

# CONCEPTOS FUNDAMENTALES

## INTRODUCCIÓN A LA PROGRAMACIÓN

# ORDENADOR O COMPUTADOR

- **Máquina** de componentes físicos y electrónicos
- **Procesa datos** de una forma **rápida** y **eficaz**
- Su **comportamiento** se rige por una serie de **programas** o aplicaciones



# PROGRAMA INFORMÁTICO

- Se trata de un **conjunto** de **órdenes** o instrucciones escritas mediante un **código**
  - Operaciones matemáticas
  - Entrada y salida de datos (pantalla, disco duro, ...)
  - Toma de decisiones, procesos repetitivos, ...
- Las órdenes (el código) se escriben en un **lenguaje de programación**

# PROGRAMA INFORMÁTICO



Muy pequeño

vs

Muy grande



# LENGUAJE DE PROGRAMACIÓN

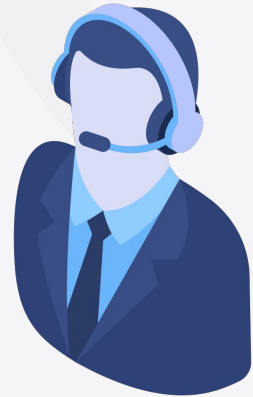
- Problemas de comunicación humano vs. computadora



0110011010  
11100011111



Ordena  
estas  
facturas  
por fecha



# LENGUAJE DE PROGRAMACIÓN

- ¿Cómo salvar este **desfase**?
- A través de un **lenguaje de programación**
  - No tan complicado para las personas como el binario
  - No tan complicado para la máquina como el lenguaje humano

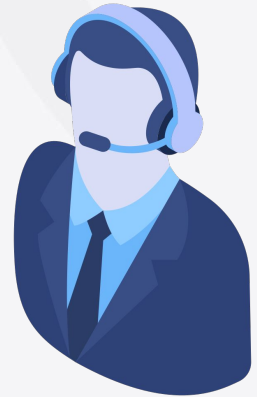
# LENGUAJE DE PROGRAMACIÓN

- Problemas de comunicación humano vs computadora



```
Collections.sort(facturas, Sort.by("fecha"));
```

*¡Ahora nos entendemos mejor!*



# TIPOS DE LENGUAJES DE PROGRAMACIÓN

- **Bajo nivel** (de abstracción): más cercanos a cómo *habla* la máquina
  - **Ensamblador**
    - Operaciones que el procesador sabe interpretar directamente.

## Assembly vs. machine code

Machine code bytes	Assembly language statements
	foo:
B8 22 11 00 FF	movl \$0xFF001122, %eax
01 CA	addl %ecx, %edx
31 F6	xorl %esi, %esi
53	pushl %ebx
8B 5C 24 04	movl 4(%esp), %ebx
8D 34 48	leal (%eax,%ecx,2), %esi
39 C3	cmpl %eax, %ebx
72 EB	jnae foo
C3	retl

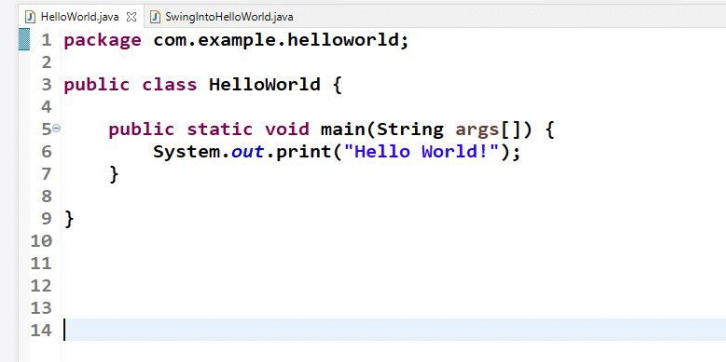
### Instruction stream

B8 22 11 00 FF 01 CA 31 F6 53 8B 5C 24  
04 8D 34 48 39 C3 72 EB C3



# TIPOS DE LENGUAJES DE PROGRAMACIÓN

- **Alto nivel:** más cercano a cómo hablamos los *humanos* (inglés)
- Gran variedad dependiendo del objetivo y las funcionalidades que incluye.
  - C: dentro del alto nivel, el más bajo nivel de todos
  - Python: generalista, web, inteligencia artificial.
  - Java: aplicaciones empresariales

A screenshot of a code editor window with two tabs: 'HelloWorld.java' and 'SwingIntoHelloWorld.java'. The 'HelloWorld.java' tab is active, displaying the following Java code:

```
1 package com.example.helloworld;
2
3 public class HelloWorld {
4
5     public static void main(String args[]) {
6         System.out.print("Hello World!");
7     }
8 }
9
10
11
12
13
14 |
```

# TIPOS DE LENGUAJES DE PROGRAMACIÓN

- ¿Qué **diferencias** hay entre un lenguaje y otro?
  - Algo similar a lo que sucede entre los idiomas humanos
    - Diferente sintaxis (escribir de izq. a der. o de der. a izq.)
    - Diferente léxico (*cout* vs *print*)
    - Diferente paradigma (forma de plantear y solucionar un problema)
    - ...

# ELEMENTOS DE UN LENGUAJE DE PROGRAMACIÓN

- **Sintaxis**

- Combinaciones de símbolos que forman un programa correcto

- **Semántica**

- Acciones realizadas por un programa que resuelven el problema correcto.

- *Juan es un soltero casado*

- Es una frase sintácticamente correcta, pero semánticamente no.

# TIPOS DE CÓDIGOS EN PROGRAMACIÓN

- Código **fuentes**
  - Es el código que escribe uno o varios programadores
  - Se escribe en un lenguaje de programación (i.e. Java)
- Código **objeto**
  - Es la *traducción* del código fuente a un código entendible por el ordenador (sistema operativo concreto)

# TIPOS DE CÓDIGOS EN PROGRAMACIÓN

- Código **ejecutable**
  - Es la unión de diferentes códigos objeto para conformar el programa completo.

# PROGRAMAS QUE RESUELVEN PROBLEMAS

- Un programa o aplicación informática surge para **resolver problemas**
  - La gestión de una empresa.
  - La compra/venta de productos.
  - Predicciones meteorológicas
- Para ello, debe seguir una serie de instrucciones precisas.

# ALGORITMO

- Es una **serie ordenada de pasos** que nos llevan a resolver un problema dado.
- Para definir un **algoritmo** *no hace falta un lenguaje de programación.*
- La ventaja de usar algoritmos en un programa informático es la **velocidad** y la **capacidad de procesamiento**.

# ALGORITMO

## Algoritmo de Euclides

	Cocientes			
	3	1	4	6
1534	403	325	78	13
	325	78	13	0
	Residuos			

← M.C.D

- Algoritmo de Euclides
- Cálculo del mcd de dos números
  - Divisiones sucesivas
  - Paramos cuando el resto es 0.



# RESOLUCIÓN COMPUTACIONAL DE UN PROBLEMA

- Para crear un programa que resuelva un problema, debemos seguir varias **fases**:
  - **Análisis** del problema
  - **Diseño** de la solución
  - **Codificación**

# RESOLUCIÓN COMPUTACIONAL DE UN PROBLEMA

- **Análisis del problema**
  - Entender bien cuál es el problema
  - Determinar cuáles son los elementos de entrada
  - Definir cuál será su salida o resultado.
- *Ejemplo: calcular el volumen de un cilindro*
  - Datos de entrada: radio y altura
  - Datos de salida: volumen

# RESOLUCIÓN COMPUTACIONAL DE UN PROBLEMA

- **Diseño de una solución**
  - No es nada gráfico. Se refiere a diseñar = pensar.
  - Proponer un algoritmo para resolver el problema
  - Representar el algoritmo: pseudocódigo, diagrama de flujo, ...
- *Ejemplo: calcular el volumen de un cilindro*
  - Fórmula:  $\pi r^2 h$ , siendo  $h$  la altura y  $r$  el radio
  - Unidad de medida:  $r$  y  $h$  en cm,  $V$  en  $cm^3$

# RESOLUCIÓN COMPUTACIONAL DE UN PROBLEMA

- **Diseño de una solución**
- *Ejemplo: calcular el volumen de un cilindro*
  - Pasos a seguir:
    - Leer el radio
    - Leer la altura
    - Cálculo del volumen
    - Escribir el volumen

# RESOLUCIÓN COMPUTACIONAL DE UN PROBLEMA

- **Codificación**
  - Trasladar el algoritmo a un lenguaje de programación
    - Escribir el código fuente
    - Depurar los posibles errores existentes

# RECAPITULAMOS

- Los **programas** o aplicaciones son los que determinan el comportamiento de un ordenador.
- Los **lenguajes de programación** permiten salvar el desfase entre el binario (lenguaje del ordenador) y el lenguaje natural.
- Los **lenguajes de programación** tienen una sintaxis y una semántica, que nos permiten escribir programas.
- Hay diferentes **tipos** de lenguajes de programación según su ámbito de aplicación.

# RECAPITULAMOS

- En programación hay tres tipos de **código**: el **fuelle**, escrito por los programadores, el código **objeto** y el código **ejecutable**.
- Un **algoritmo** es una secuencia ordenada de pasos para resolver un problema dado.
- Los algoritmos son llevados a la práctica por programas de ordenador para **aumentar** su **capacidad** de procesamiento y **velocidad**.

# RECAPITULAMOS

- Los programas sirven para resolver problemas en tres **fases**: ***análisis, diseño y codificación.***
- En el **análisis**, entendemos bien el **problema** a resolver.
- En el **diseño**, proponemos un **algoritmo** para solucionarlo.
- En la **codificación**, trasladamos el algoritmo a un código fuente de un lenguaje de programación.