

Programación I

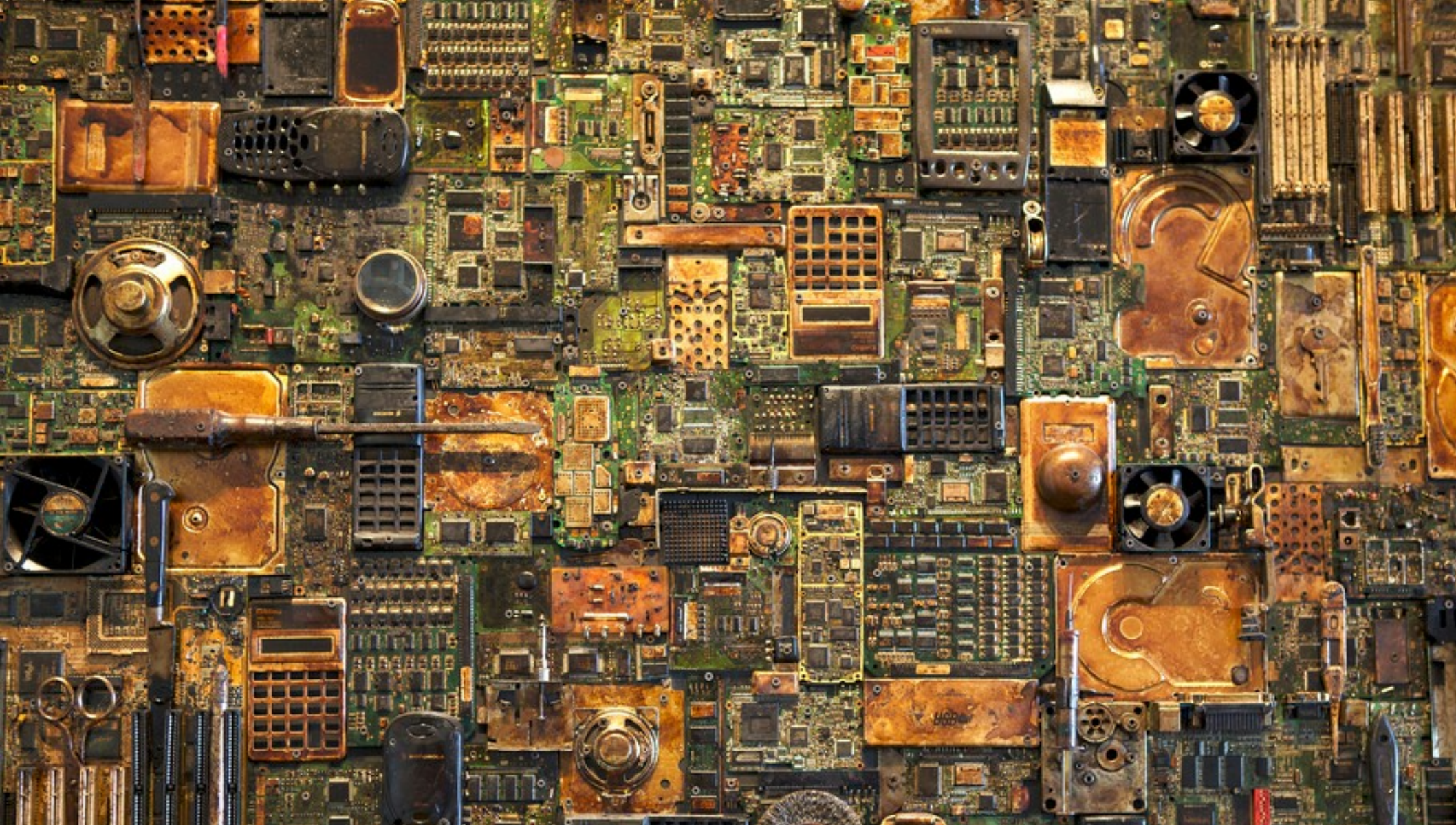
Introducción

Pablo Garaizar Sagarminaga
Borja Sanz Urquijo

Facultad de Ingeniería

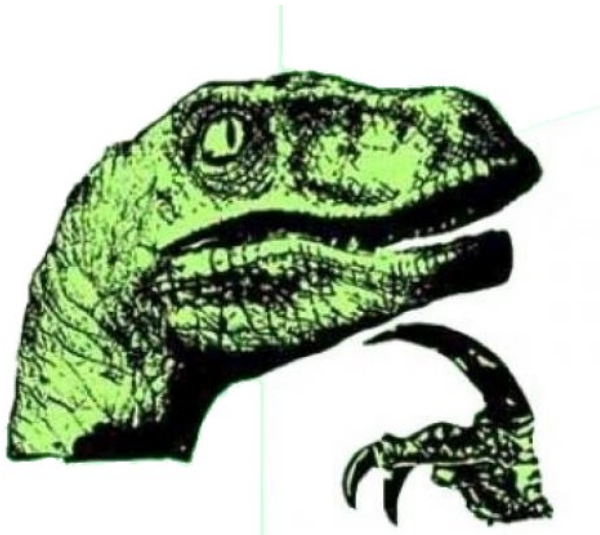
Índice

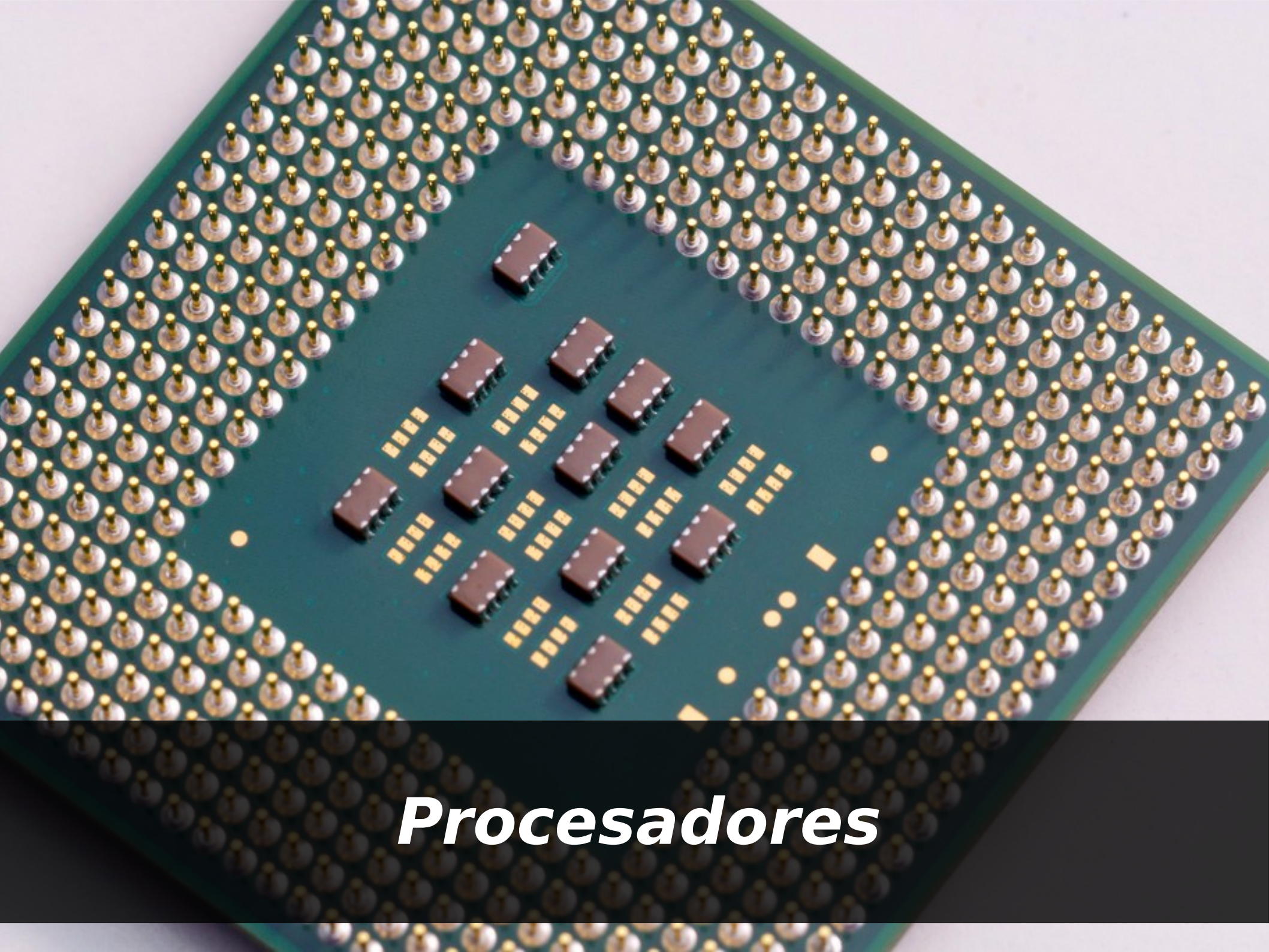
- Estructura de un ordenador
- Sistemas operativos.
- Programación de aplicaciones.



Estructura de un ordenador

¿Qué componentes dentro de un ordenador conoces?



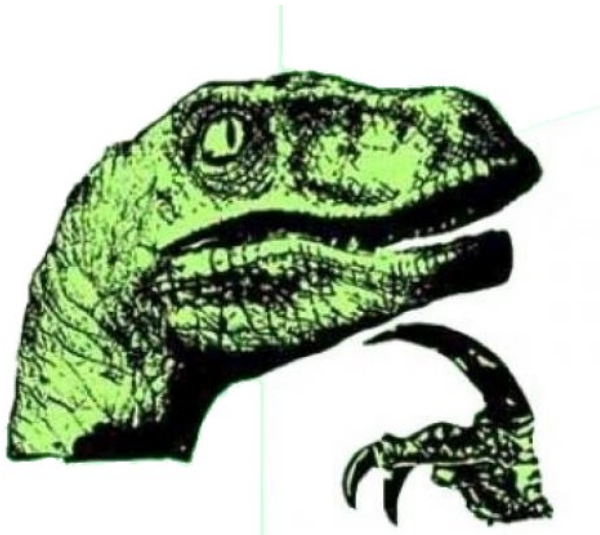


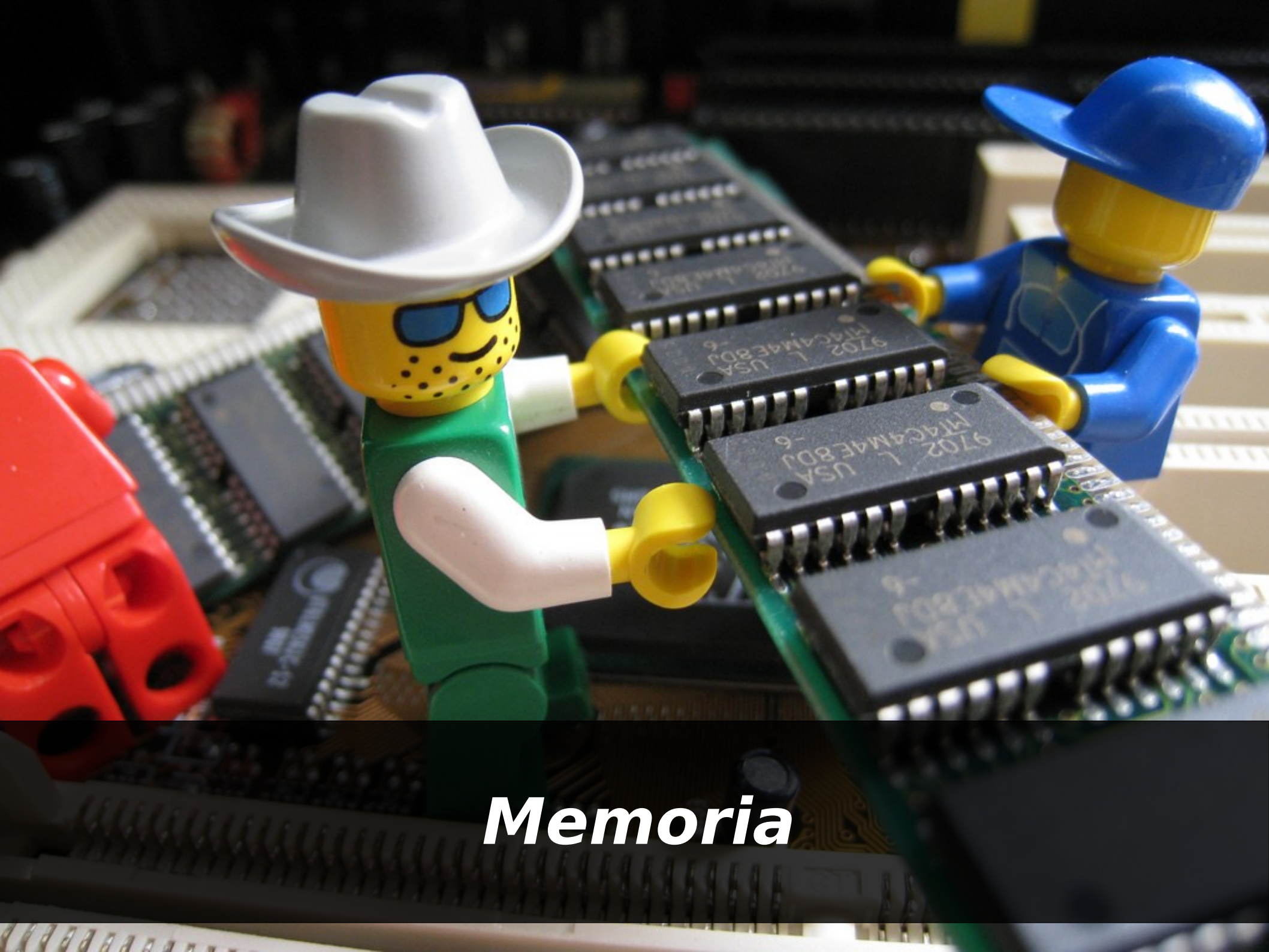
Procesadores

Procesador

- CPU: *Central Processing Unit*.
- Es el componente fundamental de un ordenador.
- Se encarga de ejecutar los programas a partir de instrucciones máquina.
- Realiza:
 - Operaciones lógicas y matemáticas básicas.
 - Accesos a memoria.

***¿Sabes lo que es la
Ley de Moore?***





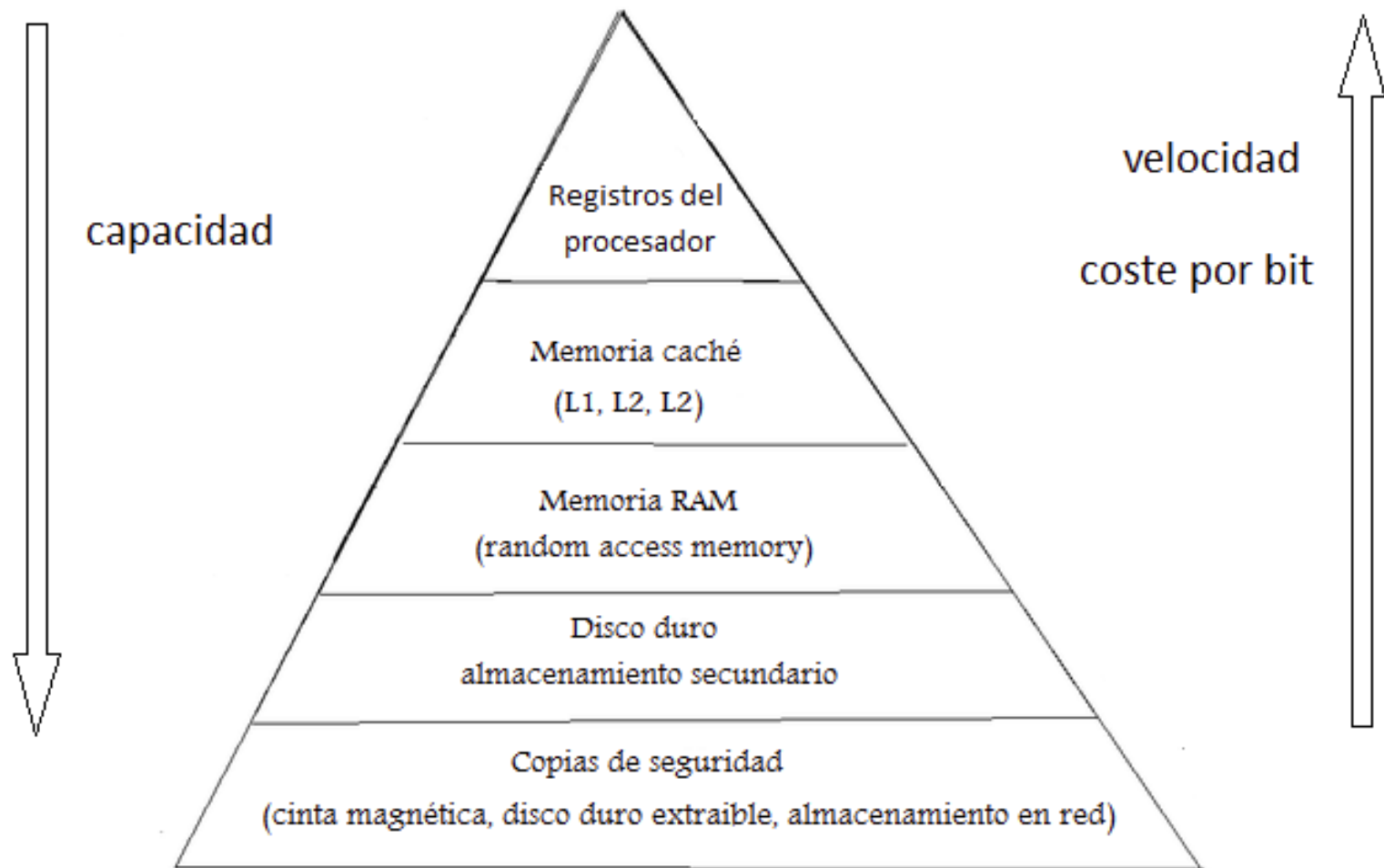
Memoria

Memoria

- Dispositivo que retiene, memoriza o almacena datos informáticos durante algún intervalo de tiempo.
- Memoria primaria:
 - Directamente conectada a la CPU.
 - Contiene programas y datos con los que se operan.
 - Es direccionable (ver bus de direcciones).
- Memoria secundaria:
 - Sistemas de almacenamiento masivo (discos, cintas, etc.).

Memoria

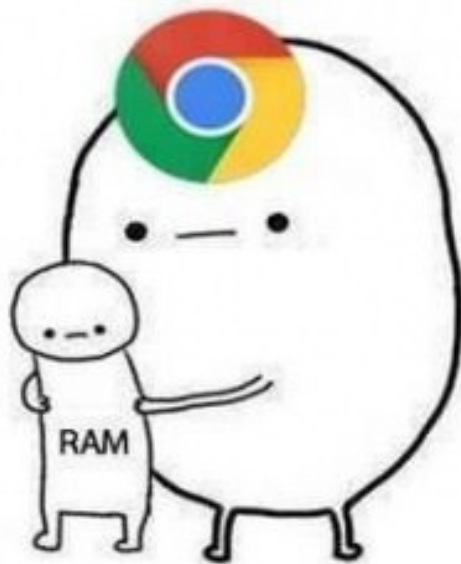
JERARQUÍA DE MEMORIA DEL COMPUTADOR



Memoria

- Tiempo de acceso: nanosegundos (10^{-9} s).

1 Hz	1 s	Latido humano.
6 Hz	150 ms	Latencia de búsqueda en DVD/CD-ROM/Floppy Disk.
100 Hz	10 ms	Latencia de acceso a disco duro.
13 KHz	77 μ s	Acceso a pendrive USB.
75 KHz	13 μ s	Escritura en un SSD.
180 KHz	5 μ s	Lectura en un SSD.
400 MHz	2.5 ns	Latencia típica de acceso a DDR DRAM.
1 GHz	1 ns	Acceso a memoria caché L1.
2 GHz	500 ps	Frecuencia de reloj típica en CPU.

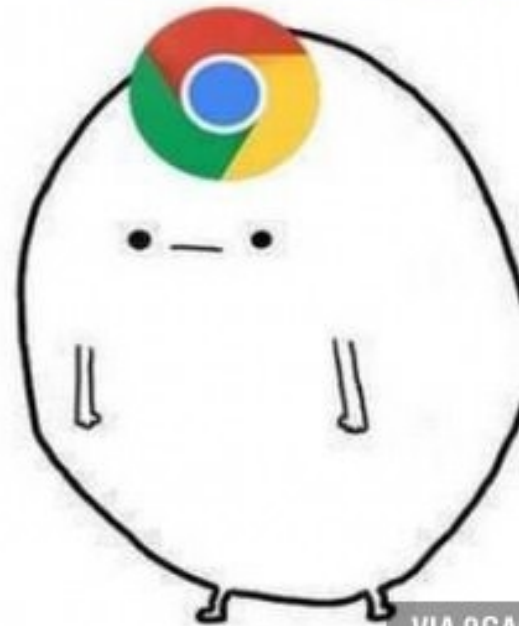


Google Chrome (32 bit)

0.3%

1,984.0 MB

0 M



VIA 9GAG.COM



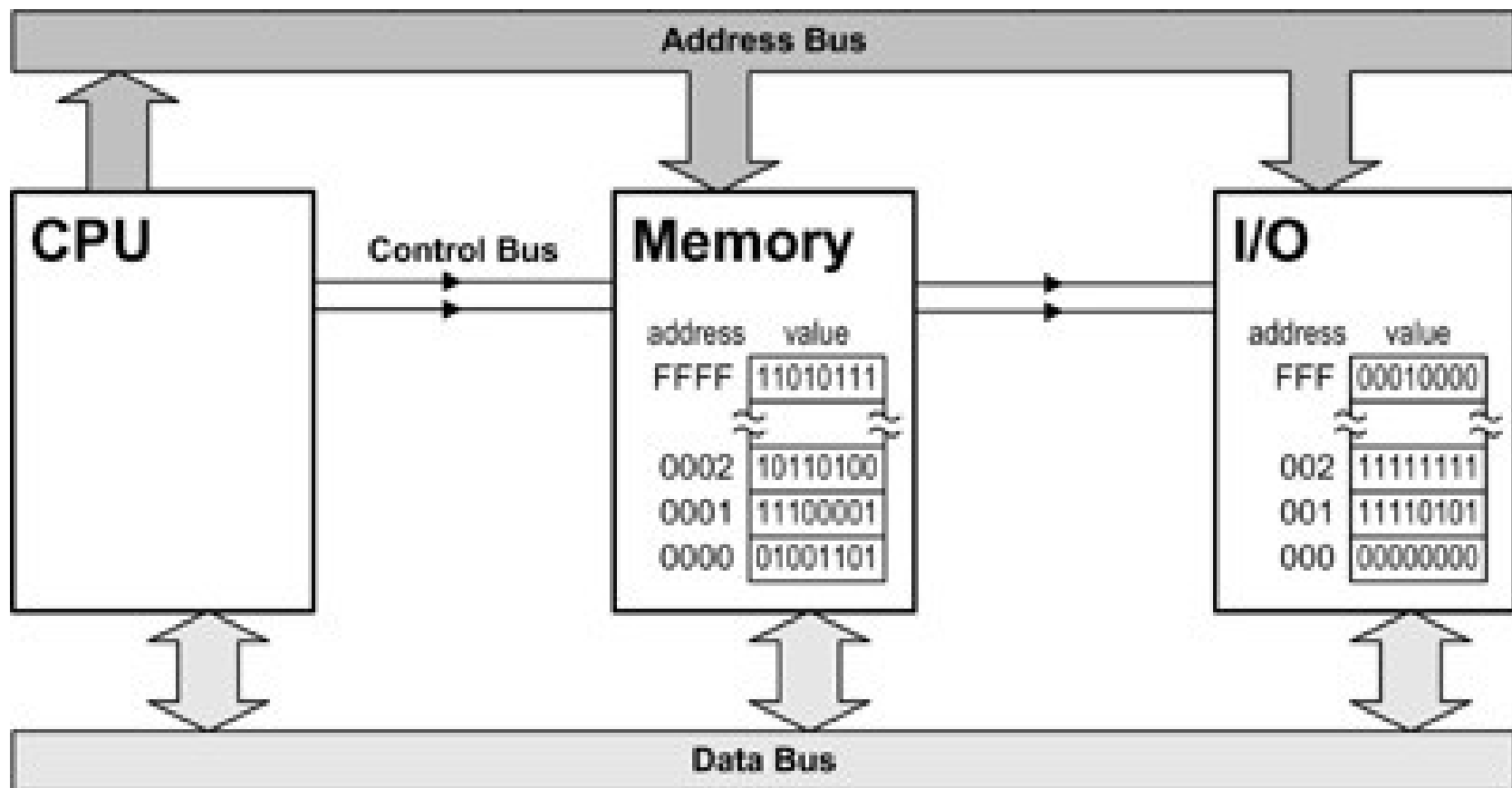


Buses

Buses

- Bus de datos:
 - Transfiere datos entre dispositivos.
- Bus de direcciones:
 - Define la dirección a la que se quiere acceder.
 - Independiente del bus de datos.
- Bus de control:
 - Controla el acceso concurrente a los buses de direcciones y de datos.

Buses



¿Qué le dice un bit a otro?

¡Nos vemos en el bus!





Almacenamiento

Almacenamiento

- Dispositivos magnéticos:
 - Discos duros.
 - Diskettes.
 - Cintas magnéticas.
- Dispositivos de estado sólido.
- Dispositivos ópticos:
 - CD-ROM, CD-R/RW.
 - DVD-ROM, DVD-R/RW.
 - Blu-ray.
- Otros.

A close-up, low-angle shot of a computer keyboard. The keys are dark with white lettering. The keyboard is illuminated with a strong blue light, creating a high-contrast, futuristic aesthetic. The focus is sharp on the keys in the foreground, while the background keys are blurred. The text "Dispositivos de entrada" is overlaid at the bottom in a white, bold, italicized font.

Dispositivos de entrada

Dispositivos de entrada

- Teclado.
- Ratón.
- Pantalla táctil.
- Escáner.
- Cámara.
- Micrófono.
- Otros.



Dispositivos de salida

Dispositivos de salida

- Pantallas.
 - CRT (*Cathode Ray Tube*).
 - TFT-LCD (*Thin Film Transistor-Liquid Crystal Display*).
- Impresoras.
- Altavoces.

Periféricos de comunicación

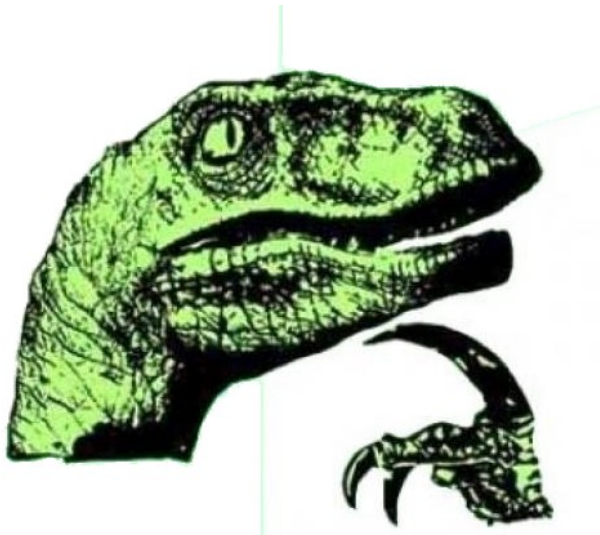
- Tarjetas de red (Ethernet / Wi-Fi).
- Módems.
- Concentradores de red (*hubs*).
- Conmutadores de red (*switches*).
- Enrutadores (*routers*).
- Bluetooth / Zigbee.
- Infrarrojos (IR).
- NFC.

FLATRON

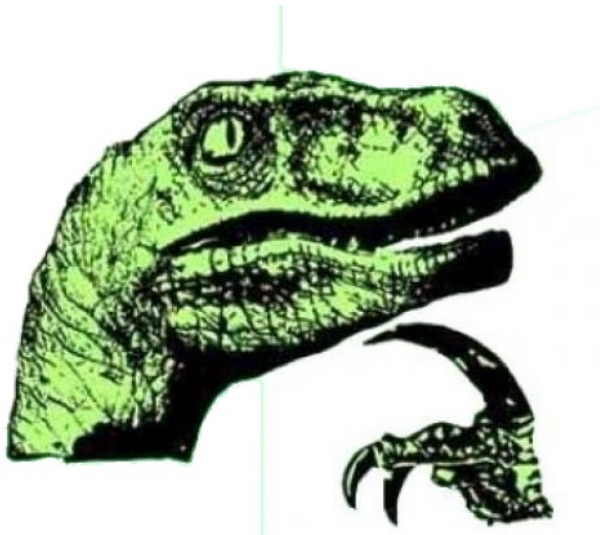
Missing operating system_

Sistemas Operativos

***¿Qué sistemas operativos
conoces?***



¿Cuál es el propósito de un sistema operativo?



Sistemas operativos





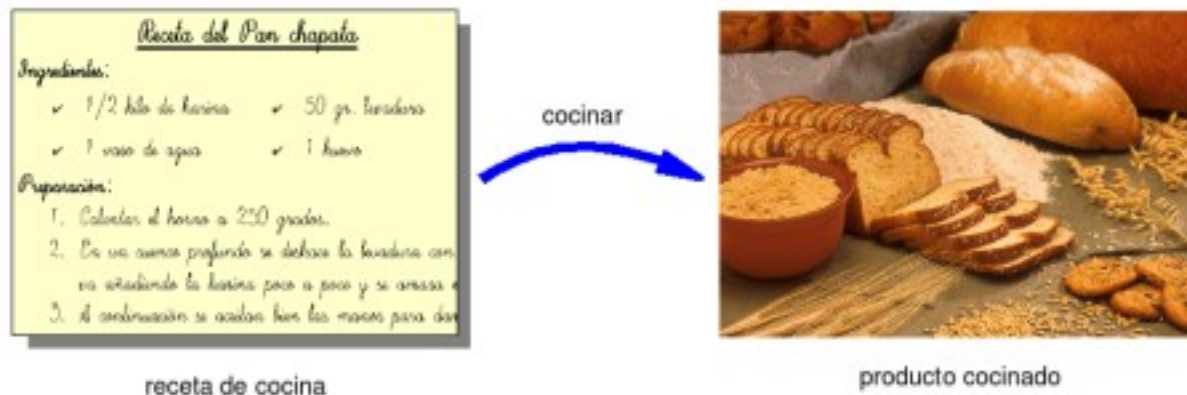
Programación de aplicaciones

La metáfora de la receta

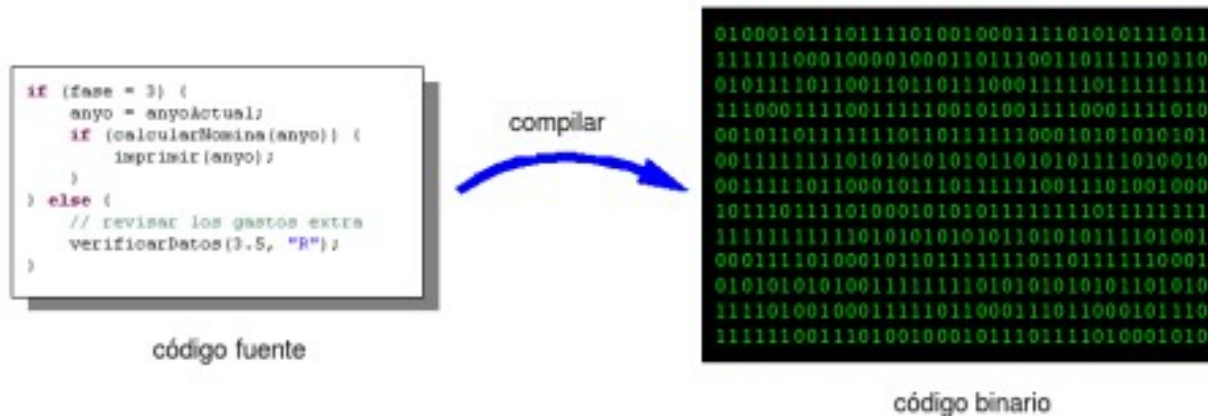
- En gastronomía:
 - Una persona piensa una **receta** para hacer un **pastel** usando un **horno**.
- En informática:
 - Una persona programa un **código fuente** para hacer una **aplicación** (código **binario**) usando un **compilador**.

La metáfora de la receta

- En gastronomía:



- En informática:





Fuente → Compilador → Binario

Lenguajes de programación

- Hay **diferentes** formas de definir el código fuente:
 - C, C++, Java, Python, Perl, PHP, Ruby, etc.
- Basta con que haya un **compilador** que **traduzca** una de esas formas a **código binario ejecutable** para que ese lenguaje funcione.



Python

Python

- Creado a finales de los 80 por **Guido van Rossum**.
- Software **libre**.
- **Multiplataforma**: Windows, Linux, MacOS, etc.
- **Multiparadigma**: imperativo, orientado a objetos, funcional, etc.
- **Interpretado**.
- Tipado **fuerte y dinámico**.
- Código **legible**.

Days 1 - 10

Teach yourself variables, constants, arrays, strings, expressions, statements, functions,...



Days 11 - 21

Teach yourself program flow, pointers, references, classes, objects, inheritance, polymorphism,



Days 22 - 697

Do a lot of recreational programming. Have fun hacking but remember to learn from your mistakes.



Days 698 - 3648

Interact with other programmers. Work on programming projects together. Learn from them.



Days 3649 - 7781

Teach yourself advanced theoretical physics and formulate a consistent theory of quantum gravity.



Days 7782 - 14611

Teach yourself biochemistry, molecular biology, genetics,...



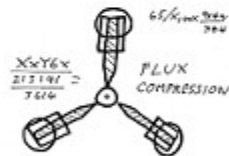
Day 14611

Use knowledge of biology to make an age-reversing potion.



Day 14611

Use knowledge of physics to build flux capacitor and go back in time to day 21.

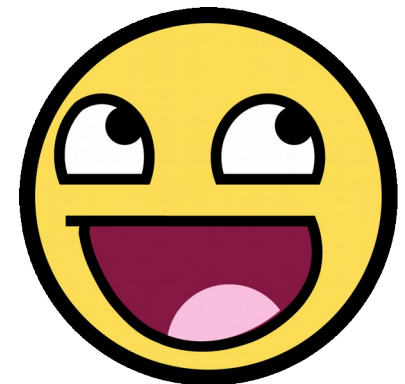


Day 21

Replace younger self.



As far as I know, this is the easiest way to "Teach Yourself C++ in 21 Days".





***The art of
computer programming***

Referencias

- Intel® 64 and IA-32 Architectures, Software Developer's Manual, Volume 1: Basic Architecture,
<http://www.intel.la/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-1-manual.pdf>
- Wikipedia.

Referencias

- Imágenes:

- [Wikipedia](#)
- Ikusi makusi, Marko Txopitea.
- <http://abstrusegoose.com>
- <https://www.flickr.com/photos/schill/6890892084>
- <https://www.flickr.com/photos/gbpublic/19872841042>
- <https://www.flickr.com/photos/jmsmith000/3097202394>
- <https://www.flickr.com/photos/jeroenbennink/3382865257>
- <https://www.flickr.com/photos/uwehermann/3417729666/>
- http://www.electronics.dit.ie/staff/tscarff/architecture/neumann_harvard.gif