# Introducción al desarrollo software.

## Lenguajes de programación.

# ¿Qué es un programa informático?

- Un programa informático es un conjunto de instrucciones que se ejecutan de manera secuencial con el objetivo de realizar una o varias tareas en un sistema.

# Lenguajes de programación

- Un lenguaje de programación es un conjunto de instrucciones, operadores y reglas de sintaxis y semánticas, que se ponen a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existente.

# Evolución histórica de los lenguajes de programación.

- A continuación procedemos a revisar como han evolucionado los lenguajes de programación a lo largo de la historia.
- Llamamos Nivel de abstracción al modo en el que los lenguajes se alejan del código máquina y se acercan cada vez más a un lenguaje natural (el que empleamos las personas).

# Lenguajes de bajo nivel: Primera generación

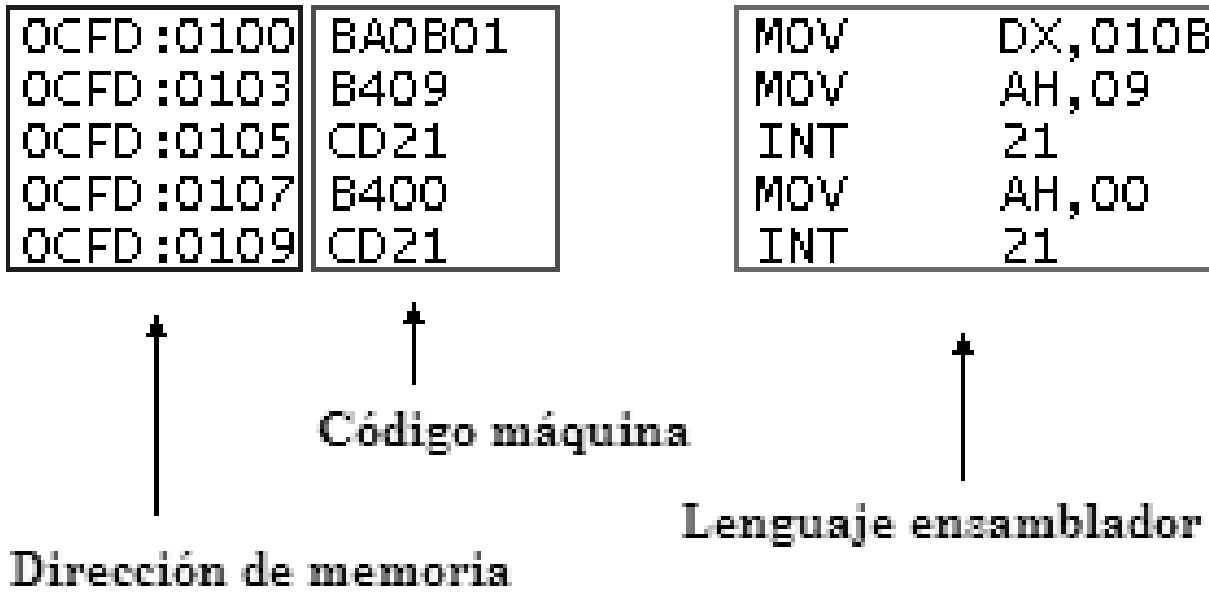
- Conocido como código máquina. Cadenas interminables de secuencias de números 1 y 0 que conforma operaciones que la máquina puede entender sin interpretación alguna.
- Características:
  - El lenguaje esta ligado íntimamente a la CPU, baja portabilidad.
  - No existen los comentarios
  - Los datos se referencia por medio de las direcciones de memoria donde se encuentran.
  - Las instrucciones realizan operaciones muy simples.
  - Porca versatilidad para la redacción de instrucciones, ya que tiene un formato muy rígido.

# Lenguajes de medio nivel: Segunda generación

- Constituye el primer intento de sustitución del lenguaje máquina por uno más cercano al usado por los humanos. Este acercamiento se plasma en las siguientes aportaciones:
  - Uso de una *notación simbólica o nemotécnica* para representar los códigos de operación. Normalmente mediante abreviaturas en ingles por ejemplo ADD.
  - *Direccionamiento simbólico*. En lugar de utilizar direcciones binarias absolutas, los datos pueden identificarse con nombres. (Variables).
  - Se permite el *uso de comentarios* entre las líneas de instrucciones,
- Aparte de esto, el lenguaje ensamblador presenta la mayoría de los inconvenientes del lenguaje máquina:
  - Repertorio muy reducido y rígido de instrucciones.
  - Baja portabilidad y la fuerte dependencia del hardware.
  - Mantiene la ventaja del uso óptimo de los recursos hardware, permitiendo la obtención de un código muy eficiente.

# Primera generación vs Segunda Generación

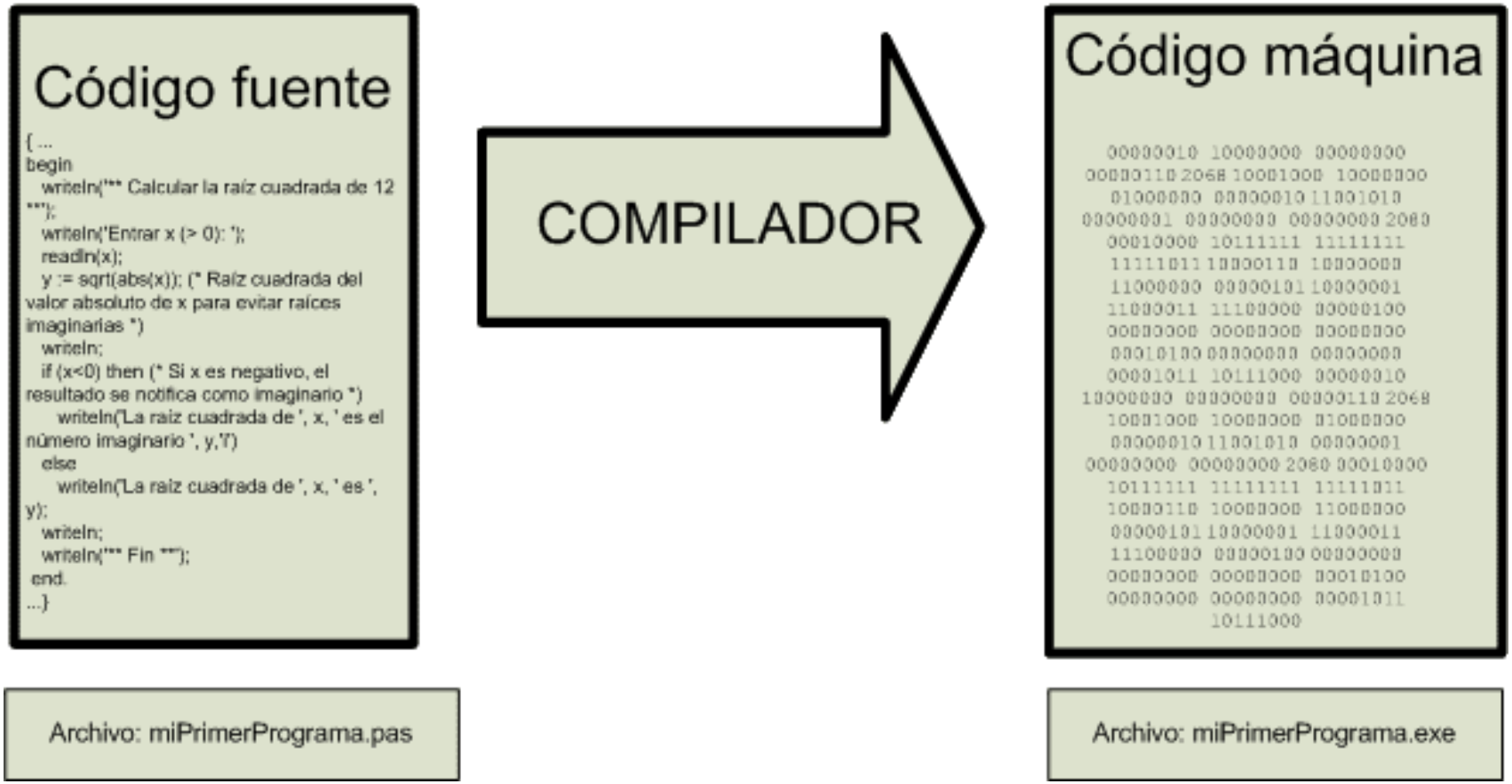
- Para convertir el lenguaje ensamblador en código máquina también conocido como código objeto haremos uso de un compilador.



# Lenguajes de alto nivel
## Tercera generación

- Los esfuerzos encaminados a hacer la labor de programación independiente de la máquina dieron como resultado la aparición de los lenguajes de programación de alto nivel. Estos lenguajes, más evolucionados, utilizan palabras y frases relativamente fáciles de entender y proporcionan también facilidades para expresar alteraciones del flujo de control de una forma bastante sencilla e intuitiva. Ejemplo: Fortran, C, C++, Java, BASIC …
- Características:
  - Son independientes de la arquitectura física del ordenador. Portabilidad.
  - Una sentencia es traducida a varias instrucciones en lenguaje máquina.
  - Las operaciones se expresan con sentencias muy parecidas al lenguaje matemático o al lenguaje natural. Se utilizan, por lo general, palabras o términos en inglés.
  - Permiten el uso de comentarios.
  - Hacen uso de variables.

# Lenguajes de alto nivel
## Tercera generación



# Lenguajes de alto nivel
# Cuarta generación

- El objetivo de los lenguajes de programación de cuarta generación es la de proveer de un nivel mayor de abstracción al programa del hardware del equipo.
- Estos lenguajes proporcionan:
  - Manejo de base de datos.
  - Desarrollo de interfaces gráficas.
  - Desarrollo de aplicaciones web.
  - Generación de informes.
  - Optimización matemática.

```
SELECT "users".* FROM "users"
  WHERE "users"."name" = $1
```

- Ejemplos: SQL, MATLAB, Scilab

# Lenguajes de alto nivel
## Quinta generación

- Un lenguaje de programación de quinta generación es un lenguaje de programación basado en la resolución de problemas utilizando restricciones dadas al programa, en lugar de utilizar un algoritmo escrito por un programador.
- Lenguajes de quinta generación se utilizan principalmente en la investigación de la inteligencia artificial. Prolog, OPS5, y el mercurio son ejemplos de lenguajes de quinta generación. Estos tipos de lenguajes también se basaron en Lisp.

# Traductores: Compiladores e Interpretes.

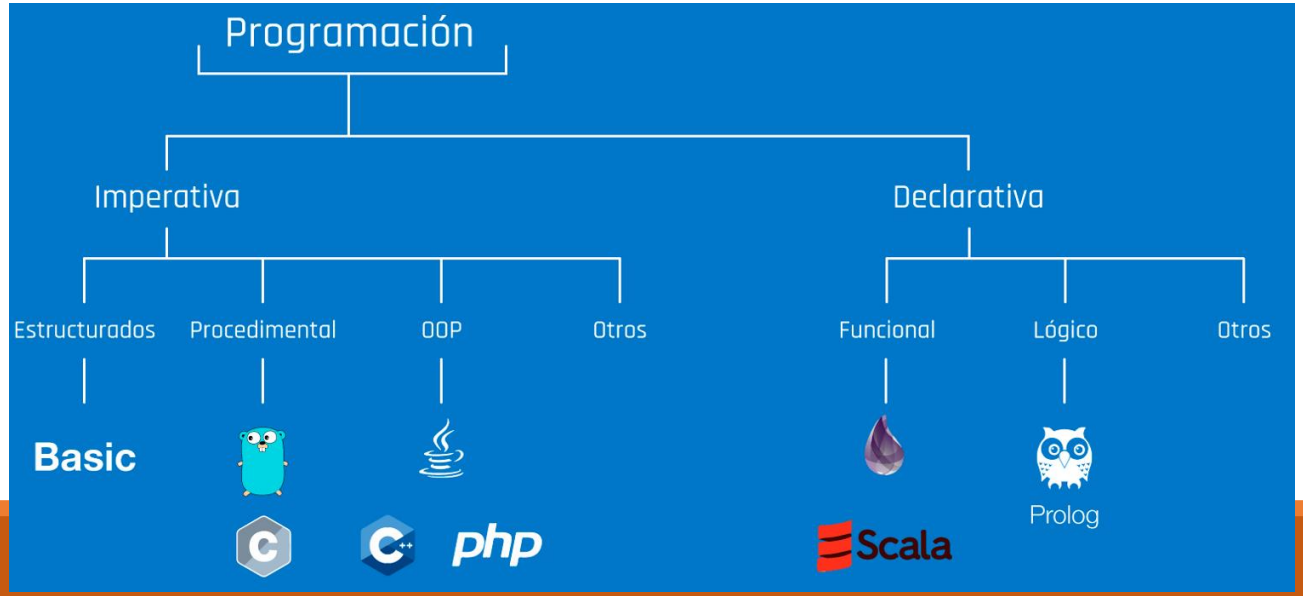
- Un **traductor** es un metaprograma que toma como entrada un programa (o parte de un programa) escrito en lenguaje simbólico, alejado de la máquina, denominado **programa fuente** y proporciona como salida otro programa, semánticamente equivalente, escrito en un lenguaje comprensible por el hardware del ordenador, denominado **programa objeto**.
- **Metaprograma**: un programa que manipula otro programa.

# Traductores: Compiladores e Interpretes. (II)

- Un **compilador** traduce completamente un programa fuente, escrito en un lenguaje de alto nivel, a un programa objeto, escrito en lenguaje ensamblador o máquina. Ejemplo: Java, C, C++ …
  - Al compilar nos devolverá un archivo ejecutable **.exe .sh .run**
- Un **intérprete** permite que un programa fuente escrito en un determinado lenguaje vaya traduciéndose y ejecutándose directamente, sentencia a sentencia, por el ordenador.
  - El intérprete capta una sentencia fuente, la analiza e interpreta, dando lugar a su ejecución inmediata, no creándose, por tanto, un archivo o programa objeto almacenaje en memoria masiva para posteriores ejecuciones. Ejemplo: Javascript, HTML, Python …

# Paradigmas de la programación.

- **Un paradigma de programación es un estilo de desarrollo de programas**. Es decir, un modelo para resolver problemas computacionales. Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis.



# Paradigmas de la programación. (II)

- **Imperativa:** Un programa se describe en términos de instrucciones, condiciones y pasos que modifican el estado de un programa al permitir la mutación de variables, todo esto con el objetivo de llegar a un resultado.
- **Declarativo:** Un programa se describe en términos de proposiciones y afirmaciones que son declaradas para describir el problema, sin especificar los pasos para resolverlo; en este tipo de programas, el estado no puede ser modificado ya que todos los tipos de datos son inmutables.

# Paradigmas de la programación. (III)

- **Estructurado:** paradigma orientado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora recurriendo únicamente a subrutinas y tres estructuras básicas: secuencial, selección e iteración.
- **Procedimental:** basada en la metodología denominada "divide y vencerás", es decir, en la división del programa en subprogramas; cada uno de ellos ejecuta una tarea independiente, y se codifican por separado unos de otros, hasta conseguir el nivel óptimo en estos subprogramas.
- **Orientado a objetos (OOP):** se basa en la idea natural de la existencia de un mundo lleno de objetos. Los objetos tienen características (atributos) y comportamiento (métodos). Los objetos interactúan entre ellos a través de relaciones.