de la lógica. ¡De acá nace nuestra necesidad de estudiar un poco al respecto!

Vamos a concentrarnos en la lógica proposicional.

### **Proposiciones atómicas y proposiciones moleculares**

La lógica proposicional trata sobre la verdad o la falsedad de las proposiciones.

Una proposición es la unidad mínima de significado susceptible de ser verdadera o falsa.

¿Qué quiere decir esto? Que una palabra o una frase aislada no nos dice nada, por ejemplo, si decimos “Walter” o “martes” no estamos diciendo algo acerca de lo cual podemos afirmar su grado de verdad, es decir no podemos concluir si es VERDADERO o FALSO.

Sin embargo, si decimos “Walter es alto”, dependiendo del Walter que conozcamos podremos decir si esta afirmación es verdadera o falsa, y es eso justamente lo que la convierte en una proposición: el hecho de poder determinar su “grado de verdad”.

Debemos distinguir dos tipos de proposiciones:

- ✓ las proposiciones atómicas, y
- ✓ las proposiciones moleculares

Las proposiciones atómicas son aquellas que no se componen de otras proposiciones. La proposición “Walter es alto” es atómica porque ninguno de los elementos que la componen es una proposición.

Como podemos observar, una proposición atómica es verdadera o falsa, y su verdad o falsedad no depende de otras proposiciones, sino de cómo es la realidad.

Las proposiciones moleculares son aquellas que están compuestas por proposiciones atómicas. Un ejemplo de proposición molecular sería: “Walter es alto y juega al fútbol”

La proposición del ejemplo es molecular porque se compone de dos proposiciones atómicas: “Walter es alto” y “(Walter) juega al fútbol”. Estas dos proposiciones atómicas están conectadas mediante el conectivo *lógico* “y”.

De una proposición molecular también podemos determinar su grado de verdad, pero a diferencia de lo que ocurre con las proposiciones atómicas, su verdad o falsedad no depende directamente de la realidad, sino que depende, o es función, de la verdad o falsedad de las proposiciones atómicas que la componen.

¿Y cómo determinamos su grado de verdad entonces? ¡Fácil! Necesitamos conocer el grado de verdad de “Walter es alto” y “(Walter) juega al fútbol” por separado, y luego basarnos en reglas lógicas y operar con los conectivos lógicos.

## **Conectivos lógicos**

Los conectivos lógicos son elementos que sirven para conectar o unir entre sí proposiciones atómicas y nos ayudan a determinar el grado de verdad de las proposiciones moleculares.

Nombre	Conectivo	Símbolo
Conjunción	AND (Y)	$\wedge$
Disyunción	OR (O)	$\vee$
Negación	NOT (-)	-

Al conectivo “Y” se lo denomina conjunción. Si dos proposiciones atómicas están unidas por un conectivo lógico “Y”, el grado de verdad de la proposición molecular que éstas componen sólo será verdadero si las dos proposiciones atómicas que la componen son ambas verdaderas, y será falsa en caso de que, al menos una de ellas, sea falsa.

CONJUNCIÓN – AND (y)			
Proposición 1	Conectivo	Proposición 2	Resultado
V	$\wedge$	V	<b>V</b>
V	$\wedge$	F	<b>F</b>
F	$\wedge$	V	<b>F</b>
F	$\wedge$	F	<b>F</b>

Al conectivo “O” se lo denomina disyunción. Si dos proposiciones atómicas están unidas por un conectivo lógico “o”, el grado de verdad de la proposición molecular será verdadero si al menos una de las proposiciones atómicas que la componen es verdadera, y será falso en caso de que absolutamente todas sean falsas.

DISYUNCIÓN – OR (O)			
Proposición 1	Conectivo	Proposición 2	Resultado
V	<b>v</b>	V	<b>V</b>
V	<b>v</b>	F	<b>V</b>
F	<b>v</b>	V	<b>V</b>
F	<b>v</b>	F	<b>F</b>

Al conectivo “-” se lo denomina negación. La negación cambia el valor de verdad de una proposición de verdadero a falso, y de falso a verdadero.

NEGACIÓN - NOT (-)		
Conectivo	Proposición	Resultado
-	V	<b>F</b>
-	F	<b>V</b>

Entonces, si analizamos los posibles casos para nuestro ejemplo sería así:

Walter es alto	y	Juega al fútbol	Resultado
V	<b>^</b>	V	<b>V</b>
V	<b>^</b>	F	<b>F</b>
F	<b>^</b>	V	<b>F</b>
F	<b>^</b>	F	<b>F</b>

El grado de verdad de nuestro ejemplo de proposición molecular varía en función del grado de verdad de las proposiciones atómicas que la componen.

## ALGORITMO

En este momento, recordemos el concepto de algoritmo, visto anteriormente de este curso:

“Un algoritmo puede definirse como una secuencia ordenada de pasos elementales, exenta de ambigüedades, que lleva a la solución de un problema dado en un tiempo finito.”

Todo algoritmo debe tener las siguientes características:

- Debe ser Preciso, porque cada uno de sus pasos debe indicar de manera precisa e inequívoca qué se debe hacer.
- Debe ser Finito, porque un algoritmo debe tener un número limitado de pasos.
- Debe ser Definido, porque debe producir los mismos resultados para las mismas condiciones de entrada.

Dado que un algoritmo es una secuencia de pasos para solucionar un problema y que utilizamos algoritmos en programación... entonces, ¿estaría bien si definimos los pasos para la solución de cualquier problema que se pueda resolver por computadora y lo escribimos en un lenguaje de programación? NO. Los algoritmos primero deben representarse (existen varias técnicas para hacerlo) y probarse antes de ser implementados en un lenguaje de programación.

Una de las técnicas de representación de algoritmos más utilizadas es el **Pseudocódigo** (pseudo o seudo, permite referirse a algo que no es original, sino que es falso o una imitación). Consiste en utilizar las normas de estructura de un lenguaje de programación y lo que se obtiene puede ser leído por una persona, pero no puede ser interpretado por una computadora.

La utilización de algoritmos en pseudocódigo, además de propiciar el aprendizaje de la programación, nos facilita compartir con otros programadores la solución a un problema dejando en manos de quien vaya a implementarlo en un lenguaje la elección del mismo.

El pseudocódigo, en este sentido, está considerado como una descripción de un algoritmo que resulta independiente de todo lenguaje de programación. Para que una persona pueda leer e interpretar el código en cuestión, se excluyen diversos datos que no son clave para su entendimiento.

### **Estructuras de control**

Cuando analizamos el algoritmo para juntar una remera del tendal (¿lo recordás?), hallamos una solución factible y concluimos que realizando las acciones paso a paso en el orden en que aparecen en el algoritmo, cualquier persona podrá realizar esa tarea. Ahora bien, si quisiéramos seguir el mismo algoritmo para juntar toda la ropa del tendal sería agotador, pues deberíamos caminar mucho juntando prenda por prenda. Es decir, deberíamos realizar el algoritmo una y otra vez desde el paso 1 hasta el paso 12.

Es entonces en este momento, en que ante una generalización del problema, nos encontramos con la necesidad de conocer las estructuras de control.

Usando las estructuras de control el programador puede controlar el flujo con el que se ejecutarán las instrucciones de un algoritmo. Estas estructuras retienen y/o transfieren el control de la ejecución de las instrucciones de un programa, y permiten contemplar un sinfín de posibilidades en un solo algoritmo. La correcta conjugación de la lógica proposicional y las estructuras de control ayudan a construir algoritmos que de otra forma serían tediosos, poco seguros y seguramente muy largos, lo que complicaría sustancialmente probarlos, depurarlos y modificarlos cómodamente.

En líneas generales, el pensamiento humano (siempre en el contexto de la solución de problemas) se mueve entre tres estructuras de pensamiento básicas, veamos:

Cuando te toca cocinar, en tu mente van apareciendo poco a poco las acciones para realizar esa tarea, y lo que va organizando tu mente es una secuencia de pasos para hacer la rica comida. Esa es la primera estructura sobre la que se mueve el pensamiento humano, la

### ***Secuencia***

Primero planeas cada secuencia de acciones (consciente o inconscientemente) antes de ejecutarlas. Cada una de las cosas que hacemos diariamente no son más que secuencias de acciones que hemos planeado para poder cumplir con nuestros objetivos en la sociedad.

Salís a comprar la carne para tu receta, pasas frente a muchos negocios, pero ninguno te interesa, vas a seguir caminando hasta llegar a la carnicería en que estás acostumbrado a comprar y ahí vas a entrar porque comprar carne es tu objetivo. Encontrás dos cortes que te interesan: uno es magro, tierno y su costo es elevado; el otro es más accesible, pero tiene grasa visible y te demandaría más tiempo limpiarlo.

En cada alternativa hay ventajas y desventajas, y sin quererlo te encontrás a cada paso analizando cuál alternativa tomar para cumplir con tus objetivos. Esto pasa muchas veces en el día y todos los días. Justamente, la segunda estructura es la

### ***Decisión***

Gracias a que existe la decisión, ante cada una de las situaciones que ejemplificamos y ante cada una de las miles de situaciones que se presentan en la vida, podés escoger entre varias alternativas la mejor. La decisión se da siempre que tengas que escoger de entre, por lo menos, dos caminos lógicos.

Hasta hace muy poco estabas acostumbrado a ir todos los días a la misma hora a la escuela, a seguir el mismo camino, a saludar a las mismas personas de la misma manera, a formar en el mismo lugar en la fila. Todos los días tenés tu propia rutina respecto a levantarte, cepillarte los dientes, desayunar... sencillamente, vivís practicando la tercera estructura que son los

### ***Ciclos***

Cuando la comida que preparaste en los ejemplos anteriores esté lista (y riquísima por supuesto), vas a sentarte a la mesa a comer igual que todos los días, y de la misma manera, vas a cargar el cubierto y lo vas a llevar del plato a la boca la cantidad de veces necesaria hasta terminar tu almuerzo. Son todas tareas cíclicas.

Podemos concluir en que cualquier acción o conjunto de acciones que hagamos siempre están enmarcadas en estas tres estructuras:

- Secuencias de acciones
- Decisión de acción
- Ciclos de acciones

## **ESTRUCTURAS BÁSICAS EXPRESADAS TÉCNICAMENTE**

### **Las secuencias de órdenes**

Para utilizar la estructura de secuencia (que a veces parece ser tan obvia) todo lo que tenemos que hacer es ir colocando una acción tras otra y, por supuesto, ser muy racionales en el orden de dichas acciones porque seguramente, notaste que en cuestión de algoritmos ¡el orden de los factores sí altera el resultado! Para escribir una secuencia de órdenes o acciones todo lo que tenemos que hacer es colocar una nueva orden o una nueva acción después de la última que hayamos colocado. Es de esta manera en la que realizamos los algoritmos anteriores:

Algoritmo Secuencia

#### **Inicio**

Dirigirnos al tendal

Elegir uno de los extremos de la prenda

Levantar las manos

Con una mano sostener la prenda

Con la otra mano quitar los broches

Llevar la mano que sostiene la prenda con fuerza hacia arriba

Abrir la mano  
Dejar la prenda en el hombro  
Con la otra mano colocar los broches en el tendal  
Movernos hacia la próxima prenda  
Repetir los pasos del 3 hasta el 10 la cantidad de veces igual a la cantidad de prendas colgadas en el tendal inicialmente.  
Dirigirnos al lugar en que nos fue indicado depositar la ropa.  
Depositara la ropa

**Fin.**

**NOTESE QUE REPETIR LOS PASOS DEL 3 AL 10 IMPLICA QUE CONOZCA DE ANTEMANO LA CANTIDAD DE PRENDAS COLGADAS**

## **Las decisiones**

Siempre que tenemos que tomar una decisión o, más bien, siempre que tengamos que utilizar la estructura de Decisiones vamos a depender de una condición. La condición es la que nos permite que podamos decidir cuál es el camino lógico correcto a tomar.

Vamos a desarrollar el mismo algoritmo para juntar ropa en el tendal, pero con una mejora: nos preguntaremos siempre si hay más ropa para juntar:

Algoritmo Decisión

**Inicio**

Dirigirnos al tendal  
Elegir uno de los extremos  
Levantar las manos  
Con una mano sostener la prenda  
Con la otra mano quitar los broches  
Llevar la mano que sostiene la prenda con fuerza hacia arriba  
Abrir la mano  
Dejar la prenda en el hombro  
Con la otra mano colocar los broches en el tendal

**Si (hay otra prenda en el tendal) entonces**

Movernos hacia la próxima prenda  
Repetir los pasos del 3 hasta el 10 la cantidad de veces igual a la cantidad de prendas colgadas en el tendal inicialmente  
Dirigirnos al lugar en que nos fue indicado depositar la ropa  
Depositara la ropa

**Fin.**

¿Notaste cómo cambia el algoritmo al utilizar el condicional **SI (condición) entonces**? Se vuelve más racional y más aproximado a la realidad.

La condición es una proposición y de acuerdo al grado de verdad que tenga regula las acciones que vienen después y que dependen del Si condicional.

Una decisión completa involucra:

- Una pregunta que evalúa una condición
- Un conjunto de acciones a realizar en caso de que la condición sea verdadera
- Un conjunto de acciones a realizar en caso de que la condición sea falsa

A veces, un condicional **Si** tiene un Sino asociado a él. Siempre que exista un **Sino** es porque está asociado a un **Si** condicional determinado.

Veamos otro ejemplo. El algoritmo a continuación resuelve el siguiente problema:

Supongamos que vas por la calle y ves una heladería y se te antoja un helado. Entrás con esta idea: si el dinero que tenes te alcanza vas a tomarte un helado de crema en cucurucho, de lo contrario vas a comprar un picolé para saciar tu antojo (limitemos el enunciado suponiendo que para el picolé seguro alcanza el dinero que tenes).

Algoritmo Helado

**Inicio**

Ingresar a la heladería

Saludar a quien atiende

Preguntar el costo de una bocha de helado

Contar el dinero que hay en tu billetera

**Si** (el dinero alcanza para pagar el costo de un helado) **entonces**

**Inicio\_Si**

Elegir el gusto de helado

Pedir el helado

Agarrar el helado

**Fin\_Si**

**Sino**

**Inicio\_Sino**

Elegir el gusto de picolé

Pedir el picolé

Agarrar el picolé

**Fin\_Sino**

Pagar

Salir de la heladería

Disfrutar tu compra

**Fin.**

- ✓ Las acciones comprendidas entre el **Inicio\_Si** y el **Fin\_Si** que se encuentran dentro de la estructura **Si** son el conjunto o bloque que se debe realizar precisamente si la condición es **Verdadera**.
- ✓ Las acciones comprendidas entre el **Inicio\_Sino** y el **Fin\_Sino** que se encuentran dentro de la estructura **Sino** asociado (en caso de tenerlo) son el conjunto o bloque que se debe realizar si la condición es **Falsa**.

## Los ciclos

Si analizamos el *Algoritmo Decisión* podemos notar que después de la estructura **Si** se encuentra la instrucción “*Repetir los pasos del 3 hasta el 10 la cantidad de veces igual a la cantidad de prendas colgadas en el tendal inicialmente*” que lo que está indicando es la necesidad de repetir un determinado bloque de acciones tantas veces como sea necesario. Esto significa que, para este caso, tenemos que conocer cuántas prendas hay colgadas. Otras veces es posible que desconozcamos la cantidad exacta de veces. Este es un claro ejemplo de la necesidad de incorporar una estructura de ciclo llamada *Mientras*, cuyo formato es **Mientras (condición) hacer** y lograr el siguiente algoritmo:

Algoritmo Ciclo

### **Inicio**

Dirigirnos al tendal

Elegir uno de los extremos

### **Mientras (hay ropa en el tendal) hacer**

**Inicio\_Mientras** (esta instrucción marca el inicio del bloque de acciones que se realizarán mientras la condición sea V)

Levantar las manos

Con una mano sostener la prenda

Con la otra mano quitar los broches

Llevar la mano que sostiene la prenda con fuerza hacia arriba

Abrir la mano

Dejar la prenda en el hombro

Con la otra mano colocar los broches en el tendal

### **Si (hay otra prenda en el tendal) entonces**

Movernos hacia la próxima prenda

**Fin\_Mientras** (esta instrucción marca el fin del bloque de acciones que se realizarán mientras la condición sea V)

Dirigirnos al lugar en que nos fue indicado depositar la ropa

Depositar la ropa

### **Fin.**

La palabra *Mientras* establece, en relación con una condición, el inicio de un conjunto de acciones que se repiten, precisamente, Mientras esa condición lo permita.

- ✓ Las acciones comprendidas entre el **Inicio\_Mientras** y el **Fin\_Mientras** que se encuentran dentro de la estructura **Mientras** son el conjunto o bloque que se debe repetir (o iterar) precisamente mientras la condición sea **Verdadera**.

### **Importancia de la indentación en las estructuras de control**

Las instrucciones que pertenecen a una estructura de control deben tener una sangría mayor que la utilizada para escribir el comienzo de la estructura. De esta forma, podrá identificarse dónde termina el conjunto de instrucciones involucradas. A esta sangría se la denomina indentación.

Este concepto se aplica a las estructuras de control vistas previamente: selección y repetición o iteración.