

INTRODUCCIÓN

Fundamentos de la Programación

2017



Hardware & Software



HARDWARE

COMPONENTES QUE
INTEGRAN
LA PARTE MATERIAL
DE UNA COMPUTADORA



SOFTWARE

PROGRAMAS,
INSTRUCCIONES
Y REGLAS INFORMÁTICAS
PARA EJECUTAR TAREAS
EN UNA COMPUTADORA

Hardware

Generalmente las computadoras de escritorio están compuestas por El gabinete, el monitor, el teclado y el ratón, pero puedes conectarle más dispositivos, que también se conocen como periféricos. (scanner, impresora, etc)

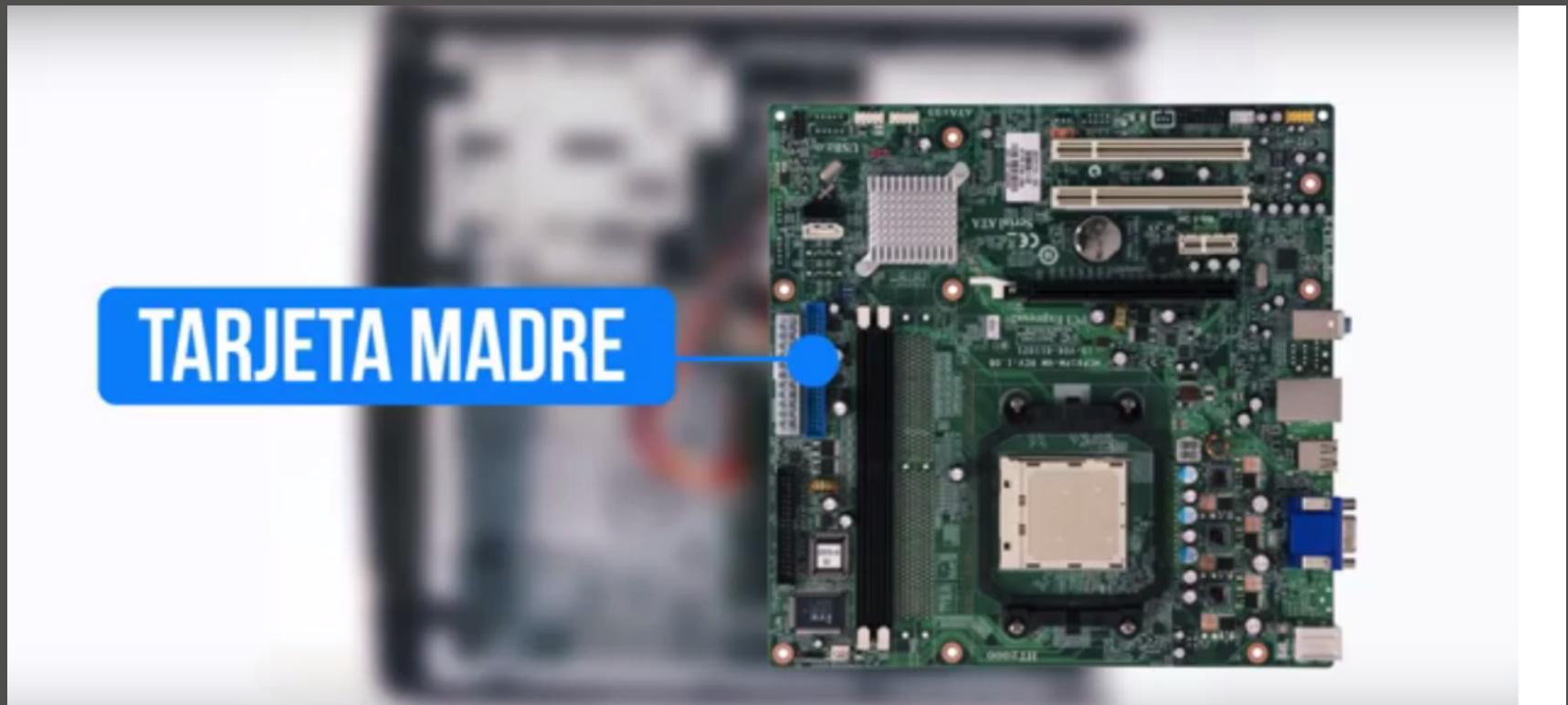


Placa Base

Placa Base o Placa Madre

Su nombre en inglés es motherboard y es la infraestructura de los componentes de tu computador. Es una placa que mantiene funcionando la CPU, la memoria, los conectores para el disco duro y unidades ópticas. También, todas las conexiones a los puertos del ordenador y los puertos USB.

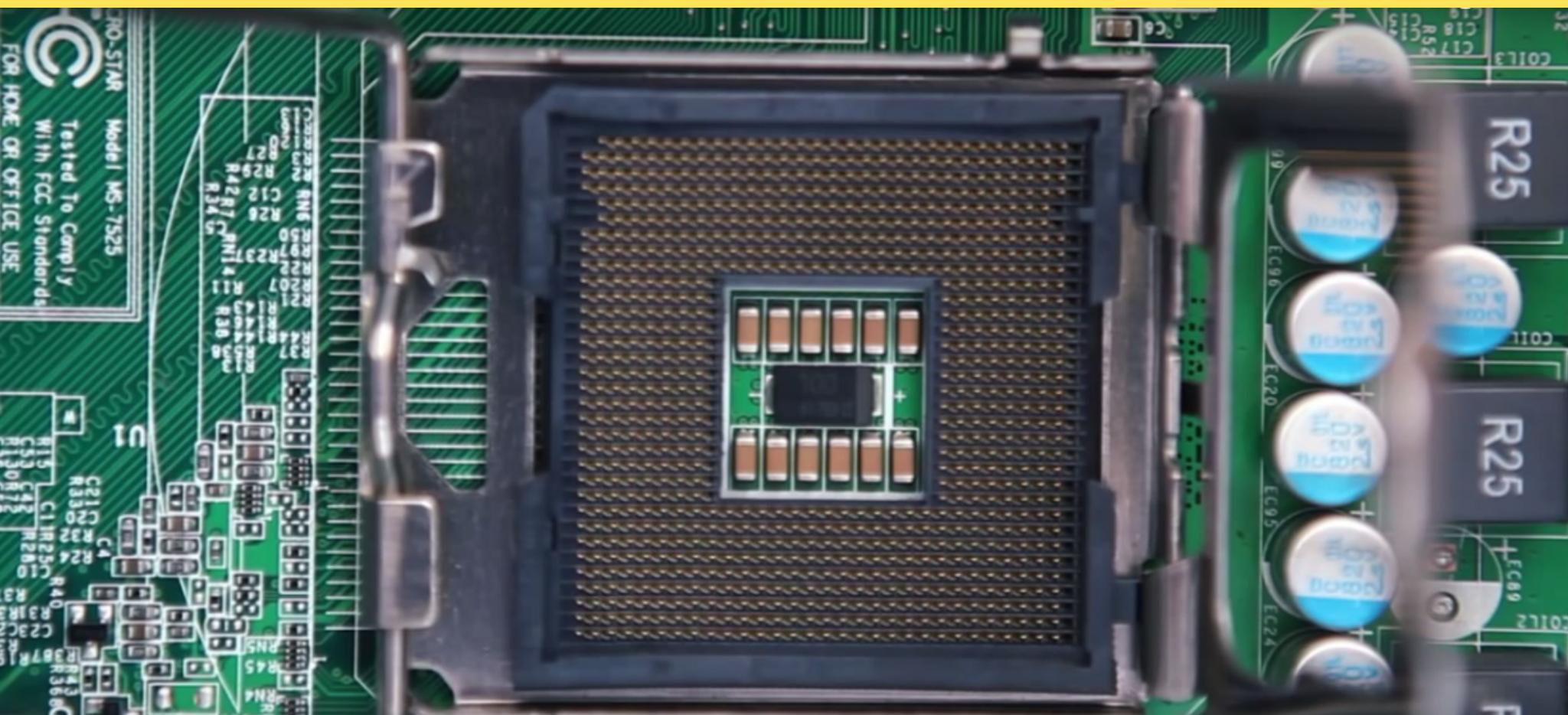
Esta tarjeta pone los límites de actualización a los componentes de tu equipo.



CPU

Unidad Central de Procesamiento CPU es el cerebro del computador. En el CPU se hacen todos los cálculos del código binario de la computadora.

Cada vez que se pulsa una tecla o un clic con el ratón se envía órdenes como instrucciones a la CPU.



Componentes de la CPU

El CPU presenta 2 componentes fundamentales:

La unidad de lógica/aritmética (ALU): es un circuito digital que calcula operaciones aritméticas (como suma, resta, multiplicación, etc.) y operaciones lógicas (si, y, o, no).

Unidad de control (CU): que extrae instrucciones de la memoria, las decodifica y las ejecuta, llamando a la ALU cuando sea necesario.

C.P.U. (Procesador)

A.L.U.
Unidad Aritmético-Lógica

Unidad de Control

PROGRAMAR

- Indicar a la computadora qué es lo que tiene que hacer

PROGRAMA

- Secuencia de instrucciones
Instrucciones que entiende la computadora
Y que persiguen un objetivo:
¡resolver un problema!

#NUESTRO PRIMER PROGRAMA

Actividad (Primitivas):
Vamos a crear una
secuencia de pasos para:

#Preparar el Mate.

#Enviar un whatsapp.



LENGUAJES DE PROGRAMACIÓN

```
function(element, attr, ngSwitchCase) {
    var previousElements = $q.all();
    var selectedElements = $q.all();
    var selectedScopes = $q.all();

    if (attr.$attr.ngSwitch) {
        var previousElements = $q.all();
        var selectedElements = $q.all();
        var selectedScopes = $q.all();

        var previousElementsLength = previousElements.length;
        for (var i = 0; i < previousElementsLength; i++) {
            previousElements[i].remove();
        }
        previousElements.length = 0;

        var selected = selectedElements.length;
        var selectedElement = selectedElements[0];
        var selectedScope = selectedScopes[0];
        if (selected > 0) {
            selectedElement.$destroy();
            selectedElements[0] = selected;
            selectedScope.$destroy();
            selectedScopes[0] = selectedScope;
        }
        selectedElements.leave(selected, function() {
            previousElements.splice(i, 1);
        });
    }

    if (selectedElements.length = 0) {
        selectedScopes.length = 0;
    }
}

selectedTranscludes = ngSwitchController.cases[
    eval(attr.change),
    (selectedTranscludes = function() {
        var selected = selectedElements.length;
        var selectedElement = selectedElements[0];
        var selectedScope = selectedScopes[0];
        if (selected > 0) {
            selectedElement.$destroy();
            selectedElements[0] = selected;
            selectedScope.$destroy();
            selectedScopes[0] = selectedScope;
        }
        selectedElements.leave(selected, function() {
            previousElements.splice(i, 1);
        });
    })
]
```



LENGUAJE MÁQUINA

Aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda entender y ejecutar el programa

LENGUAJES DE BAJO NIVEL (ENSAMBLADORES)

La programación en lenguaje máquina es difícil, por ello se necesitan lenguajes que permitan simplificar este proceso. Los lenguajes de bajo nivel han sido diseñados para ese fin.

LENGUAJES DE ALTO NIVEL

Más cercanos a los lenguajes natural y matemático
 $rdo = dato1 + dato2;$
Mayor facilidad de codificación.
Estructuración de datos.

LENGUAJES DE ALTO NIVEL

```
public class Galleta
{
    private String molde;
    private String receta;
    private String color;

    public Galleta()
    {
        //Constructor
    }

    public Galleta fabricar_galleta()
    {
        //Método para fabricar galletas
    }
}
```

Los lenguajes de alto nivel (C#, JAVA, etc.) son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a la de los lenguajes humanos - en general lenguaje inglés, lo que facilita la escritura y fácil comprensión por el programador.





Lenguajes de Alto Nivel

Los programas escritos por lenguajes de alto nivel **no son entendibles directamente** por la computadora.



Cómo hacemos que la computadora entienda los programas?

Tenemos que traducirlos al **lenguaje máquina** que entienden las computadoras. Los programas que realizan esta traducción se llaman **compiladores**, y los programas escritos en lenguaje de alto nivel se llaman **programas fuente**

A photograph showing a close-up of a person's hands typing on a laptop keyboard. The person is wearing a blue long-sleeved shirt and a black smartwatch on their left wrist. The laptop screen is dark, suggesting it is off or in sleep mode. The background is a plain, light-colored wall.

Compilador

“Compilar” significa **traducir** el código escrito en “Lenguaje entendible por humanos” a un código en “Lenguaje Máquina”, que entienden las máquinas, pero no entendible por nosotros.



Compilador

El compilador se encarga de evitar que se pueda traducir un programa con código fuente mal escrito a código máquina.

Cuando se compila se hacen verificaciones previas para garantizar que se cumple con los estándares de sintaxis obligatorios de un lenguaje.

Compilador

Código fuente

```
[...  
begin  
    writeln(** Calcular la raíz cuadrada de 12  
**);  
    writeln('Entrar x (> 0): ');  
    readln(x);  
    y := sqrt(abs(x)); (* Raíz cuadrada del  
valor absoluto de x para evitar raíces  
imaginarias *)  
    writeln;  
    if (x<0) then (* Si x es negativo, el  
resultado se notifica como imaginario *)  
        writeln('La raíz cuadrada de ', x, ' es el  
número imaginario ', y,'i')  
    else  
        writeln('La raíz cuadrada de ', x, ' es ',  
y);  
    writeln;  
    writeln(** Fin **);  
end.  
...]
```

Archivo: miPrimerPrograma.pas

COMPILEADOR

Código máquina

```
00000010 10000000 00000000  
00000110 2068 10001000 10000000  
01000000 00000010 11001010  
00000001 00000000 00000000 2068  
00010000 10111111 11111111  
1111111110000111 10000000  
11000000 00000011 10000001  
11000011 11100000 000000100  
00000000 00000000 00000000  
00000000 00000000 00000000  
00000111 10111000 00000010  
10000000 00000011 00000011 2068  
10000000 00000011 00000000  
00000000 00000000 00000000  
00000000 00000000 00000000  
00000000 00000000 00000000  
00000000 00000000 00000000  
00000000 00000000 00000000  
00000000 00000000 00000000  
10111000
```

Archivo: miPrimerPrograma.exe

En este esquema, el archivo ejecutable no es válido para cualquier sistema. Por ejemplo, si se ha generado el ejecutable para Windows, no podrá utilizarse en Mac.

El proceso en Java no se corresponde con el gráfico anterior. Pero es un tema que lo veremos más adelante.



Cómo hacemos que la computadora entienda los programas?

Preparar el mate

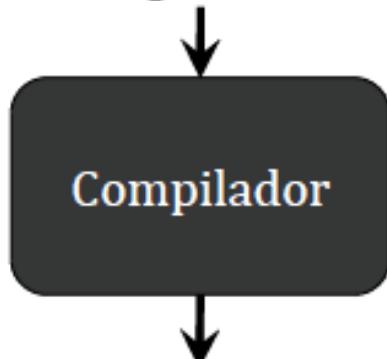
Compilador



Código
Fuente

Código
Objeto

Código fuente



```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hola Mundo!" << endl;
    return 0;
}
```

Código objeto

0100010100111010011100...

Enlazador

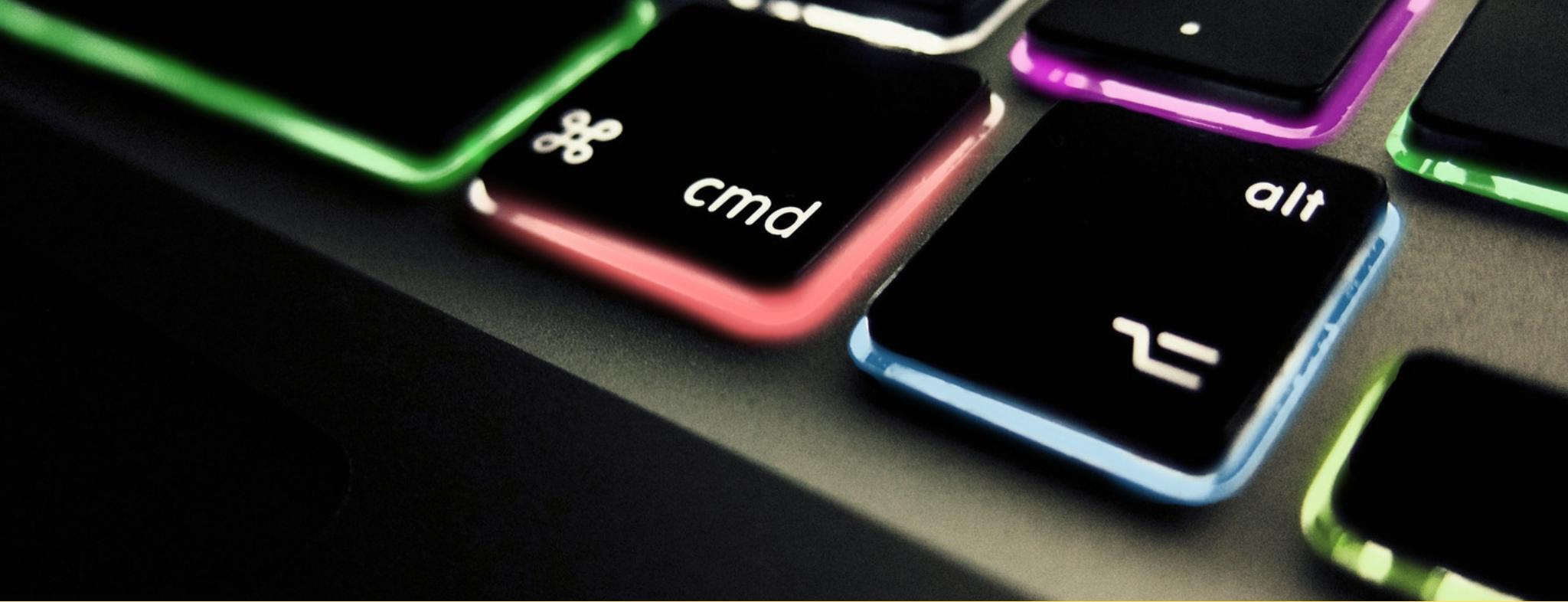
Código
objeto de
biblioteca

Programa
ejecutable

Para una arquitectura concreta
y un sistema operativo



PREGUNTAS?



PARA SEGUIR INVESTIGANDO

- La máquina de Von Neumann
- Como es la arquitectura de la unidad aritmética lógica (ALU) y la unidad de control.
- Como trabaja el compilador de JAVA