EFA
MORATALAZ

1º CFGS Desarrollo de Aplicaciones Web

ENTORNOS DE DESARROLLO

DANIEL GONZÁLEZ-CALERO JIMÉNEZ

UT1 – DESARROLLO DE SOFTWARE

EFA
MORATALAZ

1º CFGS Desarrollo de Aplicaciones Web

ENTORNOS DE DESARROLLO

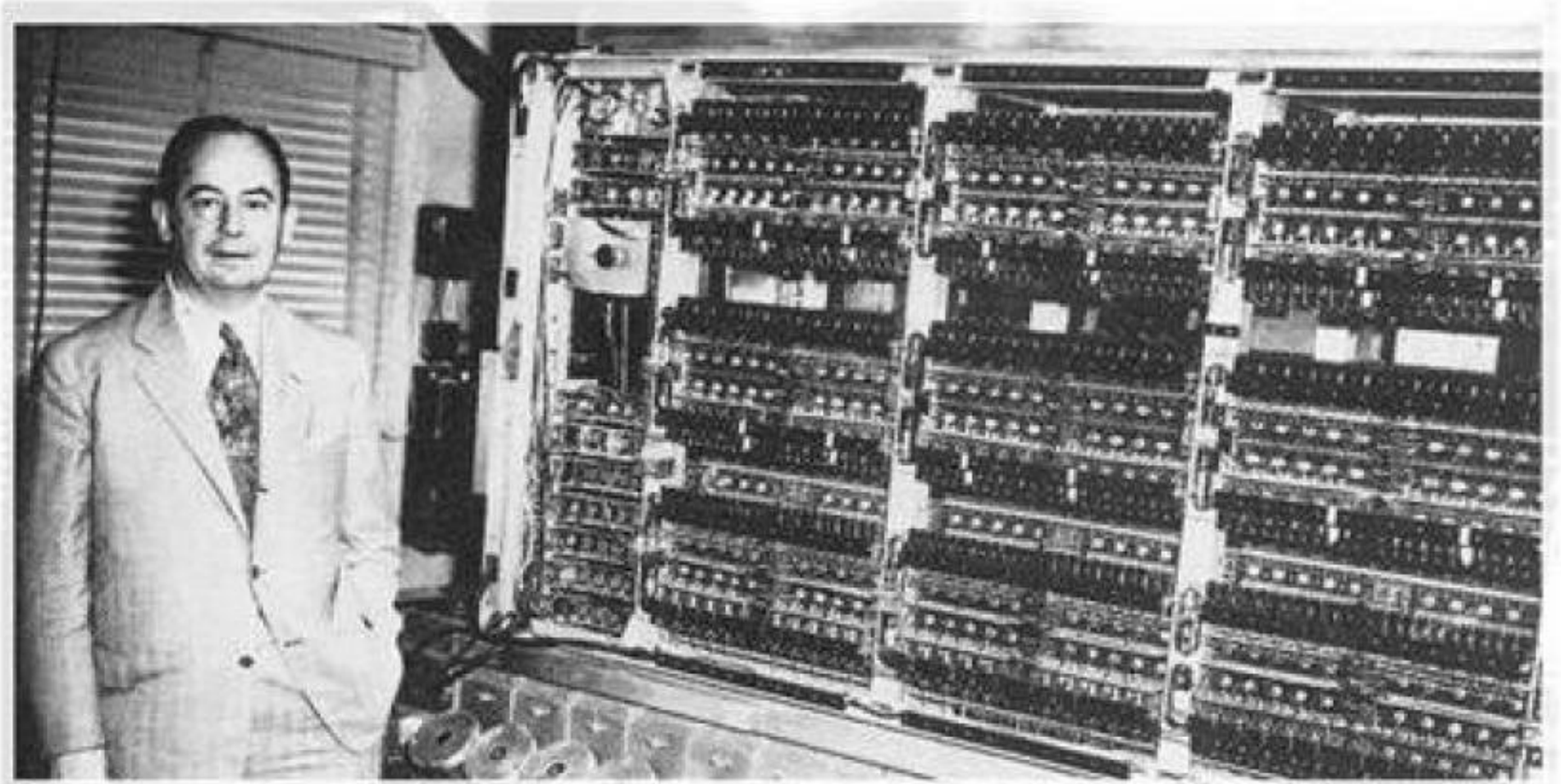
INDICE

UT1 – DESARROLLO DE SOFTWARE

CONCEPTO DE UN PROGRAMA INFORMÁTICO

- Un **programa** es una serie de instrucciones ordenadas con una finalidad concreta que realizan una función determinada.
- Para programar solo es necesario pensar **QUÉ** queremos hacer y **CÓMO** vamos a hacerlo, pensando siempre en el objetivo que queremos obtener.
- Una vez tengamos el QUÉ y el CÓMO solo tenemos que elegir un **lenguaje de programación**:
  - Java, Python, C++, Ada, etc.

- ¿Cómo interactúa un programa informático con nuestro ordenador?
- ¿Para qué sirve la memoria de nuestro ordenador?
- ¿Para qué necesito tener una CPU más potente?



Mathematicians John Von Neumann, J. Presper Eckert and John Mauchly came up with the concept of the stored instruction digital computer.

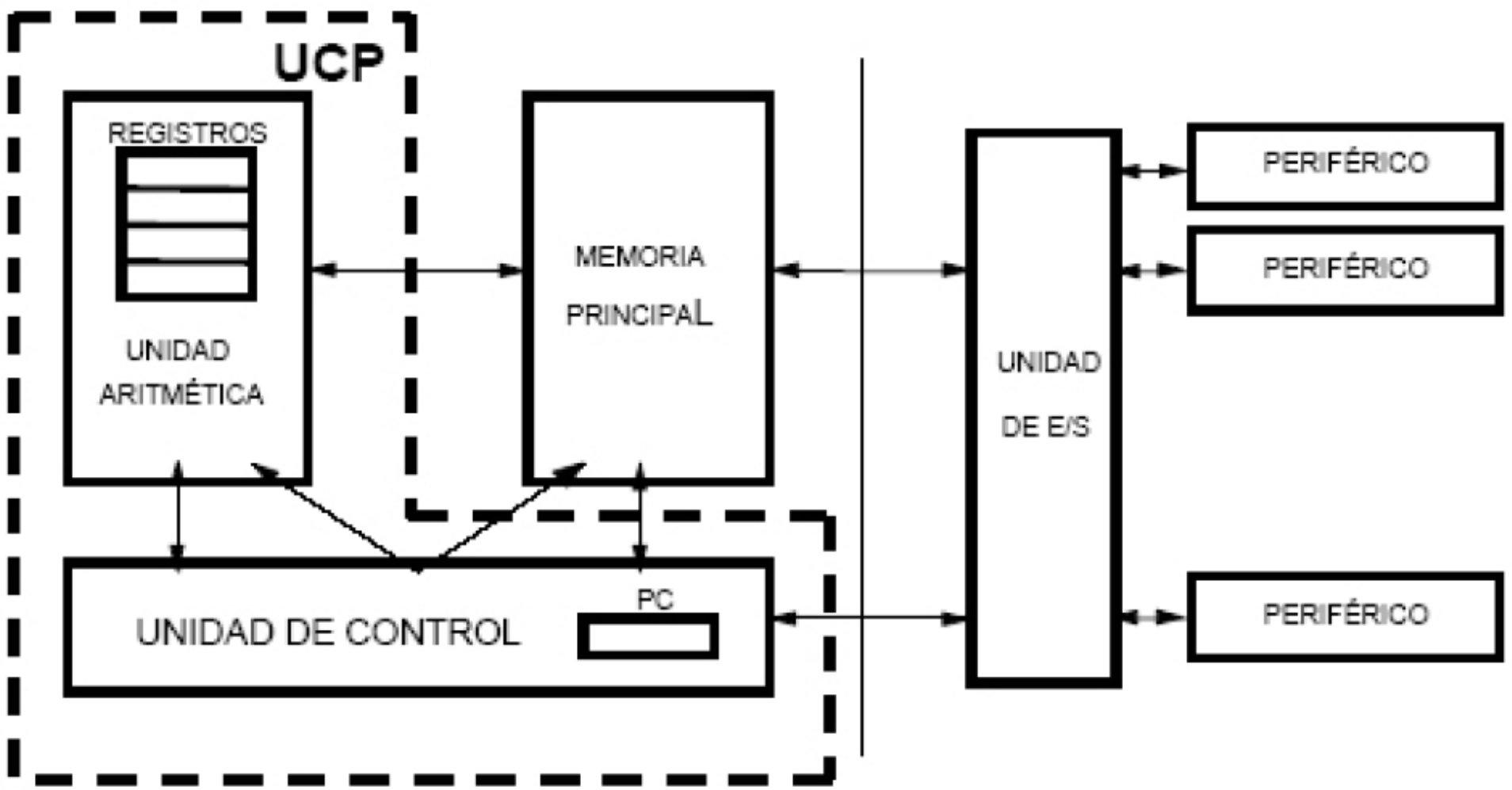
Von Neumann describió el fundamento de todo ordenador con programas almacenados:

- Ordenador con unidades conectadas permanentemente y funcionamiento coordinado desde una Unidad de Control

Aunque la tecnología ha avanzado mucho y aumentado la complejidad de la arquitectura inicial, la base de su funcionamiento es la misma y probablemente lo seguirá siendo durante mucho tiempo

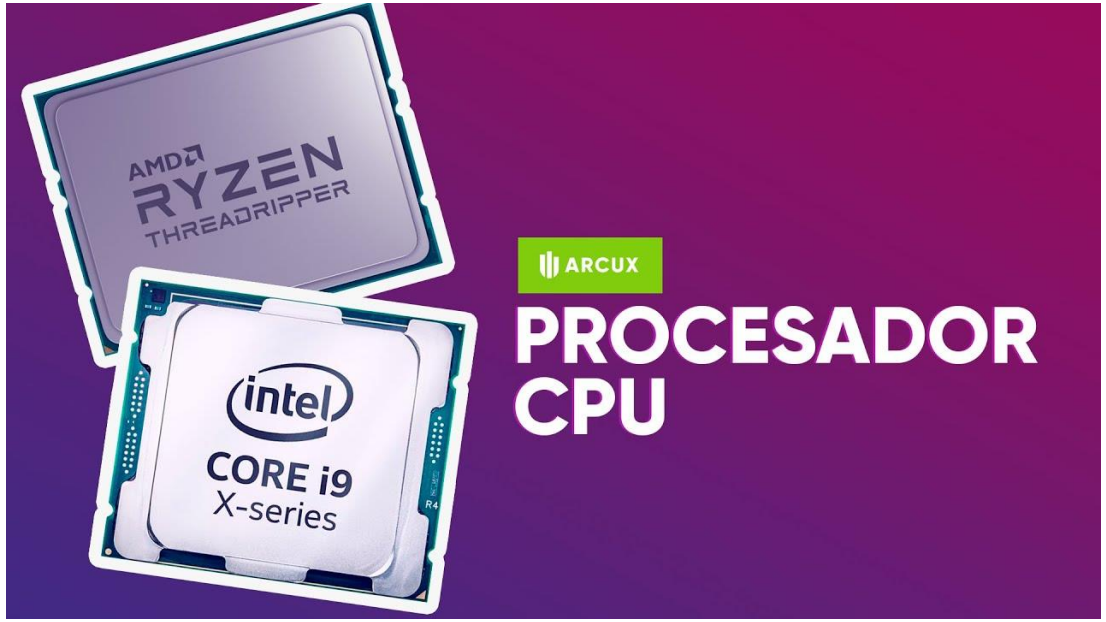
CONCEPTOS:

- **Registros**: es el lugar donde se almacenan temporalmente los datos que están en movimiento para procesarlos.
- **Buses**: son las uniones entre las distintas unidades, la memoria y los periféricos.


UCP
REGISTROS
UNIDAD
ARITMÉTICA
MEMORIA
PRINCIPAL
UNIDAD
DE E/S
PERIFÉRICO
PERIFÉRICO
PERIFÉRICO
UNIDAD DE CONTROL
PC

UNIDAD DE PROCESO CENTRAL (CPU)

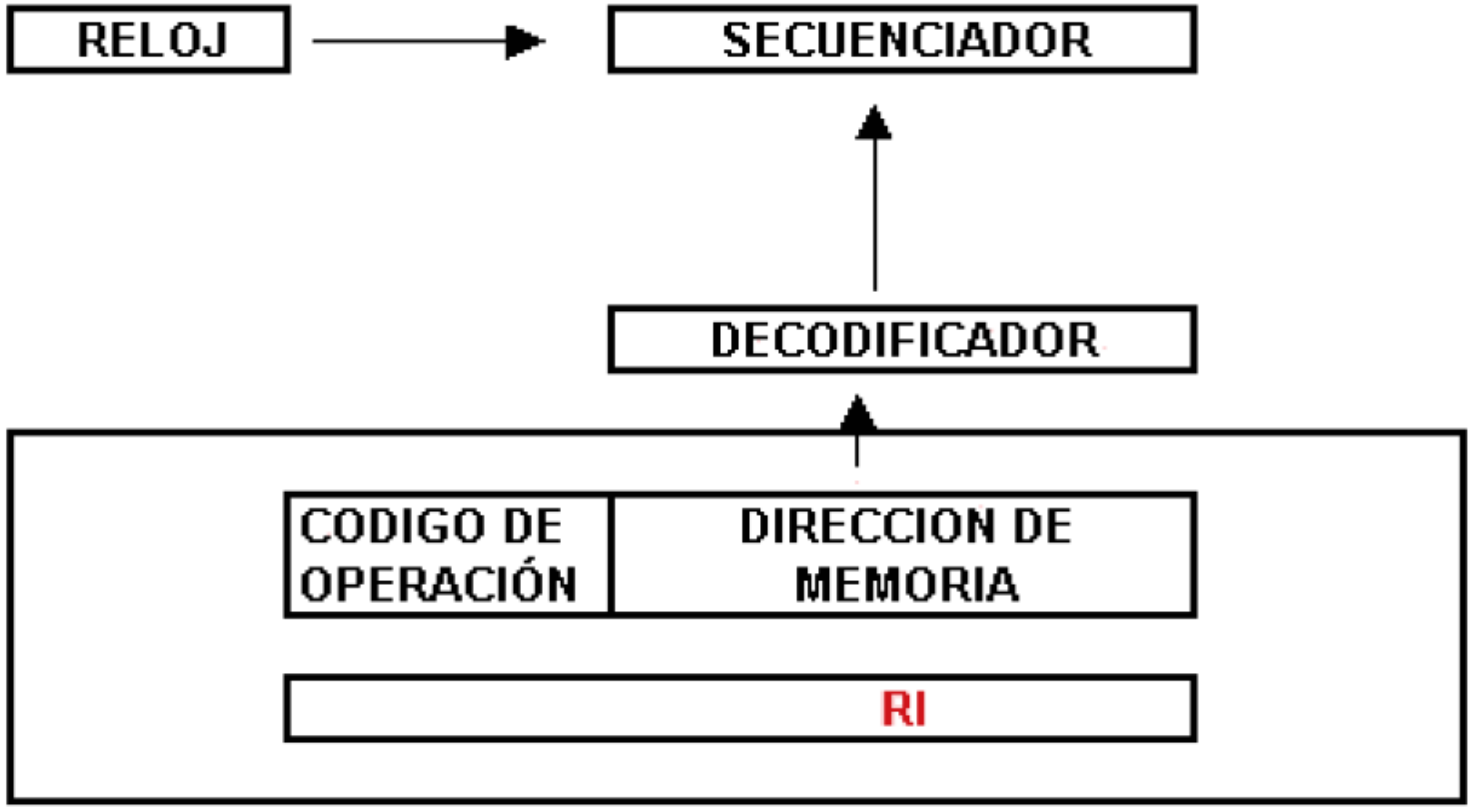
- Es la unidad encargada de controlar y gobernar todo el sistema que comprende una computadora.
- La CPU consiste en un circuito integrado formado por millones de transistores, que está diseñado para procesar datos y controlar la máquina.
- Es un factor clave para la potencia de un ordenador.
- La CPU dispone de dos unidades en su interior: la unidad de control y la unidad aritmético-lógica.



UNIDAD DE CONTROL

- La unidad de control se encarga de leer las instrucciones de los programas almacenados en la memoria.
- Se encarga de enviar las órdenes a los componentes del procesador para que ejecuten las instrucciones.
- El proceso de leer e interpretar instrucciones es el siguiente:
  - Empieza cuando llega una instrucción al **registro de instrucciones**.
  - Llega como una cadena de bits con distintas partes, referidas a la propia instrucción y a los datos que se usarán.
    Ej: 00000100
  - Posteriormente, el **decodificador** interpreta la instrucción a realizar y cómo deben de actuar los componentes del procesador para llevarla a cabo.
  - Esta acción se realiza mediante el **secuenciador**, que envía micró-órdenes marcadas por el **reloj** (el reloj genera pulsos de forma constante, expresados con GHz).

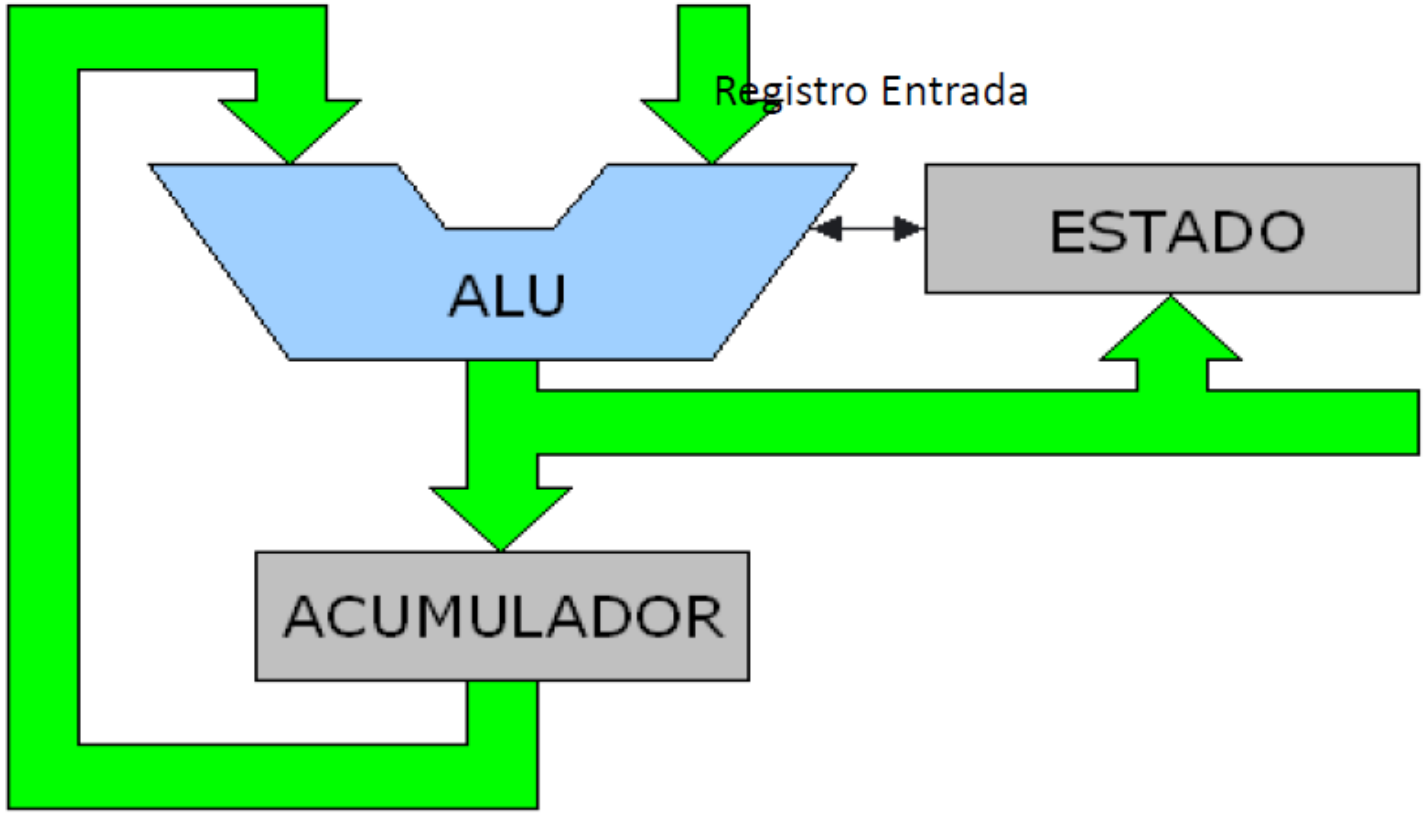
Unidad de control



UNIDAD ARITMÉTICO-LÓGICA (ALU)

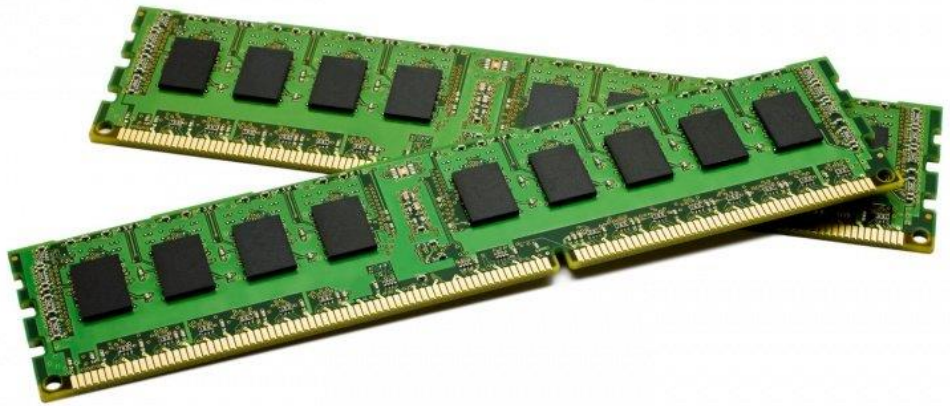
- La ALU es la encargada de realizar todas las **operaciones aritméticas** (sumas, multiplicaciones…) y **lógicas** (comparaciones).
- Esta unidad puede tener distintos diseños.
- Funcionamiento del diseño ALU de la imagen siguiente:
  - Comienza cuando le llega al **registro** de entrada un dato.
  - Posteriormente, la ALU procesa el registro de entrada junto al contenido del **acumulador**.
  - Posteriormente se deposita de nuevo en el acumulador.
  - Repitiendo esta acción se generan los cálculos.

Unidad aritmético lógica (alu)



MEMORIA PRINCIPAL

- La memoria principal en la arquitectura inicial era directamente la RAM, pero esta ha evolucionado y se han añadido memorias caché e implementado algoritmos que predicen que datos vamos a usar más frecuentemente.
- La memoria RAM es bastante sencilla, en comparación con la CPU. Se podría decir que es una tabla que contiene la dirección (o lugar) donde está cierto dato y el contenido del propio dato.
- La unidad de control contiene el **registro contador de programa**, que contiene la dirección de memoria de la siguiente instrucción, que se incrementa tras realizar una instrucción y así va recorriendo la memoria y ejecutando el programa.



Memoria principal

Tabla de memoria

Dir.	Contenido
0000	00000100
0001	00000101
0010	01100111
0011	01110000
0100	00000101
0101	00001011
0110	00000000
0111	00000000

BUSES

Todos estos elementos anteriores se comunican entre sí a través de buses, ya sea para manejar acciones a realizar por la máquina o para mover datos. Hay tres tipos de buses:

- El **bus de datos**, que permite el intercambio de datos.
- El **bus de instrucciones**, transmite las direcciones de memoria que van a ser usadas desde la CPU.
- El **bus de control**, que es el que transporta las órdenes generadas por la CPU para controlar diversos procesos de la máquina.

TIPOS DE CÓDIGO

CÓDIGO FUENTE

- Conjunto de instrucciones ordenadas escritas en un lenguaje de programación

```
1  class BattingAverage
2  {
3    public static void main(String arg[])
4    {
5       int Matches=5,totalruns=200,innings=5,notout=1;
6
7       double avg;
8
9       avg=totalruns/(innings-notout);
10
11      System.out.println("batting average="+avg);
12
13   }
14 }
```

CÓDIGO OBJETO

- Es el código resultante de compilar el código fuente.
- Lenguaje compilado → Código máquina
- Lenguaje virtual → bytecode

```
// Bytecode stream: 03 3b 84 00 01 1a 05
68 3b a7 ff f9
// Disassembly:

iconst_0        // 03
istore_0        // 3b
iinc 0, 1       // 84 00 01
iload_0         // 1a
iconst_2        // 05
imul            // 68
istore_0        // 3b
goto -7         // a7 ff f9
```

CÓDIGO EJECUTABLE

- Código ejecutable resultado de enlazar el código objeto con librerías.
- Es un programa que se ejecuta directamente en el sistema.

MÁQUINAS VIRTUALES

Una máquina virtual es un software que simula un sistema de computación y puede ejecutar programas como si fuese una computadora real.

Los procesos que se ejecutan en la máquina virtual están **limitados por los recursos** y estos procesos no pueden salir de fuera de la máquina virtual.

Uno de los usos más comunes de las máquinas virtuales es el de utilizar otros sistemas operativos para probar su funcionamiento o para realizar ciertas tareas que en nuestro sistema operativo nativo puede no resultar conveniente.

Por ejemplo: utilizamos el sistema operativo Windows para labores ofimáticas y una máquina virtual con una distribución de Linux para programar en lenguajes como Python, C, etc.

TUTORIAL DE INSTALACIÓN DE VIRTUAL BOX Y UBUNTU

https://www.youtube.com/watch?v=hLfVO0GwheU

TIPOS DE 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 éste pueda comunicarse con los dispositivos de hardware y software existentes*»
- Objetivo de los operadores, instrucciones y reglas
  - Facilitar la tarea de crear programas, permitiendo con un mayor nivel de abstracción realizar las mismas operaciones que se podrían realizar utilizando código máquina.

- En un principio, todos los programas eran creados por el único código que el ordenador era capaz de entender: **el código máquina**
  - Un conjunto de 0s y 1s de grandes proporciones.
  - Absolutamente efectivo y sin restricciones, aunque convertía la tarea de programación en una labor sumamente tediosa.
- Se tomó la determinación de dar un nombre a las secuencias de programación más frecuentes.
- Cada una de estas secuencias nominadas se las llamó instrucciones, y al conjunto de dichas instrucciones, **lenguaje ensamblador**.

Código Máquina

Lenguaje Ensamblador

```
C:\temp>debug anzuelo.com
-u 100,128
157F:0100 E91B00        JMP     011E
157F:0103 0D0A53        OR      AX,530A
157F:0106 6F            DB      6F
157F:0107 7920          JNS     0129
157F:0109 756E          JNZ     0179
157F:010B 20434F        AND     [BP+DI+4F],AL
157F:010E 4D            DEC     BP
157F:010F 20696E        AND     [BX+DI+6E],CH
157F:0112 66            DB      66
157F:0113 65            DB      65
157F:0114 63            DB      63
157F:0115 7461          JZ      0178
157F:0117 64            DB      64
157F:0118 6F            DB      6F
157F:0119 2121          AND     [BX+DI],SP
157F:011B 0D0A24        OR      AX,240A
157F:011E BA0301        MOV     DX,0103
157F:0121 B409          MOV     AH,09
157F:0123 CD21          INT     21
157F:0125 B44C          MOV     AH,4C
157F:0127 CD21          INT     21
```

- Mas adelante se empezaron a usar los ordenadores con objetivos científicos distintos a la informática, como la física o la química.
- Resultaba sumamente complicado el uso del lenguaje ensamblador para estos fines.
- Nace el concepto de **Lenguaje de alto nivel**

  p.ej: FORTRAN

```
      PROGRAM PRINCIPAL
        PARAMETER (TAMMÁX=99)
        REAL A(TAMMAX)
 10     READ (5,100,END=999) K
100     FORMAT(I5)
          IF (K.LE.O.OR K.GT.TAMMÁX) STOP
          READ *,(A(I),I=1,K)
          PRINT *,(A(I),I=1,K)
          PRINT * ,, 'SUMA=', SUM(A,K)
          GO TO 10
 99     PRINT * ,, "Todo listo"
        STOP
        END
SUBPROGRAMA DE SUMATORIA EN C
      FUNCTION SUM(V,N)
        REAL :: V(N) ! Declaración de estilo nuevo
        SUM = 0.0
        DO 20 I = 1,N
          SUM = SUM + V(I)
 20       CONTINUE
        RETURN
        END
```

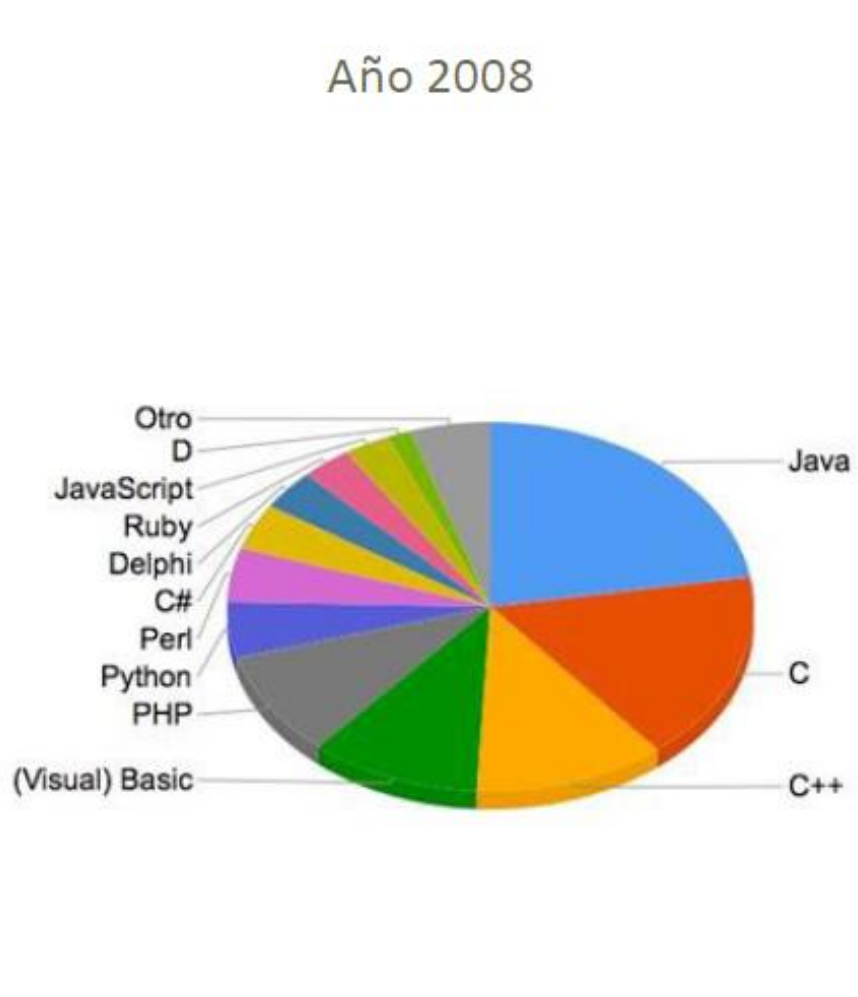
- Los lenguajes de alto nivel son aquellos que elevan la abstracción del código máquina lo más posible, para que programar sea una tarea más liviana, entendible e intuitiva.

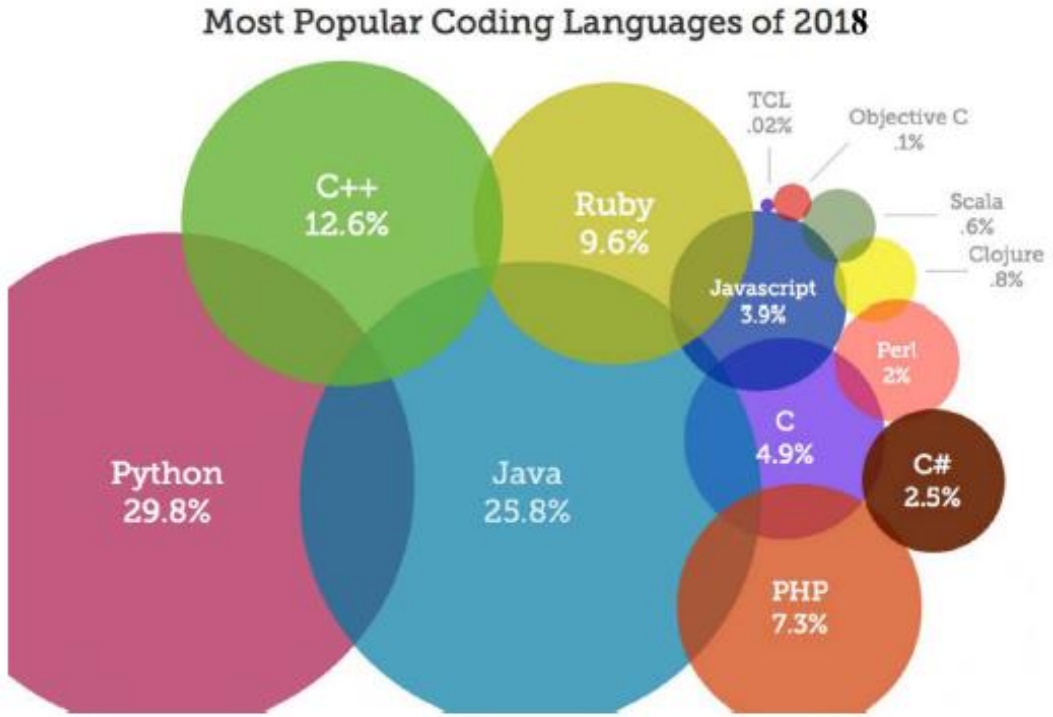
  - No obstante, nunca hay que olvidar que, usemos el lenguaje que usemos, el compilador hará que de nuestro código solo lleguen 1s y 0s a la máquina.

- La cantidad de lenguajes de programación existentes es abrumadora
- Cada uno con unas características y objetivos determinados
- Es necesario establecer alguna clasificación atendiendo a algunos criterios específicos:

- Nivel de abstracción
  - Lenguajes de bajo nivel
    - 1ª generación: *código máquina*
  - Lenguajes de medio nivel
    - 2ª generación: *Ensamblador*
  - Lenguajes de alto nivel
    - 3ª generación: Java
    - 4ª generación: Power Builder Video 4
    - 5ª generación: Prolog Video 5 y Video 6

- Forma de ejecución
  - Lenguajes compilados
  - Lenguajes interpretados
  - Lenguajes virtuales


Año 2008
Otro
D
JavaScript
Ruby
Delphi
C#
Perl
Python
PHP
(Visual) Basic
Java
C
C++


Most Popular Coding Languages of 2018
C++
12.6%
Ruby
9.6%
TCL
.02%
Objective C
.1%
Scala
.6%
Clojure
.8%
Javascript
3.9%
Perl
2%
C
4.9%
C#
2.5%
Python
29.8%
Java
25.8%
PHP
7.3%

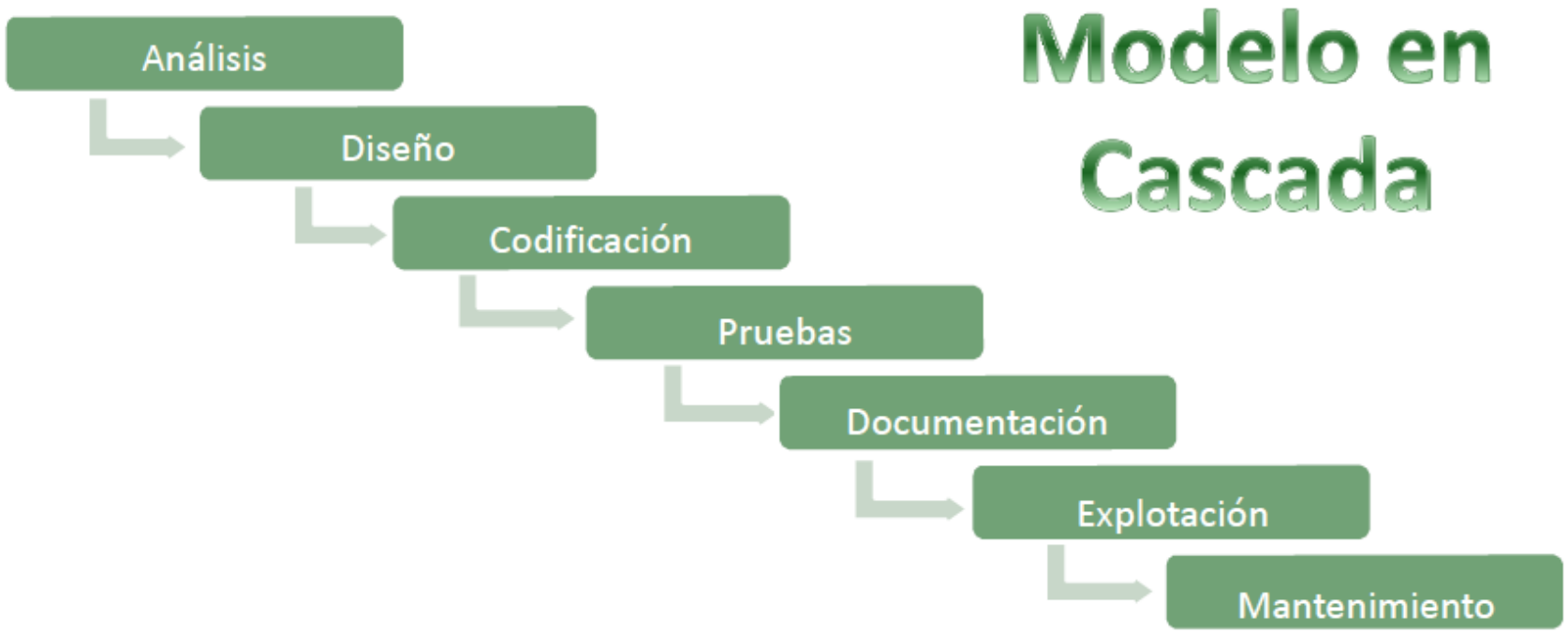
ACTIVIDAD 1.1. – Lenguajes de programación

- Individualmente, elegir un lenguaje de programación y realizar un trabajo que incluya aspectos como:
  - Origen del lenguaje
  - Evolución a lo largo de los años
  - Aplicaciones en el mundo de la informática
  - Sintaxis
  - ¿Dónde se utiliza?
  - Ejemplos de código
  - …

CARACTERÍSTICAS DE LOS LENGUAJES MÁS DIFUNDIDOS

FASES DE DESARROLLO

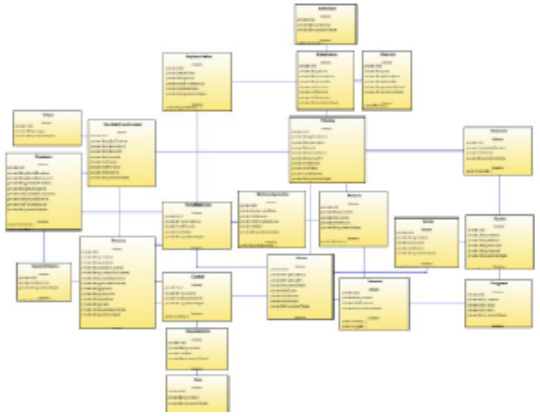
- El desarrollo de un software pasa por muchas etapas desde que se tiene la necesidad de crearlo hasta que está listo para ser usado por el usuario final
- Estas etapas forman el **ciclo de vida del software**
- Existen distintas formas de combinar las etapas, dando lugar a diferentes **modelos de desarrollo**.



Modelo en Cascada

Fase de análisis

- Se definen los requisitos del software que hay que desarrollar
- Inicialmente esta etapa con una entrevista al cliente, que expondrá lo que quiere y necesita
- Es importante tener una buena habilidad y experiencia para reconocer requisitos incompletos, ambiguos, contradictorios o innecesarios.
- Necesario mantener una comunicación bilateral durante este etapa.
- Se crea:
  - ERS (*Especificación de requisitos del sistema*)
  - Diagrama de clases



Fase de DISEÑO

- Se determina el funcionamiento de una forma general, sin entrar en detalles
- Se establecen los recursos del sistema, tanto físicos como lógicos, que se necesitarán para llevar a cabo el software.

Fase de CODIFICACIÓN

- Es la fase donde se programa el software especificado previamente.
- Si se hizo un buen análisis y diseño en las etapas anteriores solo habrá que transformar esto al lenguaje de programación
- Es frecuente que aparezcan nuevos requisitos y haya que hacer un nuevo análisis de ciertas partes.

Fase de pruebas

- En esta fase se realizan las pruebas de los desarrollos del software
- Esta fase tiene doble funcionalidad
  - Confirmar que no existen errores de la fase de codificación
  - Comprobar que el software hace lo que tiene que hacer
- Deseablemente las pruebas son realizadas por personas distintas a las que participaron en el proceso de codificación.

Fase de documentación

- Siempre que sea posible se realizará una doble documentación
  - Documentación para el usuario
    - Información completa y de calidad que ilustre como manejar la aplicación. Debería permitir a un usuario inexperto comprender la aplicación sin conocimiento previo
  - Documentación para el equipo técnico
    - Información completa del funcionamiento interno del programa y su codificación. Debería permitir a un equipo de desarrollo cualquiera entender el programa y modificarlo si fuera necesario.

FASE DE MANTENIMIENTO

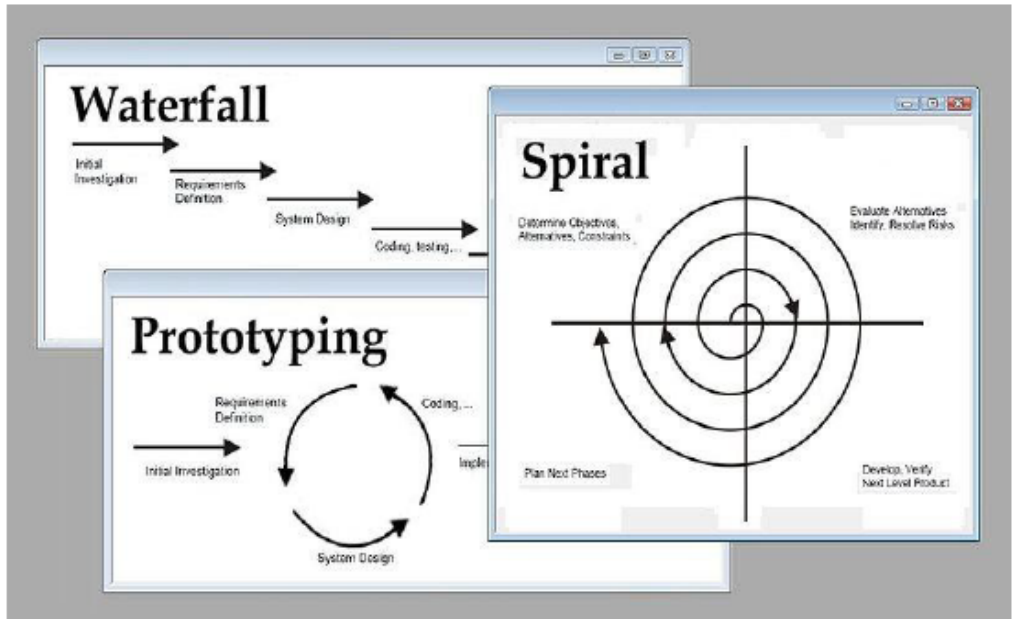
- Se arreglan fallos o errores que suceden una vez el software ya ha sido entregado
- Se actualiza el software para que no vuelvan a ocurrir estos fallos

ACTIVIDAD 1.2. – Modelo en cascada

- Dividirse en 7 grupos
- Cada grupo se encargará de una fase del modelo en cascada
  - Análisis
  - Diseño
  - Codificación
  - Pruebas
  - Documentación
  - Explotación
  - Mantenimiento

Procesos de desarrollo

- No solo existe el modelo de desarrollo en cascada.
- Dependiendo de la naturaleza del problema a implementar se puede elegir otro tipo de metodología de desarrollo
  - Cascada
  - Espiral
  - Incremental
  - Prototipado
  - RAD

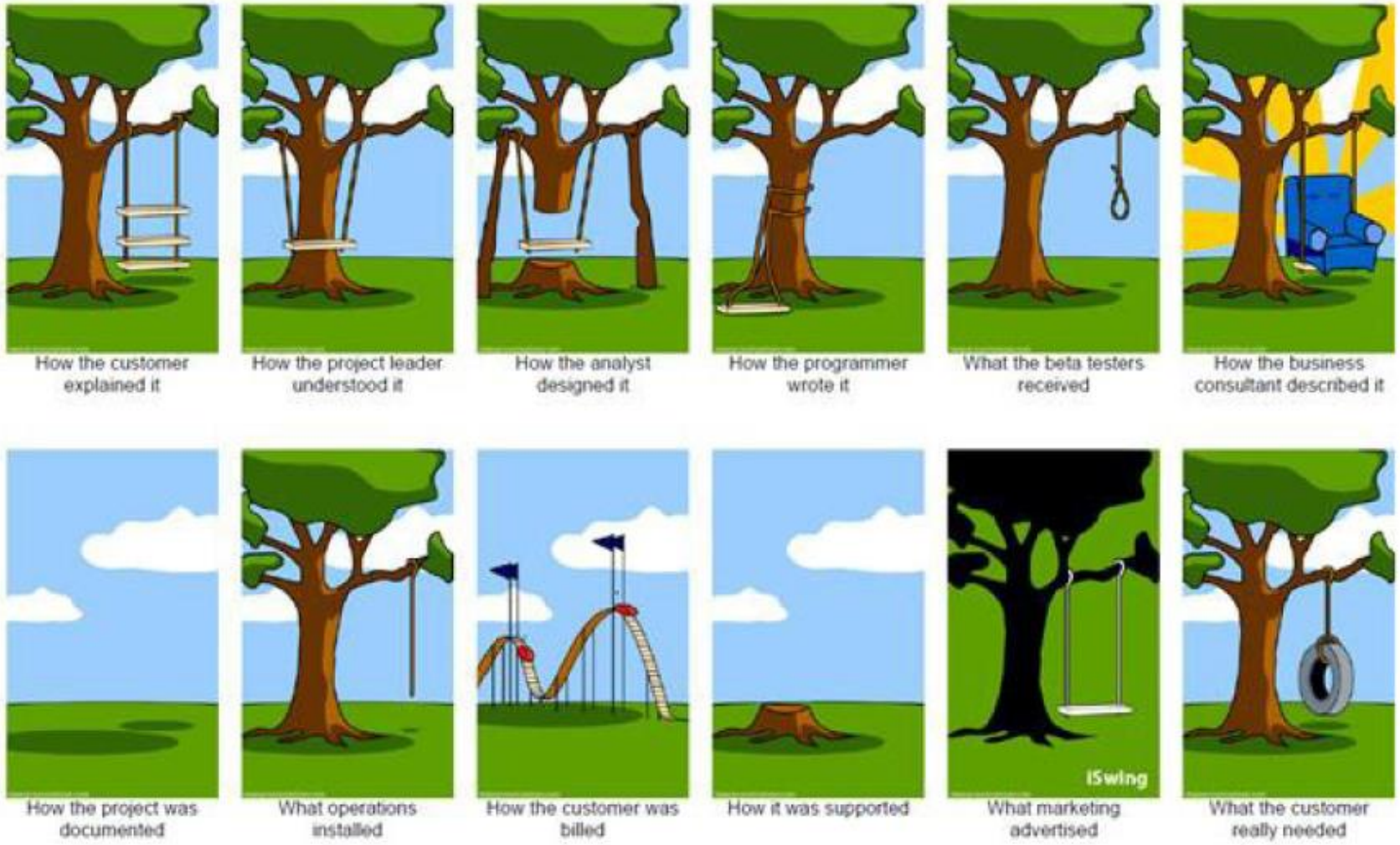


ROLES EN EL DESARROLLO

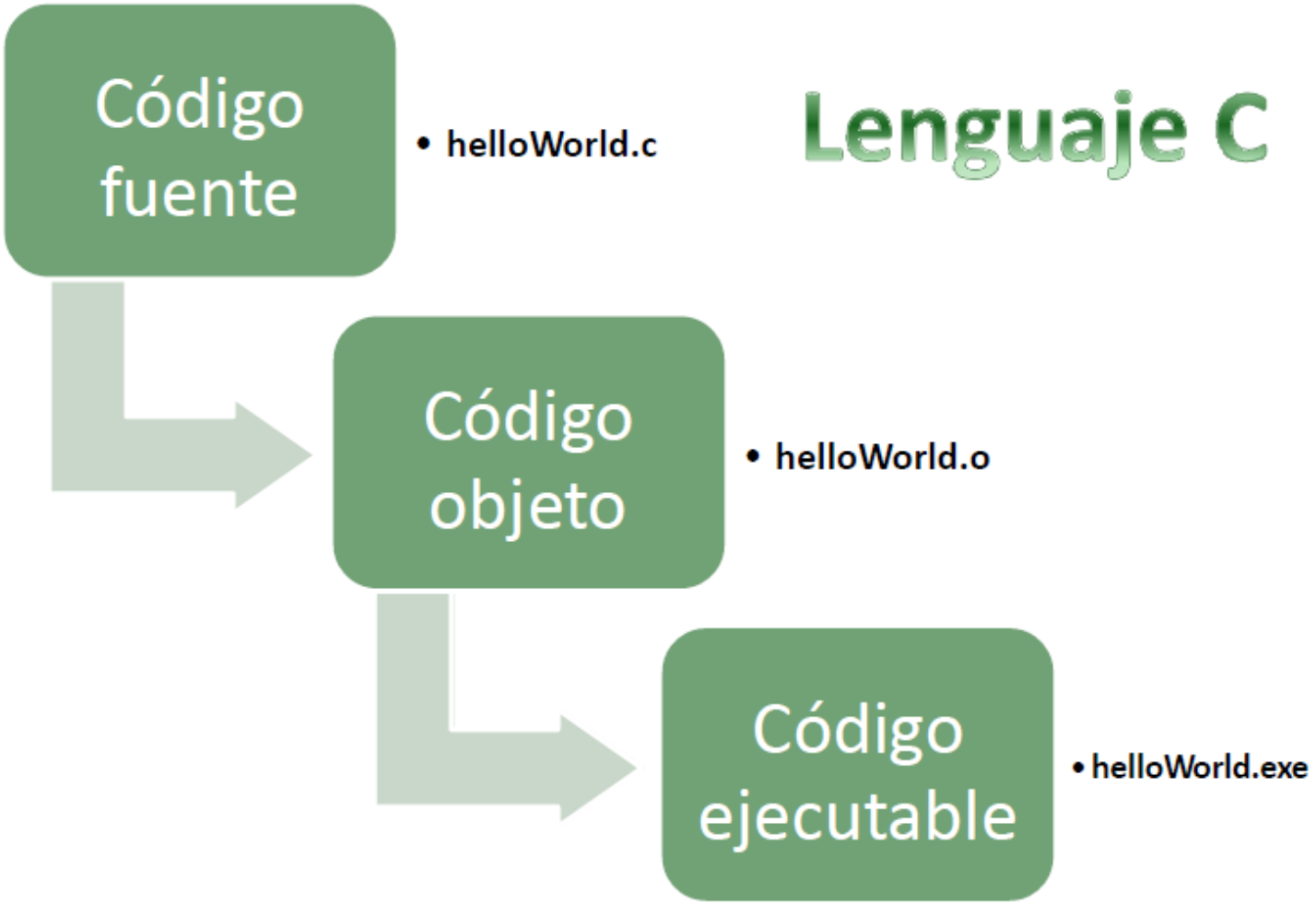
Roles en el desarrollo

- A lo largo de un proceso de desarrollo de un software suelen trabajar diferentes personas
- Suelen cambiar de una fase a otra, aunque se puede dar que una persona pueda participar en varias fases del desarrollo
- Roles:
  - Analista de sistemas
  - Diseñador software
  - Programador
  - Arquitecto de software

Roles en el desarrollo



PROCESO DE OBTENCIÓN DE CÓDIGO EJECUTABLE A PARTIR DEL CÓDIGO FUENTE


Código fuente
• helloWorld.c
Lenguaje C
Código objeto
• helloWorld.o
Código ejecutable
• helloWorld.exe

ACTIVIDAD “ROLES EN LA EMPRESA”

- Dividirse en 3 equipos
- Cada grupo se debe componer de:
  - 1 Analista
  - 1 Arquitecto
  - 1 Diseñador
  - 2 o más Programadores
- Hay premio para el equipo ganador…