



Fundamentos de Programación

Clave 1122

Semestre 2023-1

Grupo 21

Universidad Nacional Autónoma de México
Facultad de Ingeniería
División de Ingeniería Eléctrica

Tema 1. Panorama general



Contenido

- ✓ Objetivo
- ✓ 1.1 Evolución de la programación.
- ✓ 1.2 Beneficios de la programación.
- ✓ 1.3 Algoritmos en la solución de problemas y sus retos.
- ✓ 1.4 Propósito y el papel de los fundamentos de programación

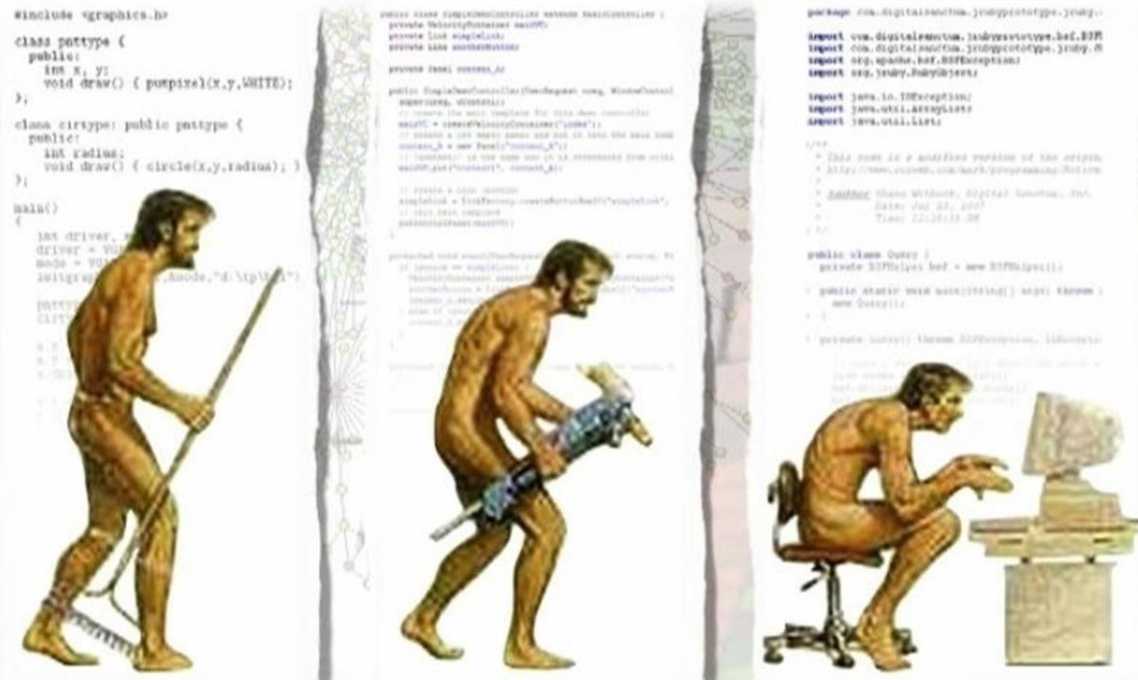


Objetivo

El alumno **definirá la importancia de la programación** como herramienta en el quehacer del **ingeniero**.



Evolución de la programación



¿Qué es la programación?

- Acción y efecto de programar.

¿Qué es programar?

- Idear y ordenar las acciones necesarias para realizar una actividad.
- Preparar ciertas máquinas o dispositivos por anticipado para que realicen una función.
- Elaborar un programa informático codificando las órdenes y datos que permiten su funcionamiento.



Programación informática

- Se refiere a la elaboración de un programa de computadora (aplicación informática o software) para que ésta realice una actividad deseada.
- Implica el diseño, la codificación y la depuración de un programa de computadora, e incluso se puede referir a su mantenimiento o actualización.

Programa de computadora

- Instrucciones que sigue una computadora para alcanzar un resultado específico (Cairó, 2005).



Software

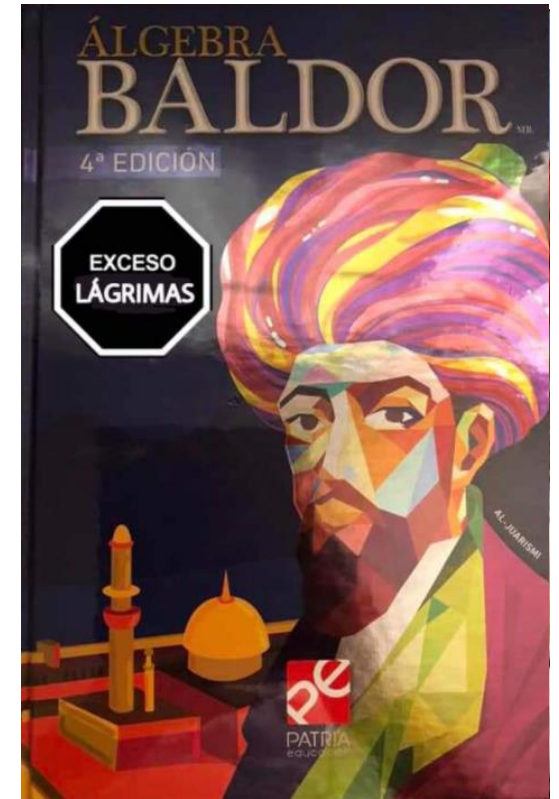
- Todos o parte de los programas, procedimientos, reglas y documentación asociada de un sistema de procesamiento de información. ISO/IEC 2382-1:1993.
- Programas de computadora, procedimientos, posiblemente documentación asociada y datos pertenecientes a la operación de un sistema de computadora. IEEE Std 829-2008.
- Programa o conjunto de programas usados para ejecutar una computadora. ISO/IEC 26514.

Permite el uso de los recursos de hardware de una computadora y el almacenamiento, procesamiento, uso, y transmisión de la información. **El software indica a la computadora qué debe hacer y cómo hacerlo.**



Historia

Año 820. Abu Abdallah Muḥammad estableció la **resolución metódica de problemas** de álgebra y cálculo numérico mediante una lista bien definida, ordenada y finita de operaciones a fin de hallar una solución al problema que se plantea: **Algoritmo**.



Año 1645. Blaise **Pascal** inventa la **Pascalina**, una de las primeras calculadoras **mecánicas**, que funcionaba a base de *ruedas de diez dientes* en las que cada uno de los dientes representaba un dígito del 0 al 9.



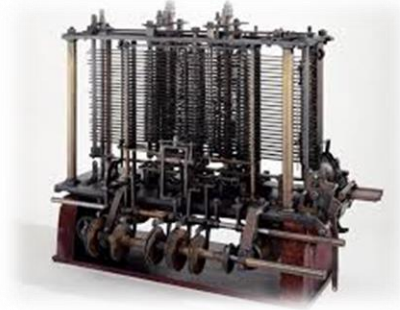
Historia

Año 1801. Josep Marie **Jacquard** utilizó un **mecanismo de tarjetas perforadas** para controlar el dibujo formado por los hilos de las telas confeccionadas por una máquina de tejer.



Historia

Año 1822. Charles **Babbage** diseñó una "[Máquina Analítica](#)", aparato de propósito general que era capaz de ejecutar cualquier tipo de cálculo matemático. Se conoce como el "**Padre de las Computadoras Modernas**".



Año 1843. Lady Ada Augusta Lovelace desarrolló el primer conjunto de instrucciones para la máquina de Babbage.



Año 1936. Alan Turing formaliza el concepto de algoritmo a través de la máquina de Turing.

Historia

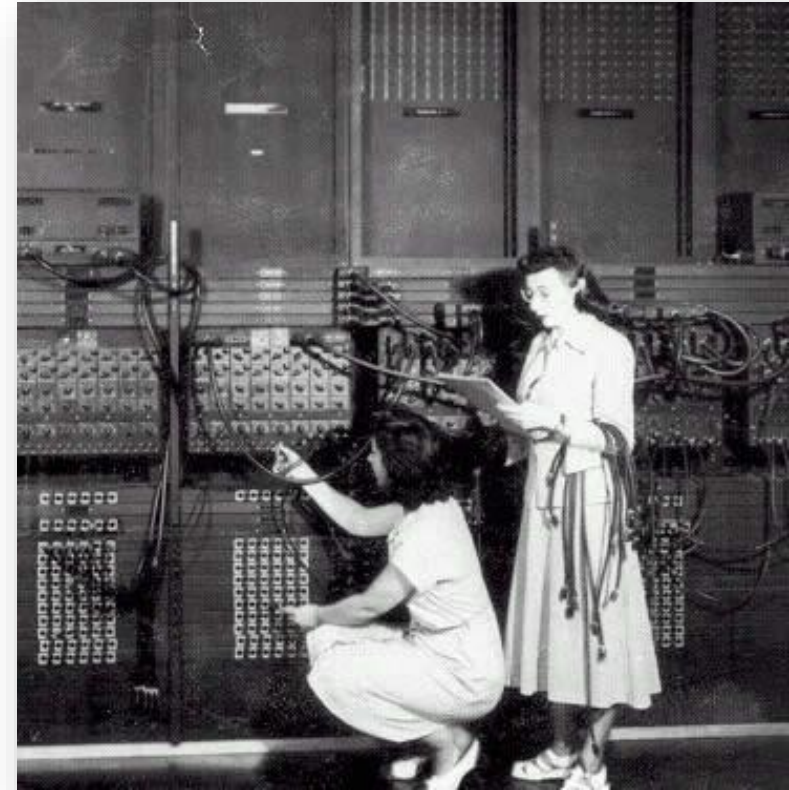
Año 1944. Se construye en la Universidad de Harvard, la **Mark I**, diseñada por un equipo encabezado por Howard H. Aiken.

Año 1946. Se construye en la Universidad de Pennsylvania la **ENIAC** (Electronic Numerical Integrator And Calculator), primera computadora electrónica de propósito general. Tenía más de 18,000 tubos de vacío, consumía 200kW de energía eléctrica y podía realizar cinco mil operaciones aritméticas por segundo.

Para efectuar las diferentes operaciones era preciso cambiar, conectar y reconectar los cables, trabajo que llegaba a tomar incluso semanas.



Historia



Cableado de la ENIAC con un nuevo programa

Historia

Año 1945 - 1946.

- **Von Neumann** planteó la idea de **almacenar instrucciones** para la computadora **en el mismo medio físico establecido para los datos**.
- Desarrolló formalmente el concepto de programa: Conjunto de instrucciones que sigue una computadora para alcanzar un resultado en específico.



Historia

Año 1949. Se desarrolló el lenguaje de programación Short Code, en el cual las instrucciones se indicaban en formato binario.

Año 1950. Grace Hooper desarrolló el compilador A-0, que traducía las sentencias en lenguaje ensamblador a código binario que la máquina interpretaba.

Año 1954. Se desarrolló la primera versión del lenguaje FORTRAN (Formula Translating System).



Historia

Año 1960. Se desarrolló el lenguaje COBOL como mecanismo de programar tareas de negocios. Incluyó como tipo de datos a cadenas de texto y números, y sus sentencias utilizaban una gramática básica del lenguaje inglés.

Año 1969. Ken Thompson y Denis Ritchie desarrollan el Sistema Operativo UNICS, conocido posteriormente como UNIX.

Año 1972. Ken Thompson y Dennis Ritchie crean el lenguaje de programación C.

Pregunta: ¿Cuándo y por qué se envió el primer correo electrónico?

Historia

Año 1994. Versión 1.0 de Python. Versión actual: 3.10.5



Año 1995. James Gosling desarrolla el lenguaje de programación JAVA.
Versión actual: 17



Año 2001. Microsoft desarrolla, como parte de su plataforma .NET, el lenguaje de programación C#. Versión actual: 10.0



Tarea 1. Fecha de entrega: Jueves 18 de agosto de 2022

Indicar en la parte superior nombre del alumno, # de tarea y fecha de entrega.

1. ¿Cuáles son las características principales de cada generación de la computación?
2. ¿Qué es un teraflop?
3. ¿Qué es una súper computadora?
4. ¿Cuáles son las 6 super computadoras más potentes de México y cuántas operaciones por segundo pueden hacer?
5. ¿Cuál es la súper computadora más potente del mundo y cuántas operaciones por segundo puede hacer?



Tarea 1. Fecha de entrega: Jueves 18 de agosto de 2022



[link](#)



Lenguaje de programación

Lenguaje diseñado para indicar instrucciones a una máquina, particularmente a una computadora. Se utiliza para simplificar el proceso de escritura de instrucciones que las computadoras pueden utilizar.



Código fuente

Instrucciones escritas en algún lenguaje de programación y almacenadas en un archivo digital de texto.



Existen **lenguajes de programación de bajo nivel y de alto nivel.**

Lenguaje de programación de bajo nivel

Proveen **muy bajo nivel de abstracción, o ninguno**, del conjunto de instrucciones de la arquitectura de la computadora.

Programación de bajo nivel

- Lenguajes Máquina
- Lenguajes Ensambladores

Lenguajes máquina

En sus inicios la programación se realizaba en Lenguaje Máquina.

- Único lenguaje que un procesador puede entender sin ninguna transformación previa.
- Las instrucciones se escribían en código binario.



Lenguajes ensambladores

El diseño de lenguajes ensambladores **sustituyó a la programación en lenguaje máquina.**

- **Uso de códigos mnemónicos o abreviaturas fáciles de recordar:**
ADD (sumar), STO (almacenar), C(comparar), SUB(restar), etc.
- Se ocupa en la programación a nivel de microprocesadores, microcontroladores, y otros circuitos integrados programables.
- Se requiere de un programa traductor que convierta las instrucciones en código máquina, tal programa se conoce como *Programa Ensamblador*.
- Portabilidad entre procesadores mínima. Cada modelo de procesador por lo general define su propio lenguaje ensamblador.

```
mov ax,0
mov ax,cx
out 70,al
mov ax,0
out 71,al
inc cx
cmp cx,100
jb 103
int 20
```

Lenguajes de programación de alto nivel

Aquellos cuya **sintaxis es más parecida a los idiomas humanos**. Es decir, **el nivel de abstracción** del lenguaje es **alto**.

Una línea de código de un lenguaje de alto nivel se traduce generalmente en muchas líneas de lenguaje máquina antes de ser ejecutadas por la computadora.

Paradigmas de programación
de alto nivel

- Programación estructurada (Lenguajes C, FORTRAN, BASIC, Shell Scripting, PHP)
- Programación orientada a objetos (C++, Java, C#, PHP)
- Otros



Lenguajes de programación compilados

Lenguajes de programación cuyo código fuente requiere un **programa compilador** para **convertir las instrucciones de un archivo de código fuente en código máquina** (instrucciones traducidas a código binario).

El **código máquina generado** a partir del código fuente se almacena en un archivo que se conoce como “**Código Objeto**”.

Cuando se **enlaza uno o más códigos objeto** se genera un “**archivo ejecutable**”; este puede ejecutar sin la necesidad de contar con el código fuente.

Cuando se desea ejecutar un programa escrito en cierto lenguaje de programación es necesario que exista un compilador para generar el apropiado código máquina de la arquitectura de la computadora donde se va a ejecutar.

Lenguajes de programación interpretados

Lenguajes de programación cuyo código fuente requiere un **programa intérprete** que **línea por línea convierte el código fuente en código máquina y lo ejecuta**, en lugar de almacenarla en un archivo.

El código interpretado se ejecuta de manera más lenta que el código compilado, debido a que la traducción y ejecución es línea por línea.

El enlazado con librerías se hace en tiempo de carga o de ejecución.

LISP y BASIC son lenguajes interpretados.



Paradigma de Programación Estructurada

Algunas características:

- Código fácil de entender debido a su estructura clara.
- Facilita el “debuggeo” del código para entender su funcionamiento y encontrar errores.
- Programación en componentes o módulos a través de funciones.
- Tres estructuras fundamentales:
 - 1. Secuenciales**
 - 2. De selección**
 - 3. De iteración**





Tema 1.2

Beneficios de la programación

Tecnologías de la Información y Comunicación (TIC)

Tecnología requerida para la creación, manipulación, almacenaje, recuperación y comunicación de la información, con lo cual se logra su eficiente organización, análisis y uso.

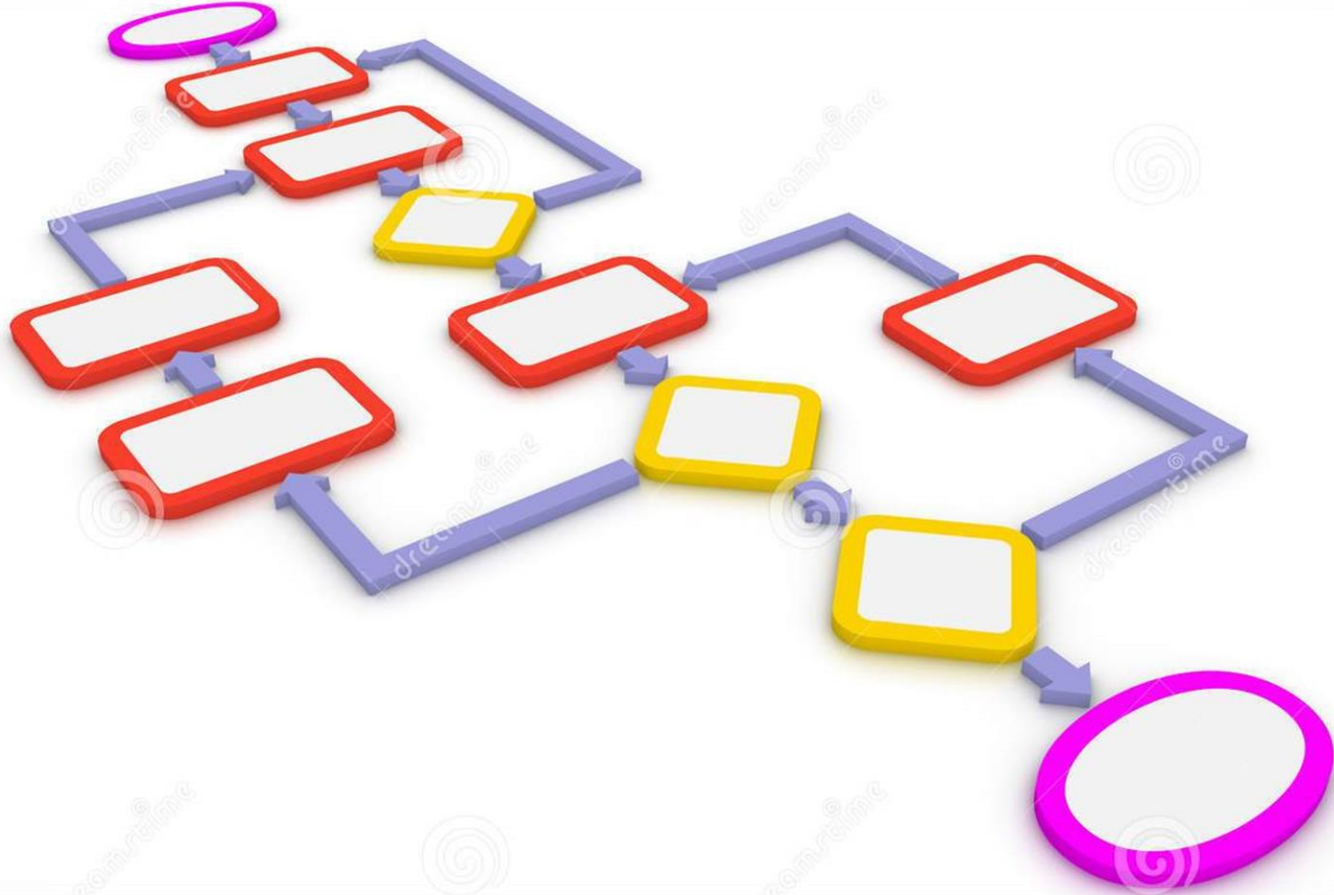
El Programa de las Naciones Unidas para el Desarrollo señala:

“Uno de los propósitos de las TIC es favorecer una amplia distribución de recursos y oportunidades, de acuerdo a los derechos humanos, democracia, creación de la sustentabilidad en salud y del medio ambiente.

Es prioridad asegurar el acceso a los medios tecnológicos y al conocimiento, manejo y aplicación de las herramientas de las TIC.”

Los beneficios de la programación se presentan a través de las TIC en la mayoría de los campos de la sociedad, al menos de forma indirecta.





Tema 1.3

Algoritmos en la solución de problemas y sus retos

Algoritmo

Conjunto de pasos, procedimientos o acciones que permiten alcanzar un resultado o resolver un problema.

No existen reglas específicas para resolver problemas. Sin embargo, se cuenta con métodos y herramientas que orientan el razonamiento necesario para resolver problemas. Tal es el caso de las soluciones basadas en algoritmos.



Algoritmo

Las tres características principales de un algoritmo son:

- **Precisión**. Los pasos a seguir deben ser indicados clara y específicamente (cero ambigüedad).
- **Determinismo**. En un algoritmo, cada que se utilicen los mismos datos de entrada, se deben producir los mismos resultados.
- **Finitud**. Todo algoritmo debe de ser de una longitud finita.



Algoritmo

En la vida diaria nos enfrentamos a situaciones que hay que resolver, desde lo muy simple a lo muy complicado.

Ejercitarse

Conseguir comida

Tarea de la escuela

Entrevista de trabajo

Salir de paseo

Trámites

Llegar a acuerdos

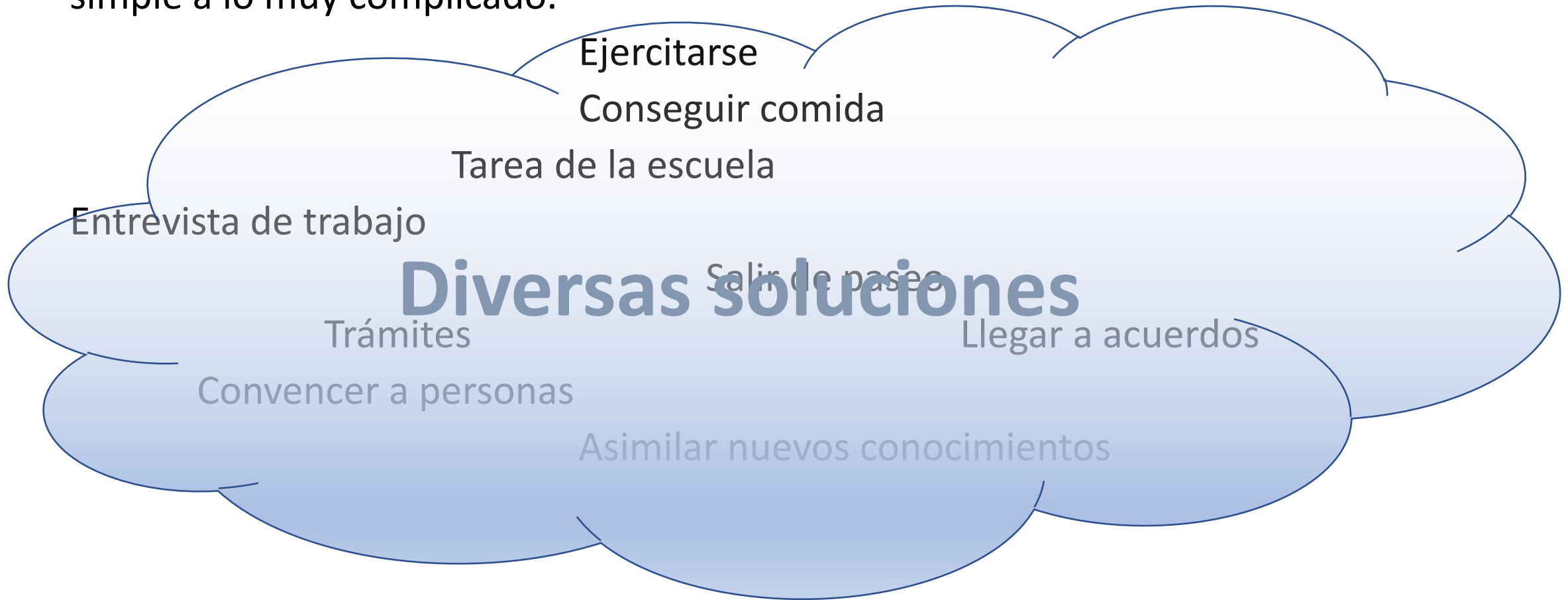
Convencer a personas

Asimilar nuevos conocimientos



Algoritmo

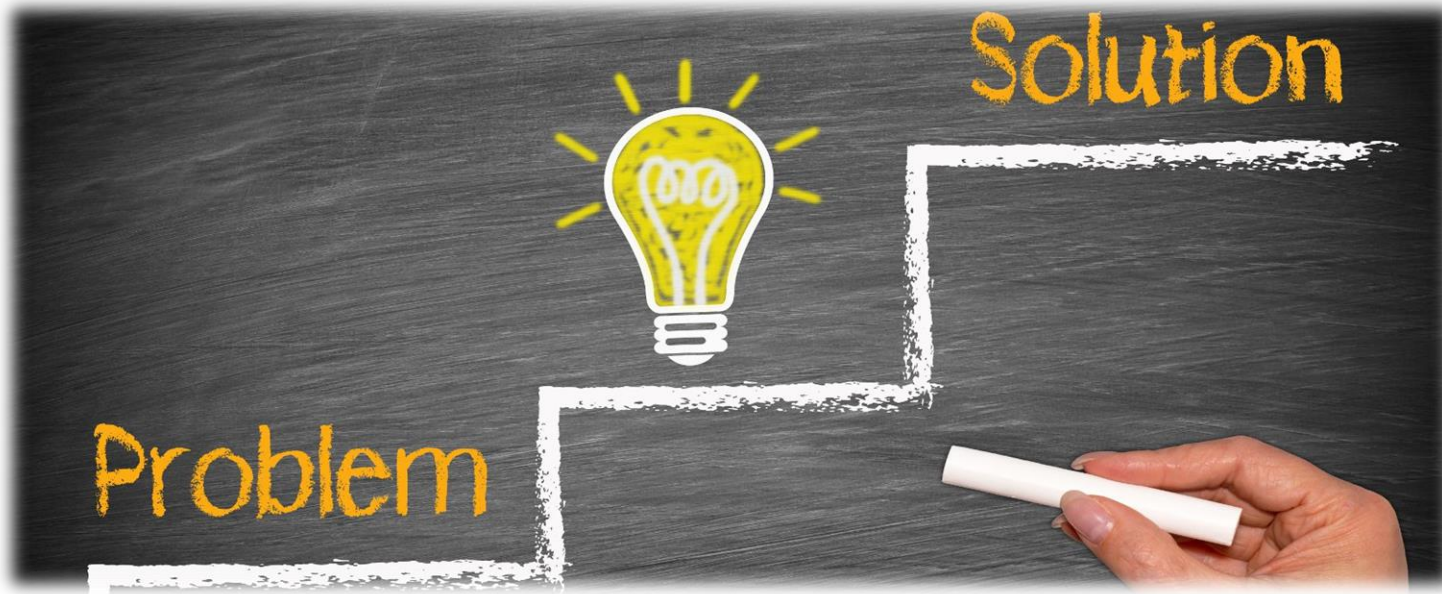
En la vida diaria nos enfrentamos a situaciones que hay que resolver, desde lo muy simple a lo muy complicado.



Algoritmo

En la vida diaria nos enfrentamos a situaciones que hay que resolver, desde lo muy simple a lo muy complicado.





Tema 1.4

Propósito y el papel de los fundamentos de la programación en la ingeniería.

Aprender a programar permite...

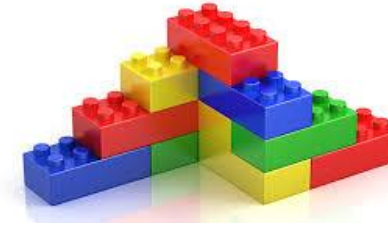


Aprender a programar permite...



Encontrar o desarrollar soluciones mediante una forma de pensamiento:

Estructurada



Flexible

Abierta





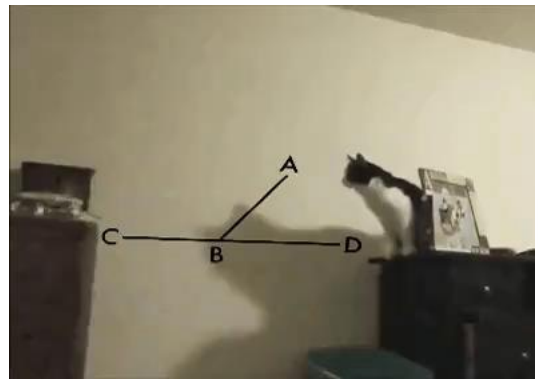
Pensamiento
lógico y
creativo



Soluciones de
manera
estructurada y
clara a partir
de pasos
básicos

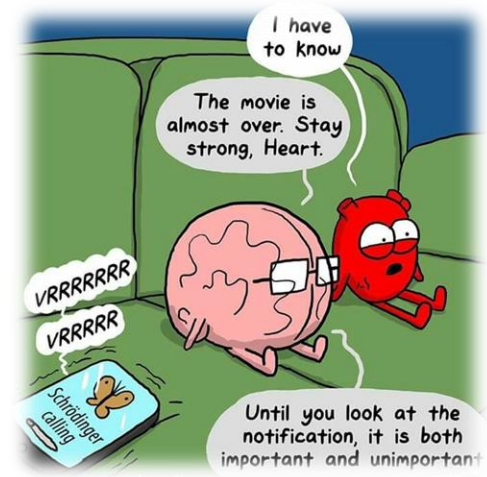
Aprender
a
programar

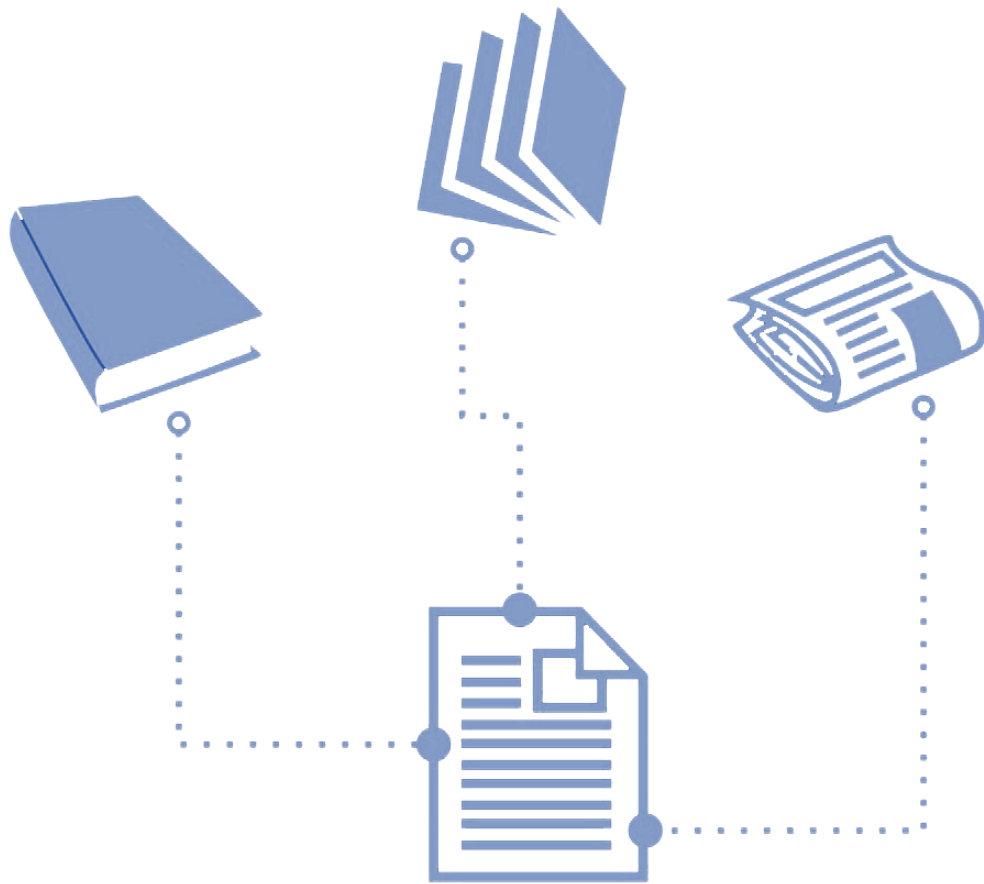
Habilidades
de análisis
sistemático



Abstracción
de
situaciones a
un entorno
de análisis y
su simulación

Habilidades
de
concentración





Referencias consultadas

1. Gómez de Silva Garza, Andrés, Ignacio de Jesús Ania Brisenio. Introducción a la computación. Primera edición. México. Cengage Learning Editores, 2008.
2. <http://www.computerhistory.org/exhibits/>
3. <http://www.computerhistory.org/babbage/>
4. Cairó, Osvaldo. Metodología de la Programación. Algoritmos, Diagramas de Flujo y Programas. 2a. Edición. México. Alfaomega, 2003
5. FELLEISEN, Matthias, FINDLET, Robert Bruce, et al. How to Design Programs. An introduction to Programming and Computing. Cambridge. MIT Press, 2001.
6. Pressman, Roger. Ingeniería de Software: Un enfoque práctico. 6ª edición, México. Editorial McGraw Hill.
7. <http://lema.rae.es/drae/?val=programaci%C3%B3n+inform%C3%A1tica>
8. [http://diccionarios.elmundo.es/diccionarios/cgi/diccionario/lee_diccionario.html?busca=programa](http://diccionarios.elmundo.es/diccionarios/cgi/diccionario/lee_diccionario.html?busca=programa&diccionario=1&submit=Buscar+)
[r&diccionario=1&submit=Buscar+](http://diccionarios.elmundo.es/diccionarios/cgi/diccionario/lee_diccionario.html?busca=programa&diccionario=1&submit=Buscar+)
9. <https://es.wikipedia.org/wiki/Programaci%C3%B3n>
10. ISO/IEC, & IEEE. (2010). ISO/IEC/IEEE 24765:2010 - Systems and software engineering -- Vocabulary. *Iso/Iec Ieee, 2010*, 410. <https://doi.org/10.1109/IEEESTD.2010.5733835>

