



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



# *Análisis de algoritmos*

## **Tema 01:** El rol de los algoritmos en la Computación

M. en C. Edgardo Adrián Franco Martínez

<http://www.eafranco.com>

[edfrancom@ipn.mx](mailto:edfrancom@ipn.mx)

[@edfrancom](#) [f edgardoadrianfrancom](#)





# Contenido

- Introducción
- Historia
- Algoritmo
- Características de los algoritmos
- Importancia de los algoritmos en computación
- Introducción a la complejidad de los algoritmos
- Análisis A Priori y Prueba A Posteriori





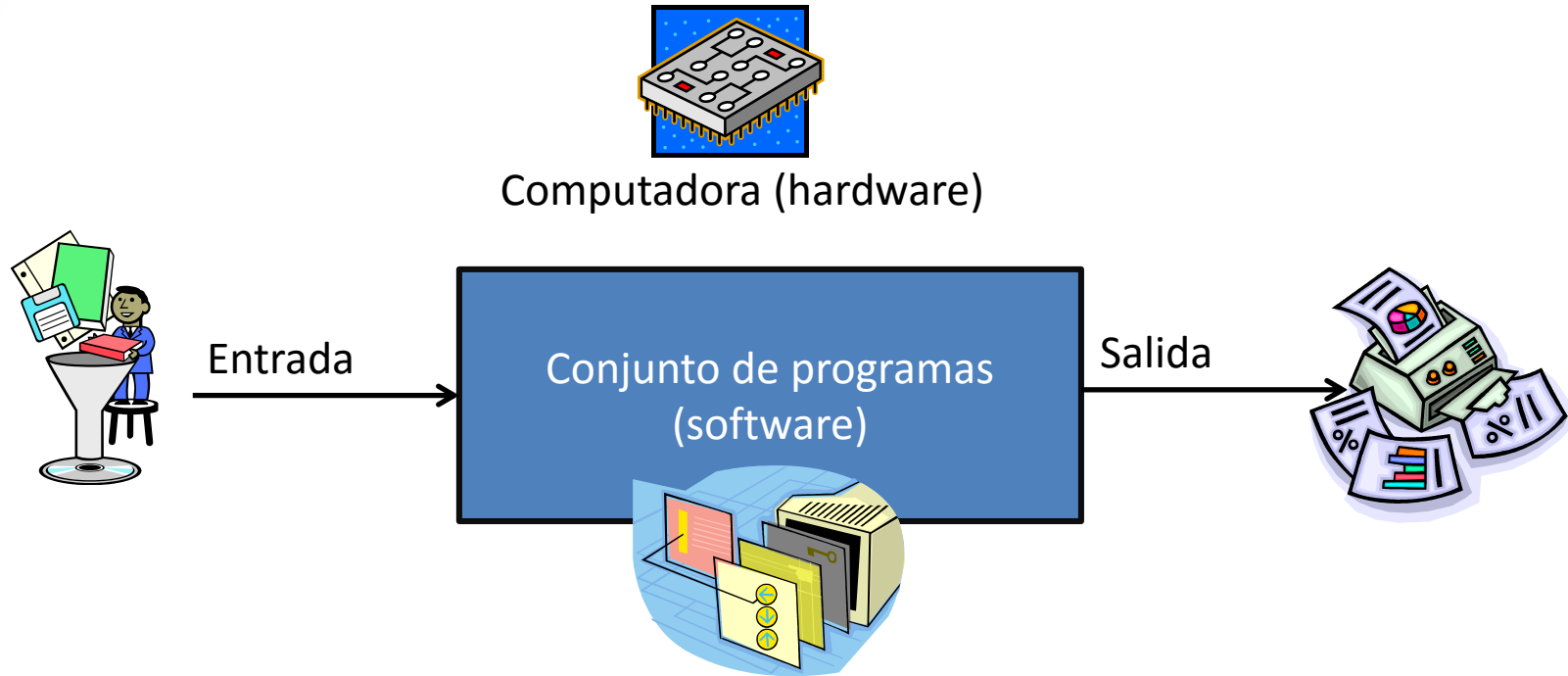
# Introducción

- Una **computadora** es una **máquina** **capaz** **de** **procesar información** digital a gran velocidad.
- Una computadora esta compuesta por un conjunto de componentes electrónicos, mecánicos e interfaces para interactuar con el exterior (usuarios u otros dispositivos) y por un conjunto de programas que determinan que operaciones llevar a cabo.
- Los datos ordenados (información) que constituyen una entrada (*input*) a la computadora se procesan mediante una lógica (programa) para producir una salida (*output*).



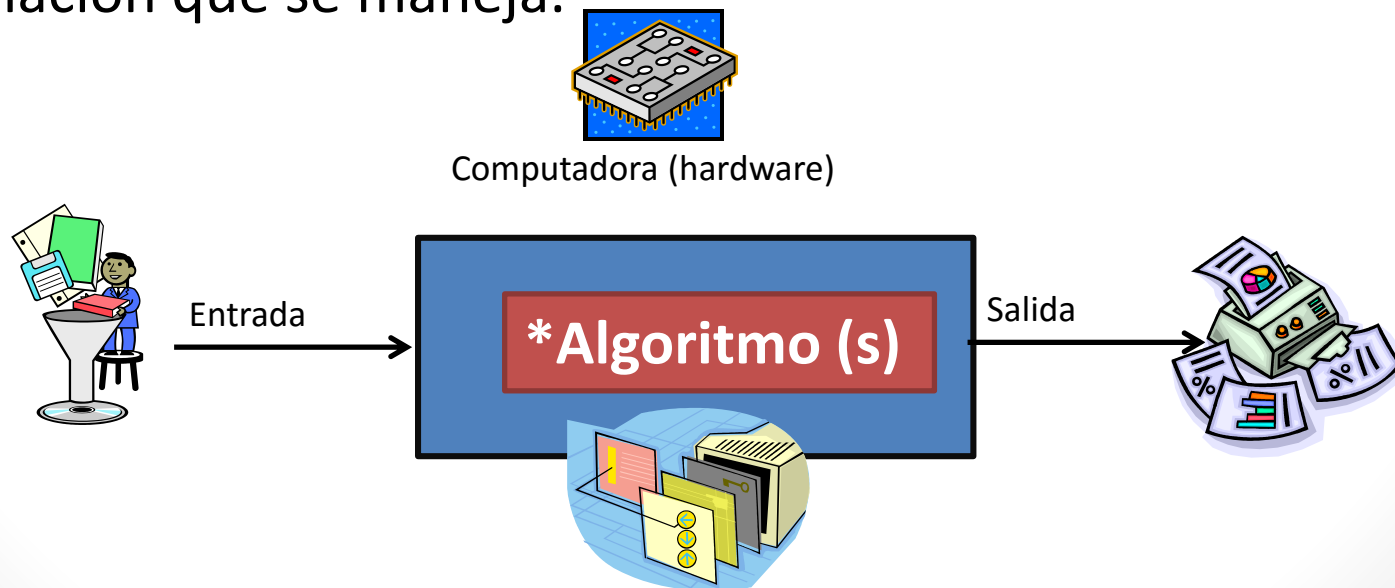


- Una computadora es una **máquina capaz de procesar información** digital a gran velocidad.



Una computadora esta **formada** por un **parte física y** otra **lógica** (hardware & software), la primera de estas esta conformada por los **elementos físicos** que la conforman (**dispositivos electrónicos y mecánicos**), la parte lógica es aquella que determina que **procesos** se van a realizar **con la información de entrada**.

- La persona responsable de indicar a la computadora la **lógica de procesamiento** recae en el que lleva a cabo la construcción del software (**programador**).
- La **razón de ser de una computadora** es poder **resolver problemas** capaces de ser **modelados** y representados en *datos coherentes y ordenados* (información), apoyándose de su **gran velocidad** y **capacidad de seguir una serie de pasos programados con anterioridad\*** y dependientes de la información que se maneja.



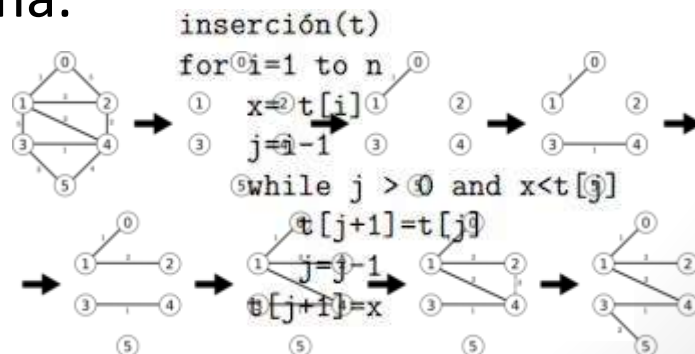
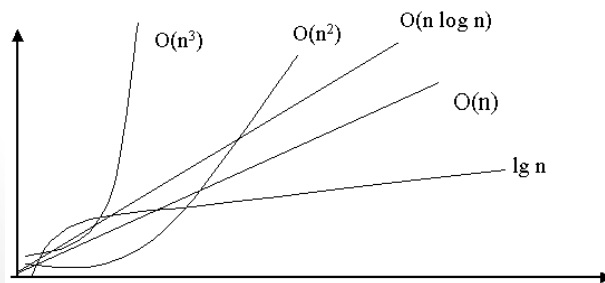


- **Algoritmo**, es un **conjunto ordenado y finito de operaciones** que permiten encontrar la solución a un problema.
- P.g. una receta de cocina, las instrucciones para armar una bicicleta, un mueble, etc.
- Los **primeros** algoritmos registrados **datan de Babilonia**, originados en las matemáticas como un método para resolver un problema usando una secuencia de cálculos más simples. Esta **palabra** tiene su **origen** en el nombre de un famoso **matemático y erudito árabe** del **siglo IX**, **Al-Khorezmi**.
- En el contexto de la computación **algoritmo** se usa para denominar a la **secuencia de pasos a seguir para resolver un problema usando una computadora**. Por esta razón, la algoritmia o ciencia de los algoritmos, es uno de los **pilares de la computación**.





- El **análisis de algoritmos** es una **parte** importante de la **Teoría de complejidad computacional**, que provee **estimaciones** teóricas para los **recursos** que necesita cualquier algoritmo **que resuelva un problema computacional dado**. Estas estimaciones resultan ser bastante útiles en la búsqueda de algoritmos eficientes.
- Los temas de mayor interés son el **análisis teórico de algoritmos** lo que permite, calcular su **complejidad en un sentido asintótico**, así como el **análisis de problemas comunes** que requieren una cantidad de procesamiento alta de los datos para poder ser resueltos con exactitud o aproximación a la respuesta óptima.





# Historia



- El término **“algoritmo”** proviene del matemático árabe **Al'Khwarizmi**, que escribió un **tratado sobre los números**. El trabajo de Al'Khwarizmi permitió preservar y difundir el conocimiento de los griegos e indios, pilares de nuestra civilización.
- Rescató de los griegos la rigurosidad y de los indios la simplicidad (*en vez de una larga demostración, usar un diagrama junto a la palabra **Mira***). Sus libros son intuitivos y prácticos y su principal contribución fue **simplificar las matemáticas** a un nivel entendible por no expertos.
- En el tratado se muestran las ventajas de usar el sistema decimal indio, un atrevimiento para su época, dado lo tradicional de la cultura árabe. La exposición clara de cómo **calcular de una manera sistemática a través de algoritmos diseñados** para ser usados con algún tipo de dispositivo mecánico similar a un ábaco, más que con lápiz y papel, muestra la intuición y el poder de **abstracción** de Al'Khwarizmi, que se preocupaba también de reducir el número de operaciones necesarias en cada cálculo.





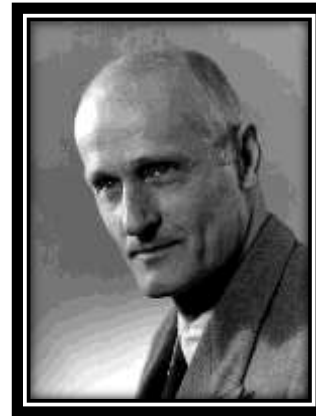


- Al-Khwarizmi vivió en Bagdad bajo los califatos de al-Ma'mum y al-Mu'tasim, en la edad de oro de la ciencia islámica. Su obra *Kitab al-jabr wa al-muqabalah* fue traducida al latín en el siglo XII dando origen al término "álgebra". En ella se compilan una serie de reglas para obtener las soluciones aritméticas de las ecuaciones lineales y de las cuadráticas; su método de resolución de tales ecuaciones no difiere en esencia del empleado en nuestros días.
- Como ya se vio es un error creer que los algoritmos son exclusivos de la computación, el algoritmo más famoso de la historia es **algoritmo de Euclides** para calcular el máximo común divisor.



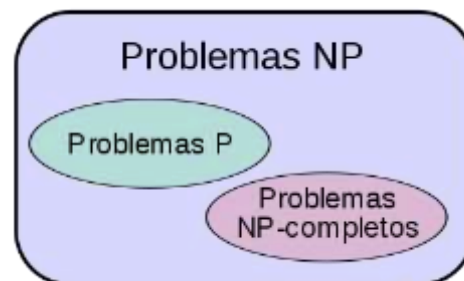


- La investigación **en modelos formales de computación** se inició en los 30's y 40's por **Alan Turing, Emil Leon Post, Stephen Kleene, Alonzo Church** y otros.
- En los 50's y 60's los **lenguajes de programación, compiladores y sistemas operativos** estaban en desarrollo, por lo tanto, se convirtieron tanto en el sujeto como la base para la mayoría del trabajo teórico.





- El poder de las computadoras en este período estaba limitado por procesadores lentos y por pequeñas cantidades de memoria. Así, se desarrollaron teorías (modelos, algoritmos y análisis) para hacer un uso eficiente de las computadoras. Esto dio origen al desarrollo del área que ahora se conoce como ***"Algoritmos y Estructuras de Datos"***.
- Al mismo tiempo se hicieron estudios para comprender la complejidad inherente en la solución de algunos problemas. Esto dio origen a lo que se conoce como la “Jerarquía de problemas computacionales” y al área de “Complejidad Computacional”.





# Algoritmo

**“Un algoritmo es una serie finita de pasos para resolver un problema”.**

- Para que un algoritmo exista:
  1. El **número de pasos** debe ser **finito**. De esta manera el algoritmo debe terminar en un tiempo finito con la solución del problema.
  2. El algoritmo debe ser capaz de determinar la **solución del problema**.
- De este modo, podemos definir **algoritmo computacional** como un **"conjunto de reglas operacionales inherentes a un cómputo"**.
  - Se trata de un **método sistemático**, susceptible de ser realizado **mecánicamente**, para resolver un problema dado en un tiempo finito.





# Características de un algoritmo

1. **Entrada:** definir lo que necesita el algoritmo
2. **Salida:** definir lo que produce.
3. **No ambiguo:** explícito, siempre sabe qué comando ejecutar.
4. **Finito:** El algoritmo termina en un número finito de pasos.
5. **Correcto:** La solución debe ser correcta
6. **Efectividad:** Cada instrucción se completa en tiempo finito.
7. **General:** Debe contemplar todos los casos de entrada.

- Una definición formal para el algoritmo es:

**“Un algoritmo es un procedimiento para resolver un problema cuyos pasos son concretos y no ambiguos. El algoritmo debe ser correcto, de longitud finita y debe terminar para todas las entradas”**





- Un algoritmo puede ser visto como una función que depende de una entrada y asocia una salida, matemáticamente se expresa de la siguiente forma:

$$y = f(x)$$

donde:

*“y” es la salida*

*“x” los datos de entrada*

*“f(x)” el algoritmo que depende de la entrada*





# Importancia de los algoritmos en computación

- **Evitan reinventar soluciones**

- Para algunos problemas de programación ya existen buenos algoritmos para solucionarlos, ya que existe un análisis de sus propiedades (confianza, eficiencia, velocidad, etc.).

- **Ayudar a generar nuevas soluciones con base en otras ya conocidas**

- No siempre existe un algoritmo desarrollado para resolver un problema.
- No existe regla general de creación de algoritmos. Muchos de los principios de proyecto de algoritmos ilustrados por algunos de los algoritmos que estudiaremos son importantes en todos los problemas de programación. El conocimiento de los algoritmos bien definidos provee una fuente de ideas que pueden ser aplicadas a nuevos algoritmos.

- **Generación de herramientas y software particular que emplea algoritmos específicos sin importar la tecnología de implementación**

- Compresión de datos, seguridad, almacenamiento de datos, comunicaciones, etc.

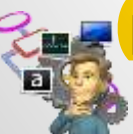






# Introducción a la complejidad de los algoritmos

- En **Ciencias de la Computación** cuando se dice que **un problema tiene solución**, significa que **existe un algoritmo susceptible de implantarse en una computadora**, capaz de producir la respuesta correcta para cualquier instancia del problema en cuestión.
- Para ciertos problemas es posible encontrar **más de un algoritmo** capaz de resolverlos.
- La tarea de decidir cuál de ellos es el mejor debe basarse en criterios acordes a nuestros intereses. En la mayoría de los casos la **elección** de un buen algoritmo está orientada hacia la **disminución del costo que implica la solución del problema**.





- El **costo** que **implica** resolver un problema genera criterios para la selección de los pasos a seguir (**programar**):
  1. **Criterios** orientados a minimizar el **costo de desarrollo**: claridad, sencillez y facilidad de **implantación**, depuración y **mantenimiento**.
  2. **Criterios** orientados a disminuir el **costo de ejecución**: **tiempo de procesador** y **cantidad de memoria utilizados**.
- Si el programa que implanta un algoritmo se va a ejecutar **unas cuantas veces**, el **costo de desarrollo es dominante**, en este caso se debe dar más peso a los primeros criterios.
- Por el contrario, si el programa se va a **utilizar frecuentemente**, **dominará el costo de ejecución**, y por ende, se debe elegir un algoritmo que haga uso eficiente de los recursos de la computadora.





- Si se implementa un sistema con **algoritmos** muy eficientes pero **confusos**, puede suceder que después no sea posible darle el mantenimiento adecuado.
- Los **recursos que consume un algoritmo pueden estimarse** mediante herramientas teóricas y constituyen, por lo tanto, una base confiable para la elección de un algoritmo.
- Siempre que se trata de resolver un problema, puede interesar considerar distintos algoritmos, con el fin de utilizar el más eficiente **¿Cómo realizar esta selección?**
  - La **estrategia empírica** consiste en **programar** los algoritmos **y ejecutarlos** en un computador sobre algunos ejemplares de prueba.
  - La **estrategia teórica** consiste en **determinar matemáticamente** la cantidad de recursos (tiempo, espacio, etc.) que necesitará el algoritmo en función del tamaño del ejemplar considerado.



- Un **algoritmo es eficiente** cuando **logra** llegar a sus **objetivos** planteados **utilizando la menor cantidad de recursos posibles**, es decir, minimizando el uso memoria, de pasos y de esfuerzo.



- Un **algoritmo es eficaz** cuando **alcanza el objetivo primordial**, el análisis de resolución del problema se realiza prioritariamente.



- ¿Puede darse el caso de que exista un algoritmo eficaz pero no eficiente?
  - Si, por lo que se deben manejar estos dos conceptos conjuntamente





- La **eficiencia** de un programa tiene dos ingredientes fundamentales: **espacio** y **tiempo**.
  - La **eficiencia en espacio** es una medida de la **cantidad de memoria requerida** por un programa.
  - La **eficiencia en tiempo** se mide en términos de la **cantidad de tiempo de ejecución** del programa.
- Ambas dependen del tipo de equipo de computo, compilador, sistema operativo, etc. por lo que no se estudiará aquí la eficiencia de los programas, sino la **eficiencia de los algoritmos**.





# Análisis A Priori y Prueba A Posteriori

- El **análisis de la eficiencia de los algoritmos** (memoria y tiempo de ejecución) consta de **dos fases: Análisis A Priori y Prueba A Posteriori**.
- El **Análisis A Priori (o teórico)** entrega una **función que limita el tiempo de cálculo** de un algoritmo.
  - Consiste en obtener una expresión que indique el comportamiento del algoritmo en función de los parámetros que influyan. Esto es interesante porque: **Permite la predicción** del costo del algoritmo. Es **aplicable en la etapa de diseño** de los algoritmos.
- En la **Prueba A Posteriori (experimental o empírica)** se **recogen estadísticas** de tiempo y espacio consumidas por el algoritmo **mientras se ejecuta**.
  - La estrategia empírica consiste en **programar** los algoritmos **y ejecutarlos** en un computador sobre algunos ejemplares de prueba, **haciendo medidas** para:
    - Una máquina concreta,
    - Un lenguaje concreto,
    - Un compilador concreto
    - Datos concretos





- La **estrategia teórica (Análisis A Priori)** tiene como **ventajas:**
  - **No depende del computador ni del lenguaje de programación, ni siquiera de la habilidad del programador.**
  - Permite **evitar** el esfuerzo inútil de **programar algoritmos ineficientes** y de **desperdiciar tiempo** de máquina para ejecutarlos.
  - **Permite conocer la eficiencia** de un algoritmo **cualquiera que sea el tamaño del ejemplar** al que se aplique.

