

# MTA - 1

**Conceptos Básicos sobre programación**

# Agenda

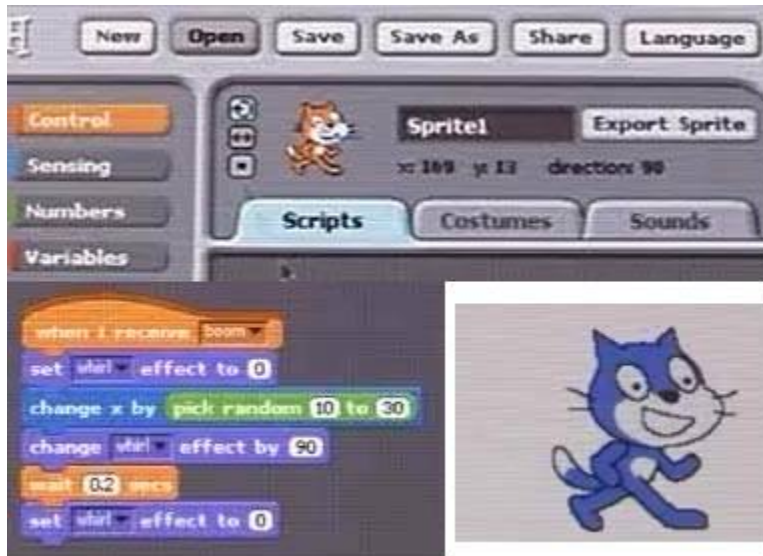
- ¿Qué es un Lenguaje de Programación?
- Clasificación.
- La programación e Ingeniería de Software.
- Fases para la elaboración de un Programa.
- Tipos de programación.
- Estándares de Programación.
- Buenas prácticas en el desarrollo de programas.

# ¿Qué es un lenguaje de programación?

- La computadora se ha acercado mucho al ser humano.
- Programar ha dejado de ser una actividad exclusiva para un grupo de personas.
- El hardware (lo físico) requiere del software (lo lógico, los programas) para funcionar.
- La computadora solamente entiende matemática binaria (1 y 0).

# ¿Qué es un lenguaje de programación?

El profesor Mitchel Resnick del MIT nos dice que la programación ha sido vista como algo más allá del alcance general. Esto es mentira.



Hoy en día la programación está al alcance de todos como por ejemplo usando <http://scratch.mit.edu/> un aplicativo que permite programar cosas simples de forma muy fácil.

# ¿Qué es un lenguaje de programación?

Los programas son escritos en algún **lenguaje de programación**, comprensible para el ser humano y luego son traducidos mediante un proceso de **compilación** o **interpretación** hacia el sistema.

# ¿Qué es un lenguaje de programación?

**Compilación:** Proceso por el cuál el código del programa es convertido a un lenguaje comprensible por el **sistema operativo**. Cada sistema operativo usa **lenguajes diferentes** por lo que cuando se compila un programa se hace para un **sistema operativo específico**.

# ¿Qué es un lenguaje de programación?

**Interpretación:** Proceso por el cuál el código del programa es **interpretado línea a línea** cada vez que se ejecuta. De esto se suelen encargar servidores específicos de ese lenguaje. Se usa mucho en **Internet** pues permite que los programas no dependan de un sistema operativo y son multiplataformas por ello (se pueden usar desde cualquier sistema operativo: windows, android, linux, etc.)

# ¿Qué es un lenguaje de programación?

- Las computadoras **no pueden actuar o tomar decisiones** por si mismas.
- **Los programas** son los que nos hacen parecer que son aparatos inteligentes.



# ¿Qué es un lenguaje de programación?

**Concepto:** “Un lenguaje de programación es un sistema notacional para describir computaciones en una forma legible tanto para la máquina como para el ser humano”

# Clasificación

Según su nivel de abstracción

- Lenguajes de bajo nivel
- Lenguajes de medio nivel
- Lenguajes de alto nivel

Según la forma de ejecución

- Lenguajes compilados
- Lenguajes interpretados

Según el paradigma de programación

- Lenguajes imperativos
- Lenguajes declarativos
- Lenguajes estructurados
- Lenguajes Funcionales
- Lenguajes Lógicos
- Lenguajes orientados a objetos

# Clasificación

## a. Por su nivel de abstracción

a.1. **Lenguajes de bajo nivel** – Muy cercanos al lenguaje de procesador (Lenguaje Ensamblador)

a.2. **Lenguajes de medio nivel** – Menos cercanos al lenguaje de procesador (Lenguaje C)

a.3. **Lenguajes de alto nivel** – Más cercanos al lenguaje humano (Java, PHP, C++, FORTRAN, etc)

# Clasificación

## b. Por su forma de ejecución

b.1. **Lenguajes compilados** – Son los que pasan por un proceso de compilación que les permite funcionar sobre un sistema operativo determinado (C, C++, Pascal, Delphi, etc).

b.2. **Lenguajes interpretados** – Son interpretados en tiempo de ejecución y no generan un producto compilado (JavaScript, LISP, PHP).

# Clasificación

## c. Según el paradigma de programación

- c.1. Lenguajes Imperativos (Java, C)
- c.2. Lenguajes Declarativos (Prolog, SQL)
- c.3. Lenguajes Estructurado
- c.4. Lenguajes Orientados a Objetos
- c.5. Lenguajes Funcionales (Miranda, LISP)
- c.6. Lenguajes Lógicos (Prolog)

# Fases para la elaboración de un programa.

- 1.- Análisis del Problema.
  - a. Dato de Entrada
  - b. Dato de Salida
- 2.- Diseño del Algoritmo.
  - a. Representación del Algoritmo
  - b. Diseño del juego de datos de prueba
- 3.- Codificación.
- 4.- Ejecución.
- 5.- Depuración.
- 6.- Documentación

# Buenas prácticas en el desarrollo de software

## 1. Corrección

Un programa es correcto si **hace lo que debe hacer**.

Hay que especificar antes de su desarrollo, **que es lo que debe hacer** para poder comparar los resultados.

# Buenas prácticas en el desarrollo de software

## 2. Claridad

Es importante que el programa sea **lo más claro y legible posible** para mejorar su mantenimiento.

## 3. Eficiencia

Debe consumir la **menor cantidad de recursos**, medidos en lo posible en tiempo y memoria.



# La programación y la ingeniería de software

1. Reconocer la necesidad de un programa para solucionar problemas o identificar la posibilidad de automatización de una tarea.
2. Recoger los requisitos del programa.  
Debe quedar claro qué es lo que debe hacer el programa y para qué se necesita

# La programación y la ingeniería de software

## 3. Realizar el análisis de los requisitos de programa.

Debe quedar claro cómo debe realizar el programa las cosas que debe hacer.

Definir las pruebas que comprueben su validez.

Evaluar y determinar la tecnología necesaria para el desarrollo del programa (lenguaje de programación, base de datos, etc)

# La programación y la ingeniería de software

## 4. Diseñar la arquitectura del programa.

Se debe descomponer el programa en partes de complejidad menor.

Definir la integración de los componentes del programa.

Elaborar los documentos o herramientas para la transmisión de la solución (Diagrama de clases, pseudocódigo, Casos de prueba)

# La programación y la ingeniería de software

## 5. Implementación.

Se debe elegir el lenguaje de programación adecuado a las necesidades del programa y a los avances tecnológicos emergentes, se elabora de acuerdo al diseño.

Se ejecuta la programación en sí misma.

# La programación y la ingeniería de software

## 6. Pruebas y validación.

Se debe probar de manera eficiente el programa, estas pruebas involucran en el proceso a todos los escenarios posibles de manera de asegurar la calidad del programa.

# La programación y la ingeniería de software

## 7. Implantar el programa.

Poner el programa en funcionamiento junto con los componentes necesarios (BD, comunicaciones, etc)

La ingeniería de Software se centra en los pasos de planificación y diseño del programa