

# Fundamentos de Ingeniería Informática

---



# Objetivos



- ¿Qué es el Software?
- Tipos de Software
- Creación de Software
- Niveles de programación
- Compiladores vs. Interpretes

*NOTA: Todas las imágenes de esta presentación están extraídas de internet*

# —

# ¿Qué es el software?



# ¿Qué es el software?



**Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.**

- En lugar de manipular interruptores o cables, los programadores escriben programas y los introducen en la memoria del computador, donde se almacenan como secuencias de 0 y 1.
- El concepto de **programa almacenado** apareció en 1945, y se le atribuye a John Von Neumann.

# Tipos de software: SEGÚN SU FIN



## Software de aplicación

- Programas y utilidades que cumplen una tarea específica.
- Producto final.
- Permiten que una máquina de uso general se convierta en un sistema de uso específico

## Software de programación

- Utilidades digitales que permiten a los programadores desarrollar programas informáticos utilizando diferentes lenguajes de programación o bases de datos.
- Ej. Editores de texto, compiladores, intérpretes, enlazadores, depuradores,...

## Software del sistema

- Permite al usuario usar la interfaz de SO.
- Conecta las aplicaciones con los recursos de HW
- Ej. Contraladores, herramientas de diagnóstico, SO,...

# Tipos de software: ENTORNO



de escritorio



app



web

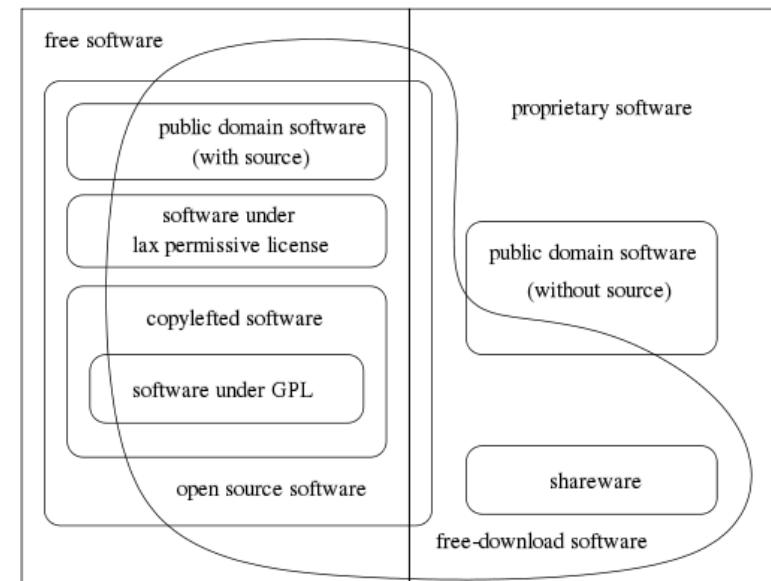


“en la nube”

# Tipos de Software: Licencia/derechos de autor



- Cuando se adquiere un programa se adquiere en realidad el derecho a usarlo, es decir, una **licencia**.
- El acuerdo que se acepta al utilizar un programa se denomina **EULA** (End User License Agreement)
- Según el Proyecto GNU
  - Freeware. Software gratuito, protegido por derechos de autor.
  - Shareware. Uso con limitaciones.
  - Software libre. Permite copiarse, modificarse y distribuirse.
  - Software de código abierto. Propiedad intelectual compartida.
  - Software privativo. No es libre, pertenece a una empresa.
  - Software comercial. Su finalidad es generar ganancia económicas.



# Creación de Software



## Metodología

- Describe cómo es el Proceso de Elaboración del Software, desde la idea o necesidad, hasta la entrega del producto final.
- INGENIERÍA DEL SOFTWARE.

## Tecnología

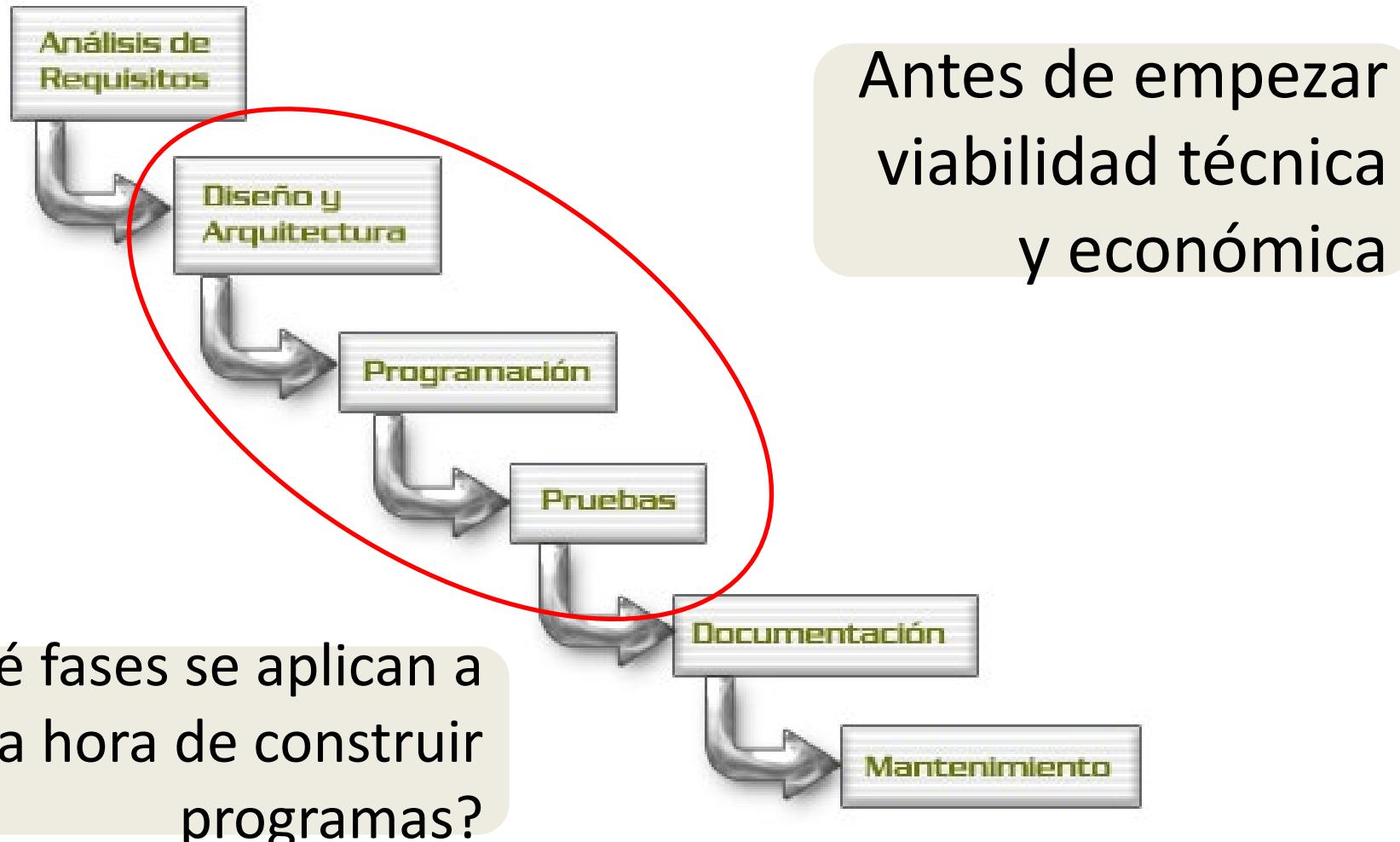
- Conjunto de herramientas (técnicas) utilizadas en cada etapa para producir los resultados esperados.
- DESARROLLO Y PROGRAMACIÓN.

```
admission_extension_online_controller.py
1 # -*- coding: utf-8
2 from odoo import http
3 from odoo.exceptions import ValidationError
4 from datetime import datetime
5 import calendar
6 import math
7 import pytz
8 import io, base64
9
10 class AdmissionExtensionOnlineController(http.Controller):
11     @http.route('/get/type_wise_program', website=True, auth='public')
12     def type_wise_program(self, **kwargs):
13         if len(kwargs['types'])<=0:
14             return "None"
15
16         types = kwargs['types']
17         program_list = []
18         domain = []
19
20         if types == 'local_bachelor_program_hsc':
21             domain = [('course_id.is_local_bachelor_program','=',True)]
22         elif types == 'local_bachelor_program_a_level':
23             domain = [('course_id.is_local_bachelor_program','=',False), ('course_id.is_a_level','=',True)]
24         elif types == 'local_bachelor_program_diploma':
25             domain = [('course_id.is_local_bachelor_program','=',False), ('course_id.is_diploma','=',True)]
26         elif types == 'local_masters_program_bachelor':
27             domain = [('course_id.is_local_masters_program','=',False), ('course_id.is_bachelor','=',True)]
28         elif types == 'international_bachelor_program':
29             domain = [('course_id.is_international_bachelor','=',True)]
30         elif types == 'international_masters_program':
31             domain = [('course_id.is_international_masters','=',True)]
32         elif types == 'international_masters_program':
33             domain = [('course_id.is_international_masters','=',True)]
34
35         admission_register_list = http.request.env['admission.register'].search(domain)
36         admission_register_list = admission_register_list.sorted(['state'])
37         admission_register_list = admission_register_list.filtered(lambda x: x.state != 'application')
38
39         for program in admission_register_list:
40             if program.state == 'application':
```

# Creación de Software. Ciclo de vida clásico.



## Fases del Proceso de Desarrollo del Software

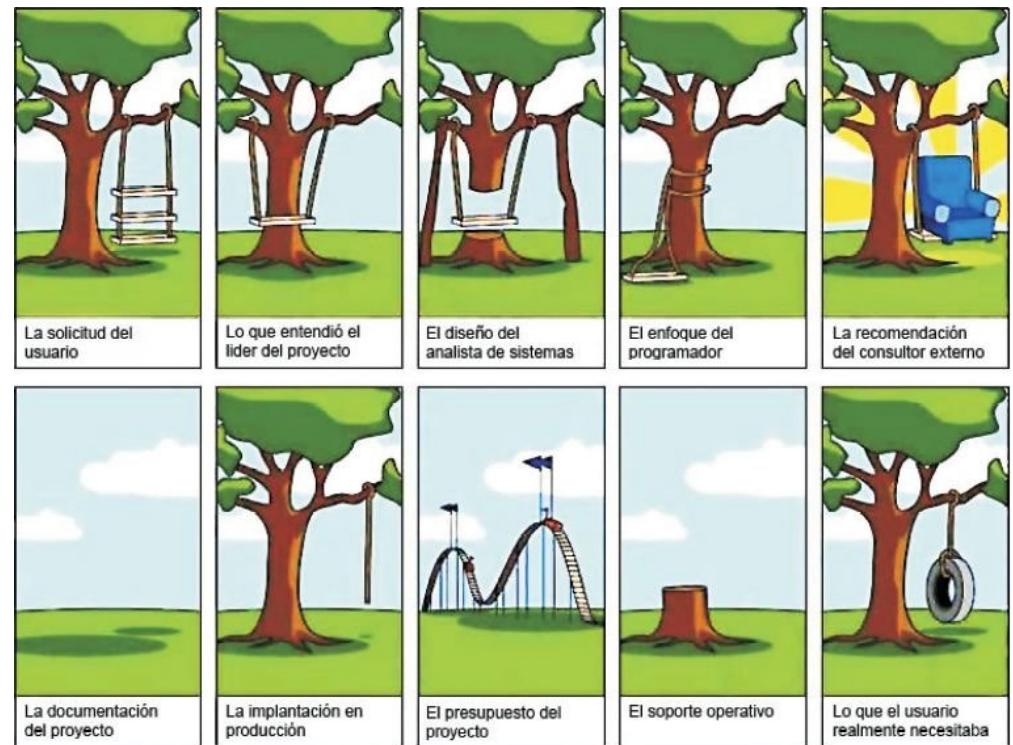


# Creación de Software. Ciclo de vida clásico.



## Análisis de requisitos ¿Qué se quiere hacer?:

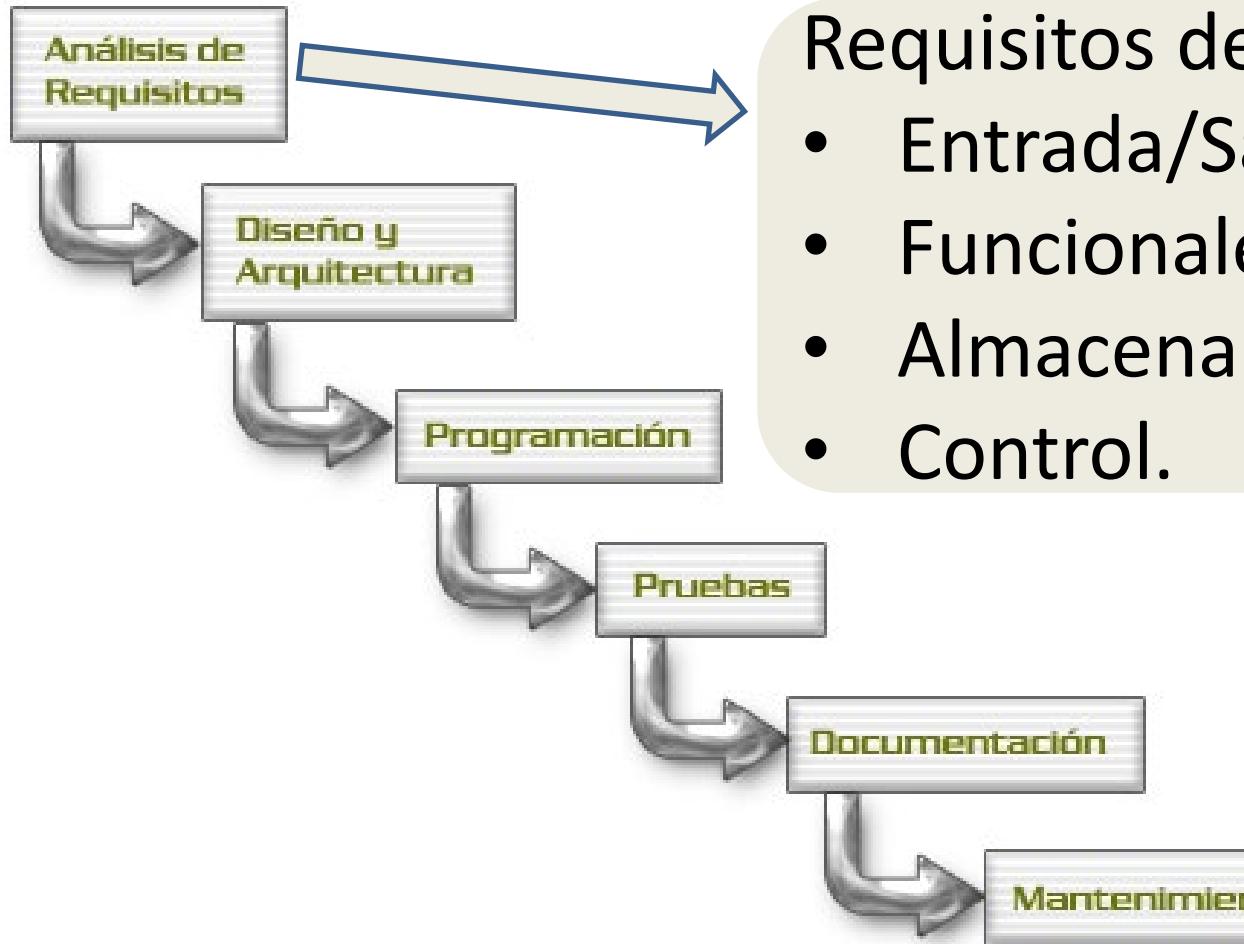
- En esta fase se analizan las necesidades del cliente y de los usuarios finales del software para determinar **qué objetivos** debe cubrir.
- El resultado de esta fase son dos documentos que contienen la especificación completa de lo **QUÉ** se debe hacer:
  - **URD** (User Requirement Document)
  - **SRD** (Software Requirement Document)
- Es importante señalar que los requisitos deben estar bien definidos y **recoger todas las necesidades del sistema**, dado que son el punto de partida para el resto del proceso.



# Creación de Software. Ciclo de vida clásico.



## Fases del Proceso de Desarrollo del Software

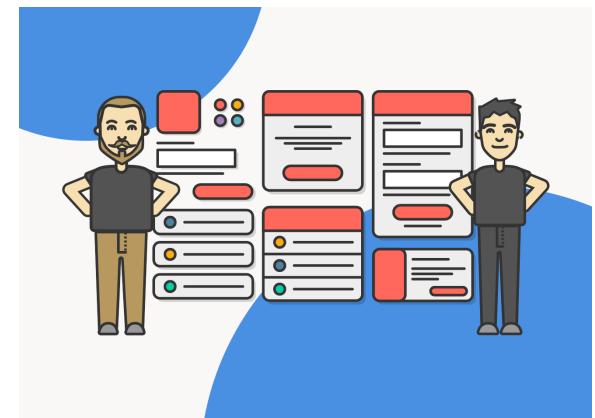


# Creación de Software. Ciclo de vida clásico.



## Diseño del Sistema. ¿Cómo se hará?

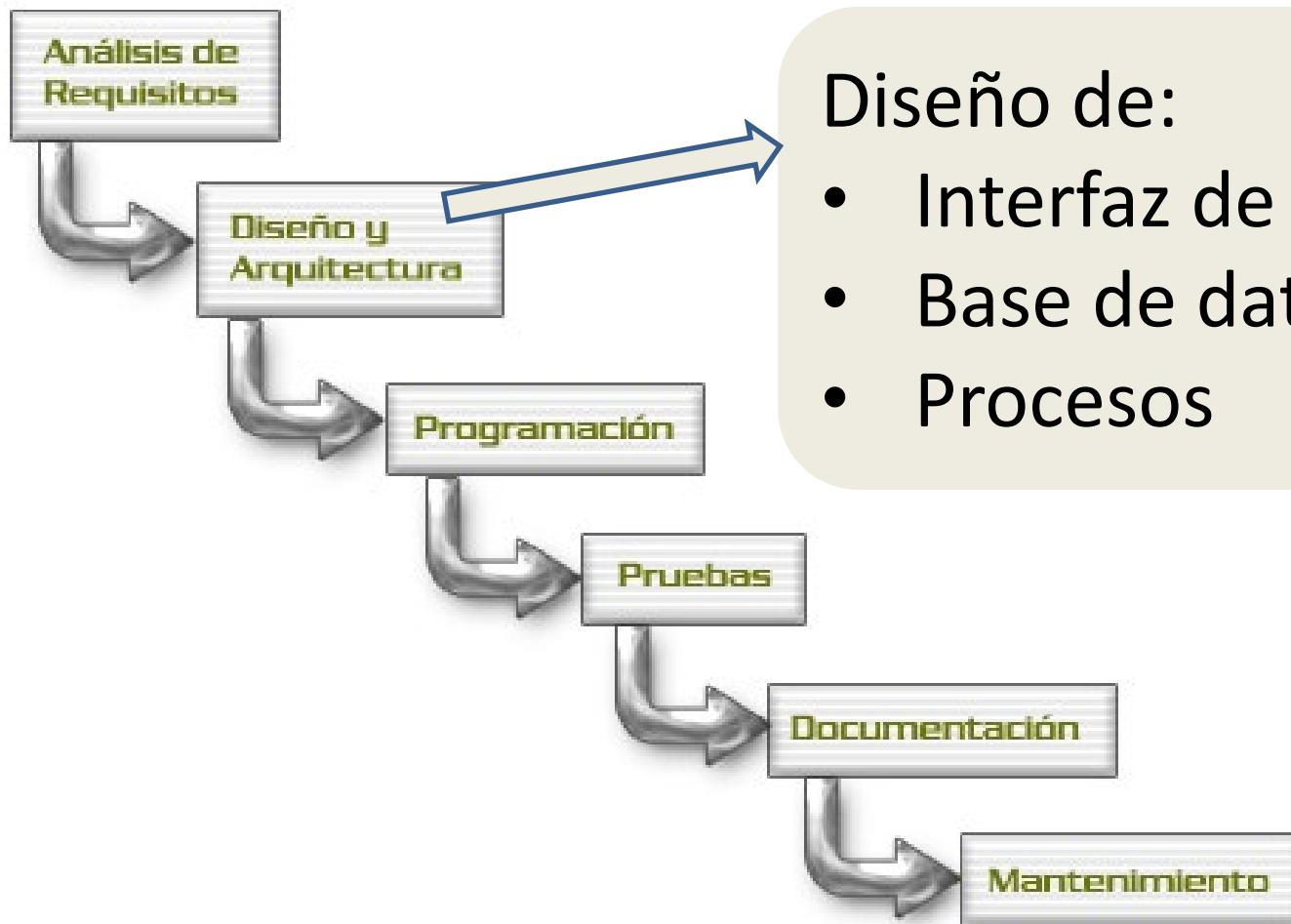
- Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo.
- El resultado de esta fase es el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.
- Es conveniente distinguir entre
  - Diseño de alto nivel o **arquitectónico**: define la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. (ADD)
  - Diseño detallado: define los algoritmos empleados y la organización del código para comenzar la implementación. (DDD)



# Creación de Software. Ciclo de vida clásico.



## Fases del Proceso de Desarrollo del Software



# Creación de Software. Ciclo de vida clásico.



## Programación.

- Es la fase en donde se **implementa el código fuente**, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las **bibliotecas y componentes reutilizables dentro del mismo proyecto** para hacer que la programación sea un proceso mucho más rápido.

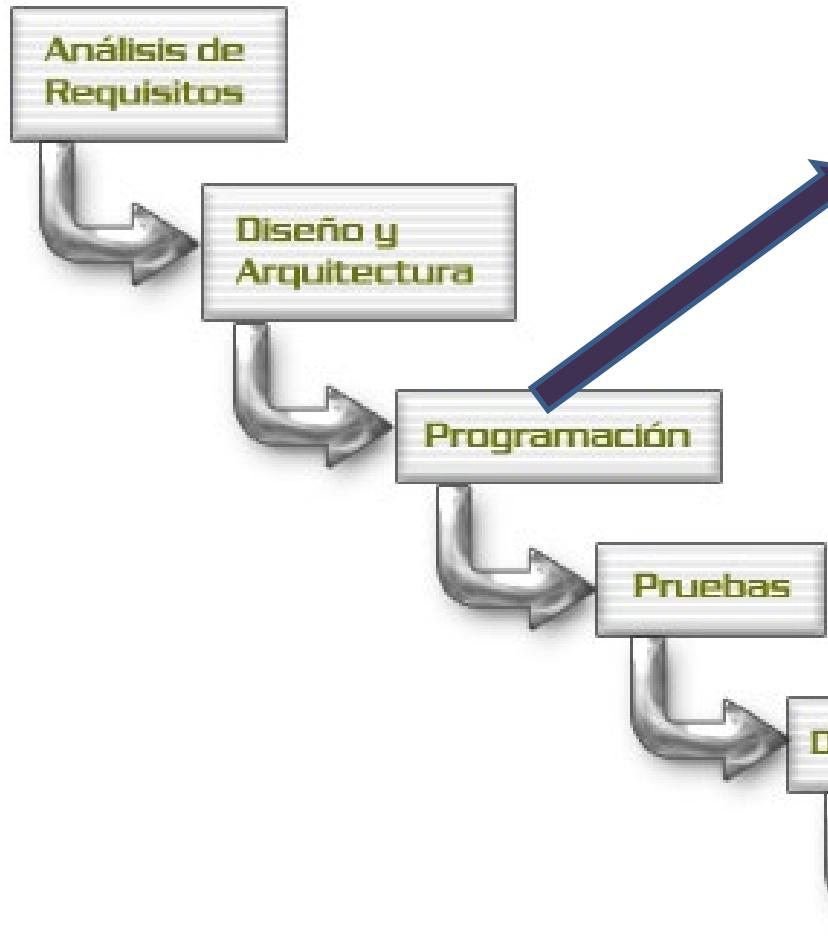
## Pruebas

- La fase de pruebas es **crítica** en un sistema, dada la dificultad de construir sistemas software sin fallos. Las pruebas pueden ser llevadas a cabo por
  - Los propios equipos de desarrollo
  - Equipos específicos de prueba
  - Usuarios finales

# Creación de Software. Ciclo de vida clásico.



## Fases del Proceso de Desarrollo del Software



**Herramientas**  
Compiladores,  
Editores, SDK,...

**Lenguajes y paradigmas**  
C, C++, Java, C#, PHP,  
HTML,...  
Programación  
estructurada, POO

# Creación de Software. Ciclo de vida clásico.



## Documentación y mantenimiento

- El proceso de documentación es una tarea transversal a todos los pasos, aunque se culmina en este punto.
- Una de las etapas mas críticas, (se destina un elevado porcentaje de los recursos), es el mantenimiento del Software, ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.
- Implica tareas, durante toda la vida útil del producto, de
  - Monitorización.
  - Evaluación.
  - Reparación.
  - Mejora del sistema.





Existen dos grupos principales:

- Metodologías tradicionales.
  - Los ciclos de desarrollo son poco flexibles.
  - No se permiten realizar cambios.
  - Organización del trabajo lineal.
- Metodologías ágiles.
  - Son flexibles y ágiles.
  - Altamente utilizadas en la actualidad.
  - Metodologías incrementales.
  - Creación de equipos de trabajo autónomos.

# Creación de Software. Metodologías tradicionales.



Destacan:

- **Waterfall (Cascada).** Construcción secuencial con revisión e producto en cada etapa.
- **Prototipado.** Basada en la construcción de un prototipo funcional para la recogida de feedback.
- **Espiral.** Combinación de las dos anteriores. Añada análisis de riesgo.
- **Incremental.** Construcción progresiva del proyecto.
- **Diseño rápido de aplicaciones (RAD).** Permite desarrollos de alta calidad en un periodo corto de tiempo, con un alto coste.



# Creación de Software. Metodologías ágiles.



Destacan:

- **Kanban.** Se dividen las tareas en porciones mínimas y se organizan en un tablero como tareas pendientes, en curso y finalizadas.
- **Scrum.** Metodología incremental. Se itera sobre bloques de tiempo cortos y fijos con resultados completos en cada loque.
- **Lean.** Pequeños equipos de trabajo muy capacitados y versátiles realizan las tareas en poco tiempo. El tiempo y el coste permanecen en un segundo plano.
- **Programación Extrema (XP).** Basada en relaciones interpersonales. Creación de buen ambiente de trabajo y feedback constante del cliente.



- En esta asignatura se trabajará el desarrollo web:
  - Implica lenguajes de programación y tecnologías diferentes del desarrollo de aplicaciones de escritorio o apps
  - Puede consultarse la evolución que han tenido estas tecnologías en este [enlace](#).

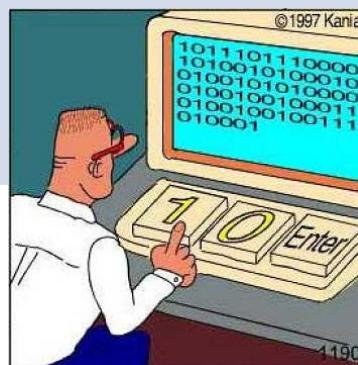


# Niveles de programación.



## Lenguaje máquina:

- Lenguaje más primitivo de programación.
- Basado en la interpretación binaria (0/1).



## Lenguajes de bajo nivel:

- Fáciles de interpretar.
- Pueden variar según la máquina en la que se esté programando.
- Ejemplo: Ensamblador.

Lenguaje de alto nivel

Lenguaje ensamblador

Lenguaje de máquina

## Lenguajes de alto nivel.

- Lenguajes más utilizados.
- Dividido en generaciones cronológicas.
- Ejemplos: C++, Python, C#, SQL, PHP,...

Usuario

Computadora

# Compiladores vs. intérpretes



- Su función es la convertir el código de software que se ha escrito a un formato ejecutable y legible por máquina.

## Compiladores

- Traduce código fuente a lenguaje intermedio
- Produce un archivo objeto

## Interpretes

- Traduce y ejecuta un programa

# Compiladores vs. intérpretes



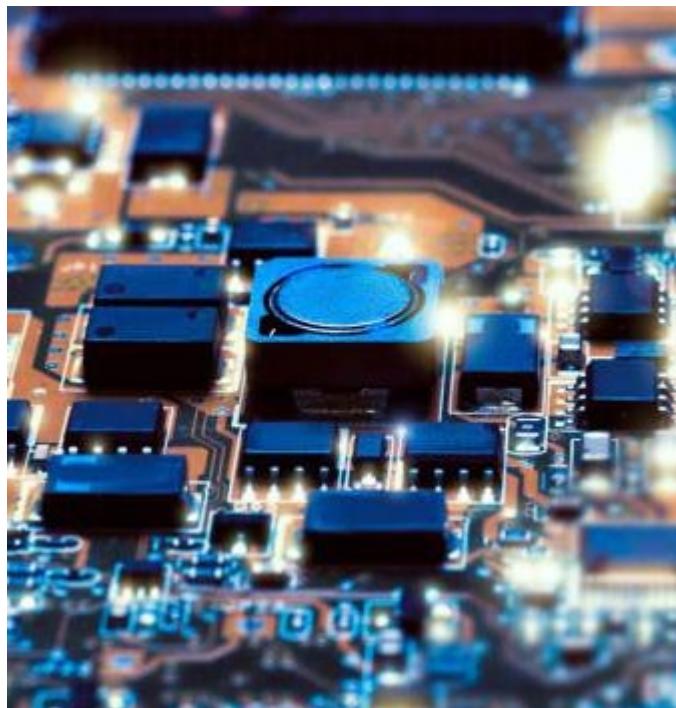
	Intérprete	Compilador
Momento en que se traduce el código fuente	Durante el tiempo de ejecución del software	Antes de ejecutar el software
Procedimiento de traducción	Línea por línea	Siempre todo el código
Presentación de errores de código	Después de cada línea	En conjunto, después de toda la compilación
Velocidad de traducción	Alta	Baja
Eficiencia de traducción	Baja	Alta
Coste de desarrollo	Bajo	Alto
Lenguajes típicos	PHP, Perl, Python, Ruby, BASIC	C, C++, Pascal

	Ventaja	Inconveniente
Intérprete	Proceso de desarrollo sencillo (sobre todo en términos de depuración)	Proceso de traducción poco eficiente y velocidad de ejecución lenta
Compilador	Proporciona al procesador el código máquina completo y listo para ejecutar	Cualquier modificación del código (resolución de errores, desarrollo del software, etc.) requiere volverlo a traducir

\*Fuente.

# Fundamentos de Ingeniería Informática

---



Universidad  
Francisco de Vitoria  
**UFV** Madrid

## TEMA 5

### **EL HARDWARE.** **Sistemas** **informáticos.**

# Objetivos



- Clasificar distintos **tipos de equipos informáticos** en función de su propósito y potencia de cómputo.
- Describir la **estructura básica** de un ordenador y las diferencias entre la Arquitectura Von Neumann y la Harvard.
- Explicar las **funciones del procesador**, los componentes principales que lo forman y los distintos niveles de paralelismo que se aplican para mejorar el rendimiento.
- Exponer las implicaciones que tiene que un procesador sea de **32 o 64 bits**.
- Enumerar qué tipos distintos de **memoria** aparecen comúnmente en un ordenador.
- Explicar la función del **adaptador de E/S** y argumentar la necesidad de usarlo.
- Enumerar y describir brevemente los diferentes mecanismos de **comunicación** entre procesador y dispositivos periféricos.

# —

# ¿Qué es un COMPUTADOR?



## Definición de Ordenador de la R.A.E

Máquina electrónica dotada de una **memoria** de gran capacidad y de métodos de **tratamiento de la información**, capaz de resolver problemas aritméticos y lógicos gracias a la utilización automática de **programas** registrados en ella

## Nuestra definición

...

# —

# ¿Qué es un COMPUTADOR?



# Tipos de sistemas por propósito.



- Propósito general.
- Propósito específico.



# Tipos de sistemas por potencia.



- Supercomputadores.
- Mainframes (servidores de gama alta).
- Servidores de gama media y básica.
- Computadores personales (de sobremesa y portátiles).
- Computadores móviles.

**Servidor:** computador conectado a una red de transmisión de datos que da servicio compartido a múltiples usuarios. Pueden ser de diferentes tipos: servidor de aplicaciones, servidor de archivos, servidor de red, ...

# Tipos de sistemas por potencia.



- Supercomputadores
  - Cálculo intensivo de tipo científico y técnico (High Performance Computing, HPC)
  - Pueden llegar a tener miles de procesadores.
  - Memoria principal > 1 TB
  - Disco > 100 TB
  - Lista en [www.top500.org](http://www.top500.org)
  - Aplicaciones:
    - Estudio del genoma
    - Simulaciones climatológicas
    - Nuevos medicamentos

Supercomputador [Frontier](#)



# Tipos de sistemas por potencia.



- Supercomputadores: MARENOSTRUM

Supercomputador más potente de España, situado en el Centro de Supercomputación de Barcelona.

Estuvo en el puesto **5 del mundo en 2005** y en 2011 ocupaba el 170, actualmente está en el puesto 98

Consta de 153,216 cores y dispone de un procesador Xeon Platinum 8160 24C 2.1GHz.

La memoria principal es de **41.664 GB** (40 TB) y la capacidad de almacenamiento en discos es de **240 TB**.



# Tipos de sistemas por potencia.



- **Mainframes. Servidores de gama alta.**
  - Acceso a grandes bases de datos desde terminales (entidades bancarias, reserva de billetes,...)
  - Varios cientos de procesadores.
  - De cientos a miles de usuarios.
- **Servidores de gama media y básica.**
  - Similares a los mainframes, pero están ubicados en departamentos o empresas pequeñas
  - Se controlan remotamente.
  - Suelen disponerse en racks.
  - Hasta cientos de usuarios.



# Tipos de sistemas por potencia.



- **Computadores personales.**

- Monousuario
- Los de gama alta se denominan **estaciones de trabajo (workstations)**: mayor capacidad de cálculo y gráfica
- Incluye los PC de sobremesa (fijos) y los portátiles
- Existe muchos modelos y fabricantes

- **Computadores móviles.**

Pequeño, ligero, alimentación por acumuladores

Memoria RAM

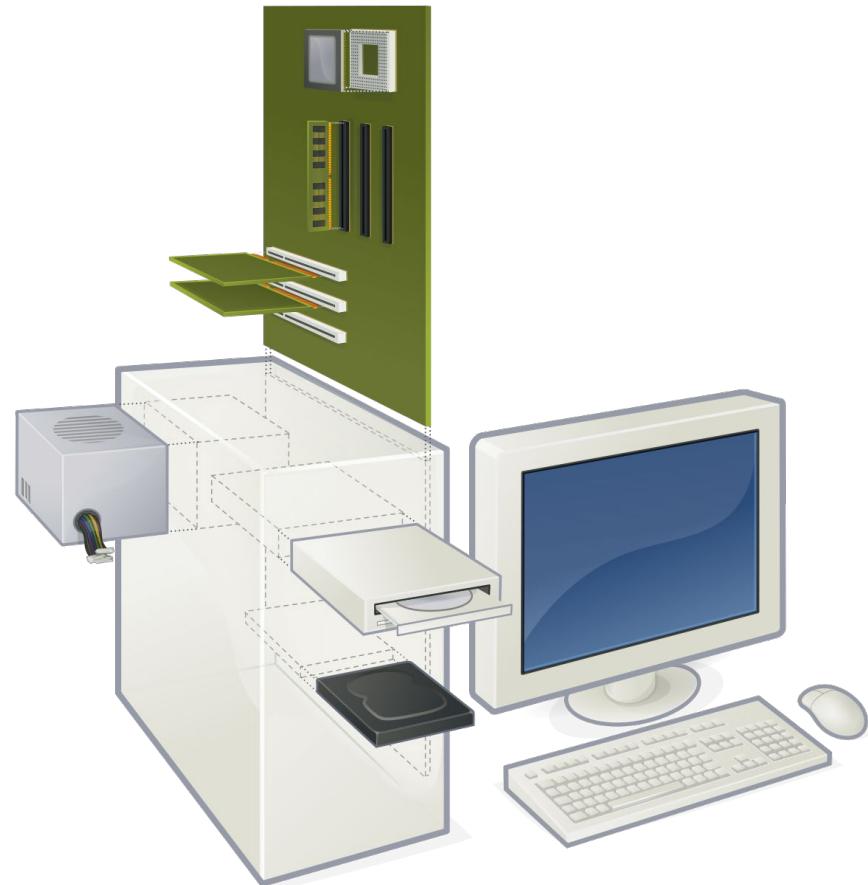
Memoria Flash



# Estructura básica de un computador.



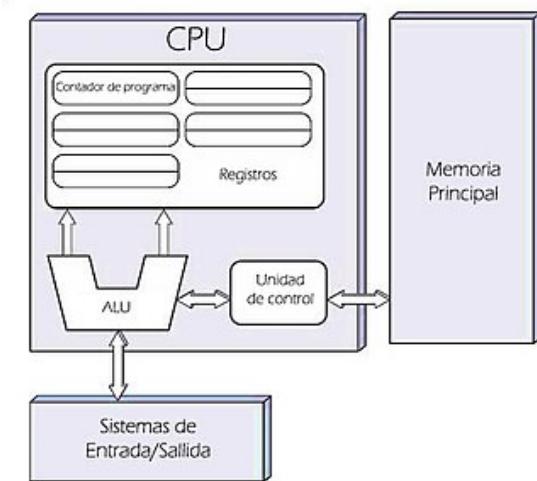
- En todo computador de propósito general pueden distinguirse 4 elementos fundamentales:
  - Procesador (CPU).
  - Sistema de almacenamiento.
  - Sistema de entrada/salida.
  - Sistemas de interconexión.



# Estructura básica de un computador. Arquitectura Von Newmann.



- A.W. Burks, H.H. Goldstine, J. Von Neumann proponen en 1946 una arquitectura para una máquina llamada IAS.
- Esta arquitectura está formada por 3 módulos:
  - **CPU**: manipulación de la información mediante la ejecución de instrucciones
  - **Memoria**: almacenamiento de datos e instrucciones
  - **Unidad de Entrada/Salida**: para intercambio de información con el exterior
- Una única memoria para almacenamiento de datos e instrucciones (secuencias de bits), con una organización lineal por palabras
  - **Palabra**: cantidad de información que se lee o escribe en cada acceso a memoria
  - El acceso a cada palabra se realiza mediante una *dirección*
- Secuenciamiento implícito de instrucciones
  - Las instrucciones se ejecutan de una en una, siguiendo el orden en que se encuentran almacenadas en la memoria



# Estructura básica de un computador. Arquitectura Von Newmann.



- **Instrucciones del lenguaje máquina:** ejecutan operaciones sencillas sobre datos elementales
  - instrucciones de transferencia de datos entre memoria y CPU
  - instrucciones aritméticas y lógicas
  - instrucciones de desplazamiento y rotación de bits
  - instrucciones de ruptura de secuencia o salto
  - etc.

# Estructura básica de un computador.

## Arquitectura Von Newmann.



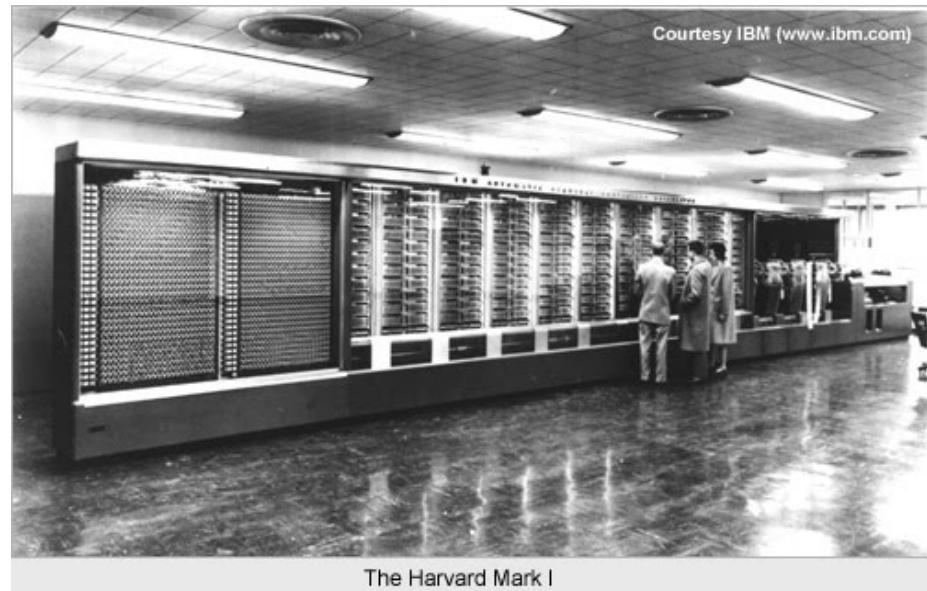
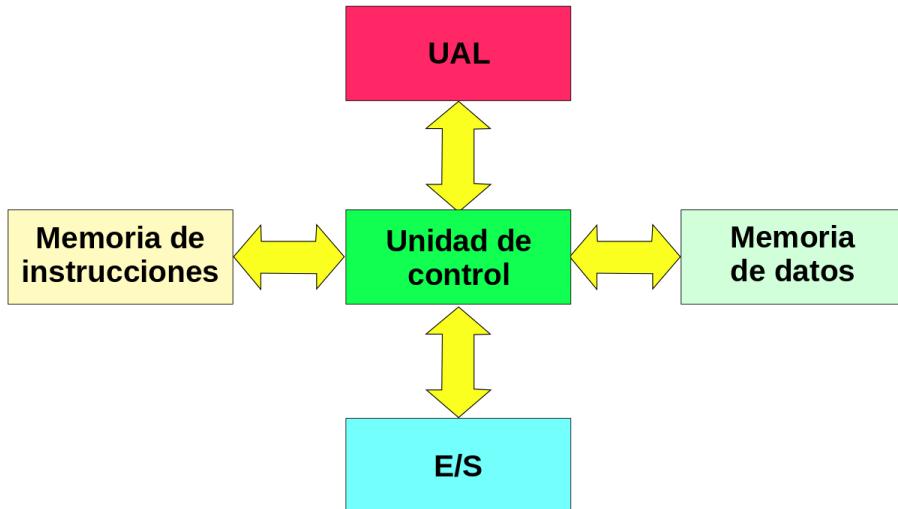
- Para ejecutar un programa hay que ejecutar las instrucciones que lo forman, una a una.
- La ejecución de cada instrucción implica comúnmente las tareas reflejadas en la figura de la derecha.
- Estas tareas se suelen realizar en etapas, donde cada etapa, a su vez, se realiza en un ciclo de reloj.
- El número de etapas y la duración de cada ciclo de reloj depende del procesador.



# Estructura básica de un computador. Arquitectura Harvard.



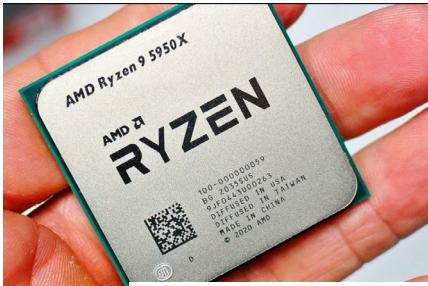
- Arquitectura de computadora con pistas de almacenamiento en la que las memorias están separadas para datos e instrucciones.
- La primera computadora con esta arquitectura fue la Harvard Mark 1, que almacenaba en cintas perforadas las instrucciones y los datos en interruptores electrónicos.



# Procesador



- Componente encargado de la ejecución de instrucciones.
- ¿Qué procesadores conoces?



## Intel Core i5-1035G4

### Marca del procesador

La mayoría de los fabricantes de portátiles se decantan por instalar procesadores Intel en sus equipos; aunque en algunas marcas y familias de portátiles también es posible encontrar procesadores marca AMD y Qualcomm, e, incluso, recientemente Apple ha anunciado su intención de volver a montar sus propios procesadores.

### Familia

Intel cuenta con diferentes familias de procesadores, pero para un uso doméstico las más habituales son Atom, Celeron, Pentium y Core.

### Gama

Un número más alto indica mayor potencia y precio.

### Generación

Cuento más alto es el número, más reciente es el procesador. En el ejemplo tenemos un procesador Intel Core i5 de 10<sup>th</sup> generación.

### SKU

Dentro de la misma generación, cuanto más alto es el número, más reciente es el procesador.

### Sufijo

Indica capacidades concretas del procesador. Las más frecuentes en los portátiles son:

- **G1 a G7:** indica el nivel de los gráficos. A mayor número, mejor rendimiento .
- **HQ:** la H indica que es un procesador de alto rendimiento optimizado para dispositivos portátiles. La Q indica que son cuatro núcleos.
- **U:** indica que es de bajo consumo de energía ("Ultra" Low power).
- **Y:** indica que son ordenadores de consumo de energía extremadamente bajo.
- **K:** indica que se puede modificar la frecuencia de trabajo del procesador (GHz) para un mayor rendimiento.

# Procesador. Componentes básicos.



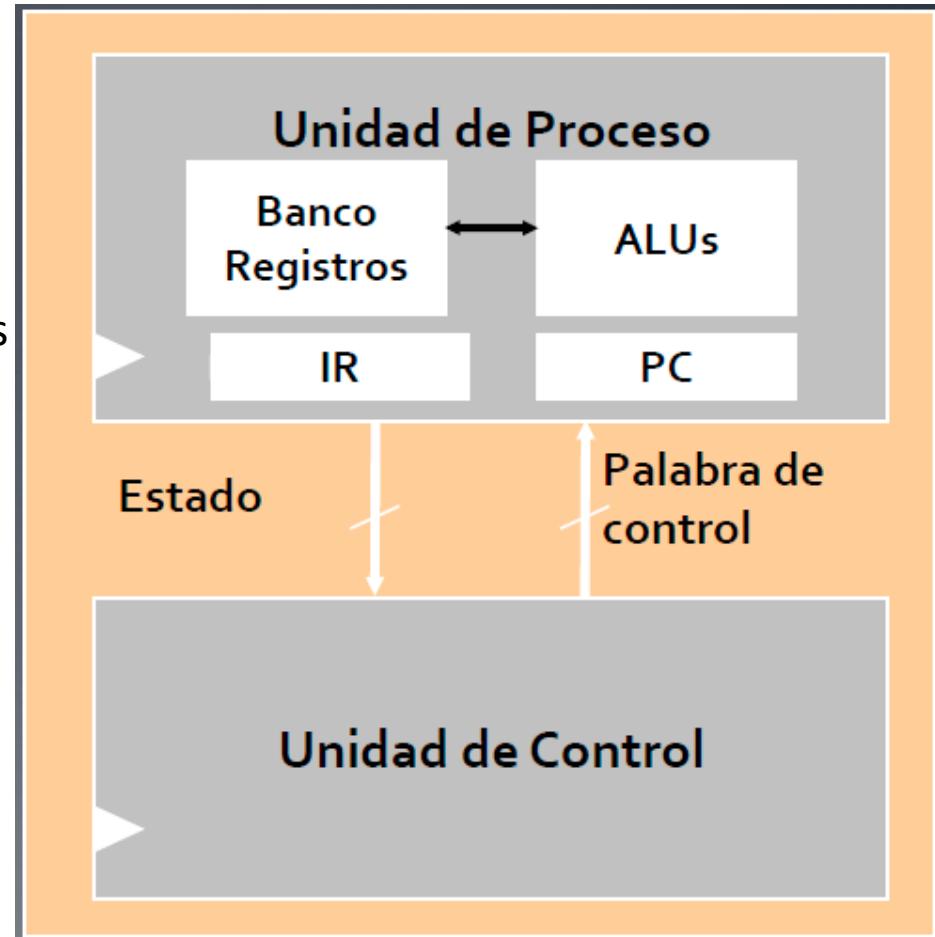
*Se compone principalmente de:*

- **Unidad de proceso** (UP): se encarga de realizar las acciones básicas que permiten completar la ejecución de una instrucción máquina.
  - búsqueda de la siguiente instrucción.
  - Descodificación.
  - lectura de operandos.
  - Ejecución.
  - escritura del resultado.
- **Unidad de control** (UC): se encarga de asegurar que la UP realiza las acciones básicas en la secuencia correcta, generando las señales de control que gobiernan su funcionamiento.

# Procesador. Componentes básicos.



- Los principales elementos de la **Unidad de Proceso** son:
  - **Unidades de ejecución:** unidades aritméticas para enteros, unidades aritméticas para coma flotante, etc.
  - **Registros de propósito general:** conjunto de registros que pueden utilizarse para almacenar datos de manera temporal.
  - **Registros de control y estado:** Registro de instrucciones (IR) y Contador de Programa (PC) entre otros.



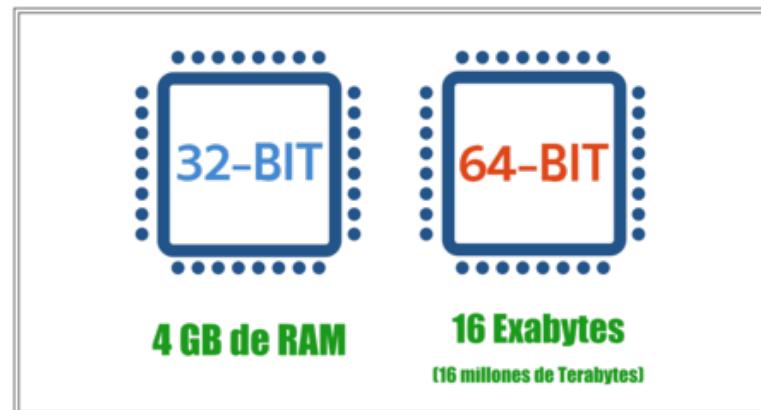
# Procesador. Número bits.



¿Qué significa que un procesador sea de 32 (o 64) bits?

- Es el tamaño, en número de bits, con los que cuentan los registros internos de procesador. También se refiere a la capacidad de procesamiento de la Unidad Aritmética Lógica (ALU), que es la responsable de todos los cálculos matemáticos de la computadora.

En la actualidad casi todos los procesadores instalados son de 64 bits.



# Procesador. Número bits.

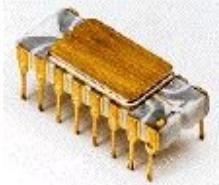


- ¿Qué implicaciones tiene?
  - Los datos enteros que se pueden operar están limitados a 32 (o 64) bits. Si se quiere operar con datos mayores, hay que hacer la operación por partes.
    - Ejemplo: ADD \$1,\$2,\$3 es una instrucción del procesador MIPS R2000, que es de 32 bits. Los registros almacenan números de 32 bits, y la suma se realiza sobre esos 32 bits.
  - Las direcciones de memoria están limitadas a 32 (o 64 bits) con lo que también se limita el espacio de direcciones y, por tanto, la cantidad total de memoria RAM que puede tener el sistema
    - 32 bits =>  $2^{32}$  direcciones de memoria distintas => espacio de direcciones de 4GB (memoria máxima que puede tener el sistema)

# Procesador. Número bits.



- Evolución del número de bits de los procesadores de intel



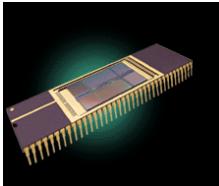
1971: **4004**  
4 bits



1974: **8080**  
8 bits



1978: **8086**  
16 bits



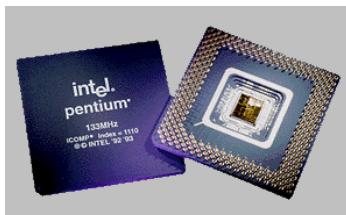
1982: **80286**  
16 bits



1985: **80386**  
32 bits (IA32)



1989: **80486**,  
32 bits



1993: **Pentium**,  
32 bits

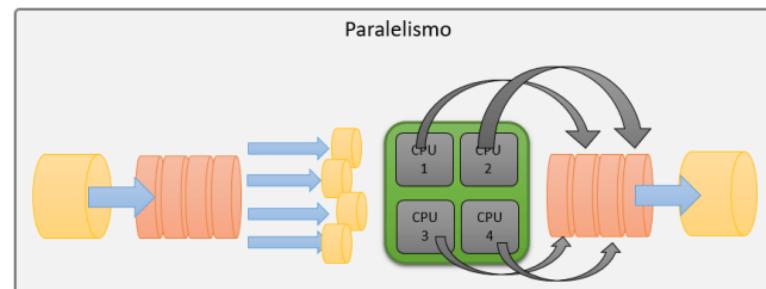


2000: **Pentium 4**, 32 bits  
2004: **Pentium 4**, 64 bits

# Procesador. Paralelismo.



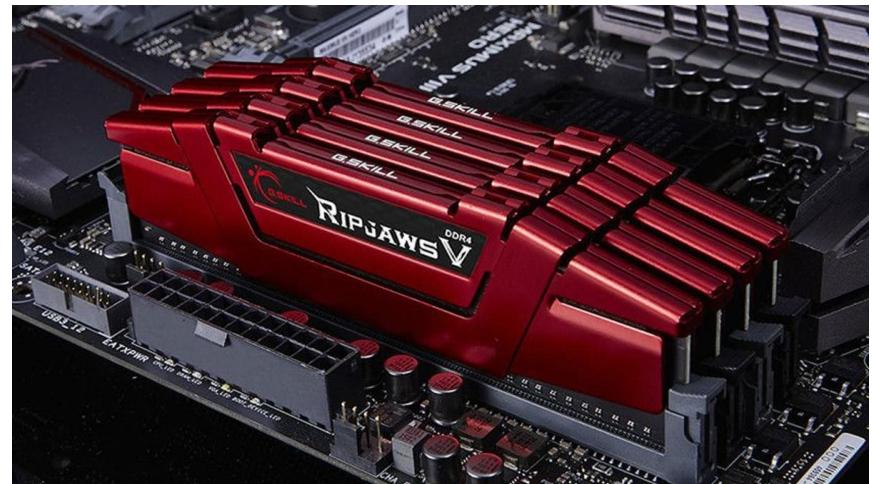
- Un principio básico aplicado a la mejora del rendimiento de los procesadores es HACER TODO LO POSIBLE EN PARALELO.
- Aparece paralelismo en varios niveles
  - A nivel de instrucción.
    - Procesadores **superescalares** y VLIW: pueden ejecutar varias instrucciones de un mismo proceso en paralelo.
    - Es transparente para el SO y al programa.
  - A nivel de proceso:
    - **Multiprocesador**: sistema con al menos dos procesadores.
    - **Microprocesador multinúcleo**: microprocesador que consta de varios procesadores (núcleos) en el mismo circuito integrado.
    - No es transparente para el SO y para el programa.



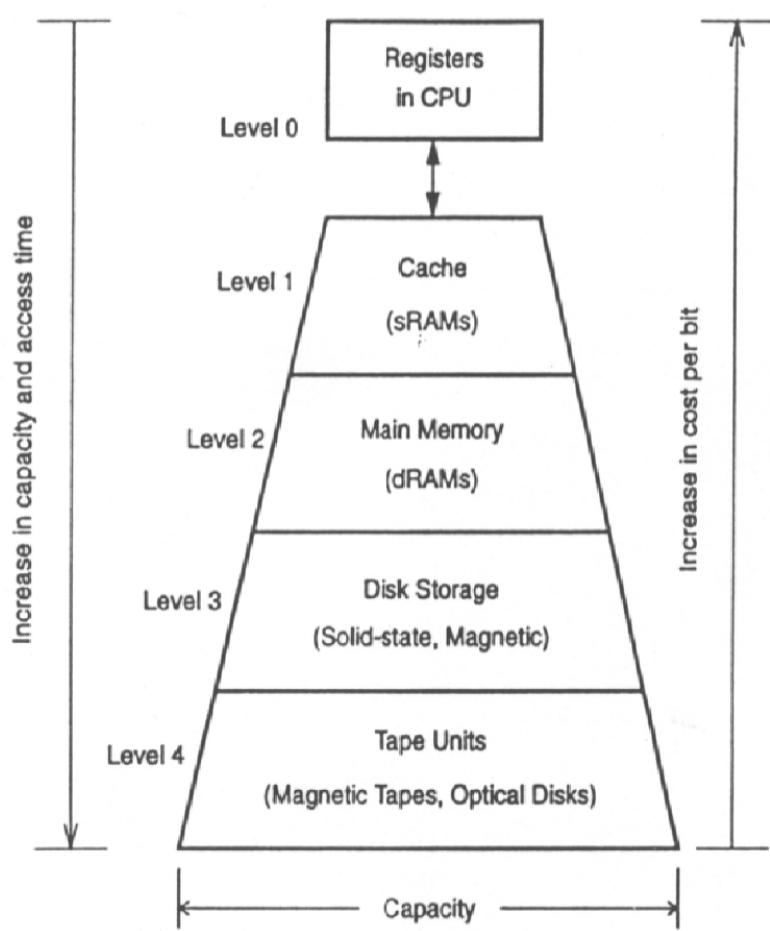
# Memoria.



- Existen diferentes tipos de memoria dentro de un sistema informático, que se diferencian en características como:
  - **Capacidad** (tamaño medido en bytes).
  - Prestaciones (**Tiempo de acceso**: tiempo que tarda en realizarse una operación de escritura o lectura).
  - Soporte (magnético, óptico, semiconductor).
  - Modo de acceso (secuencial, directo, aleatorio).
  - Otras:
    - Volátil / No volátil
    - Lectura / Lectura-escritura



# Memoria.



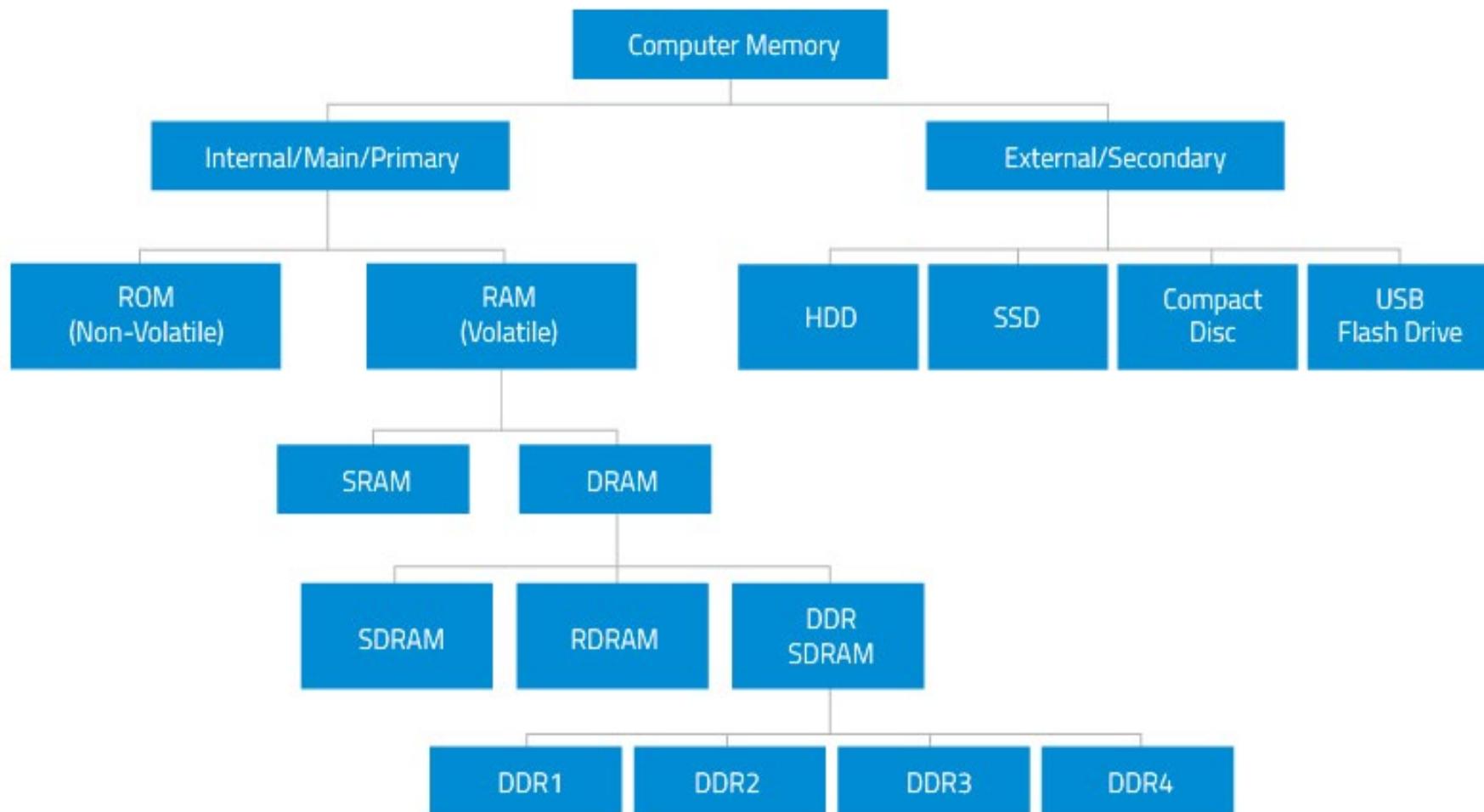
Gestionados por el compilador (asignación de registros) y el hardware (procesador)

Gestionada por el hardware (unidad de gestión de memoria o MMU)

Gestionada por el hardware (MMU) y por el SO

Gestionados por el SO con una intervención limitada del usuario.

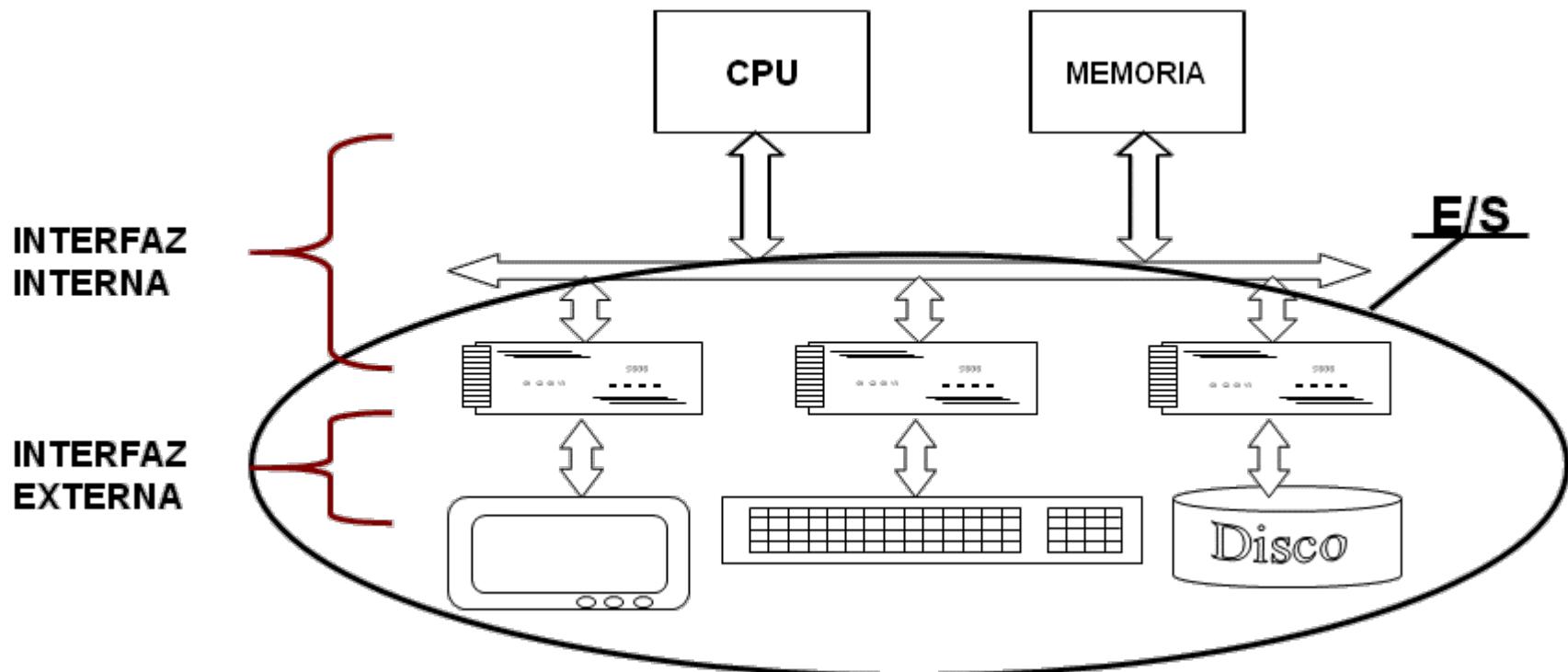
# Memoria.



# Periféricos de E/S



- **Periféricos:** Dispositivos que permiten conectar al ordenador con el exterior. Se conectan a la Unidad Central de Proceso a través de las unidades de Entrada/Salida.



# Periféricos de E/S. Clasificación.



- **Función:**
  - **Comunicación:**
    - Comunicación Hombre-Máquina: Terminales.
    - Comunicación Impresa: Impresoras, lectores ópticos, digitalizadores, etc.
    - Comunicación Máquina-Máquina: Módems y adaptadores de red local.
    - Comunicación con un sistema físico: Periféricos de control.
  - **Almacenamiento:**
    - Discos magnéticos.
    - Discos ópticos.
    - Memorias semiconductoras: flash, SD...
- **Sentido de la comunicación:**
  - De entrada: teclado, escáner, lector de caracteres ópticos o magnéticos, etc.
  - De salida: impresora, monitor.
  - De entrada / salida: modem, tarjeta de sonido, tarjeta de red.
- **Distancia al ordenador:**
  - Locales.
  - Remotos.

# Periféricos de E/S. Comunicación con la CPU.



- Existen 3 técnicas de E/S principales:
  - **E/S programada.**
    - La operación de E/S se produce bajo el control directo y continuo del programa que la solicita.
  - **E/S mediante interrupciones.**
    - El programa genera una orden de E/S y después continua ejecutándose hasta que el dispositivo interrumpe para indicar que la operación ha concluido.
  - **Acceso directo a memoria (DMA).**
    - Un procesador de E/S toma el control de la operación para transferir un gran bloque de datos entre el dispositivo y la memoria principal, sin intervención de la CPU.

# Fundamentos de Ingeniería Informática

---



## TEMA 6

## SISTEMAS OPERATIVOS



# Objetivos



## Objetivo general

- Comprender las funciones básicas de los sistemas operativos.

## Objetivos específicos

- Definir sistema operativo y detallar sus funciones básicas
- Diferenciar monoprogramación, multiprogramación y multiprocесamiento.
- Diferenciar proceso de programa y describir los estados en que puede hallarse un proceso.
- Explicar en qué consiste la planificación de procesos.
- Explicar cuáles son los objetivos principales de la memoria virtual y la diferencia entre memoria paginada y segmentada
- Explicar el objetivo de la gestión de la entrada/salida
- Describir qué es un sistema de ficheros.

# Contenidos y bibliografía



## CONTENIDOS

- Conceptos clave.
- Gestión del procesador.
- Gestión de la memoria.
- Gestión de entrada/salida.
- Gestión de almacenamiento masivo.

## BIBLIOGRAFÍA

- PRIETGO, A. **Introducción a la informática**, 4<sup>a</sup> Edición. Editorial McGraw-Hill. Madrid. 2006. ISBN 9788448146245
- STALLINGS, W. **Sistemas operativos**, 5<sup>a</sup> Edición. Editorial Pearson Education. Madrid. 2005.

# Conceptos clave

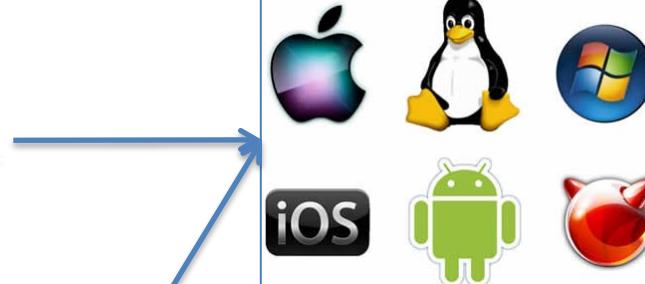


- Un sistema operativo es un programa (entra dentro de la categoría de **Software de Sistema** vista en el tema 3) que tiene por objeto **facilitar el uso eficiente del computador**, tanto por parte de los **usuarios finales** que utilizan el sistema como por parte de las **aplicaciones** que se ejecutan sobre el mismo.
  - **Facilita el uso:** Genera una especie de máquina virtual/ capa de abstracción de hardware, que es un ordenador simplificado en el que el sistema operativo carga con el trabajo de acercamiento al hardware y lo hace transparente para el usuario.
  - **Uso eficiente del computador:** coordina las diferentes velocidades y características de funcionamiento de los dispositivos del equipo y reparte su uso de forma optimizada entre los distintos usuarios y tareas.

# Conceptos clave



- Gestión y acceso a los recursos hardware

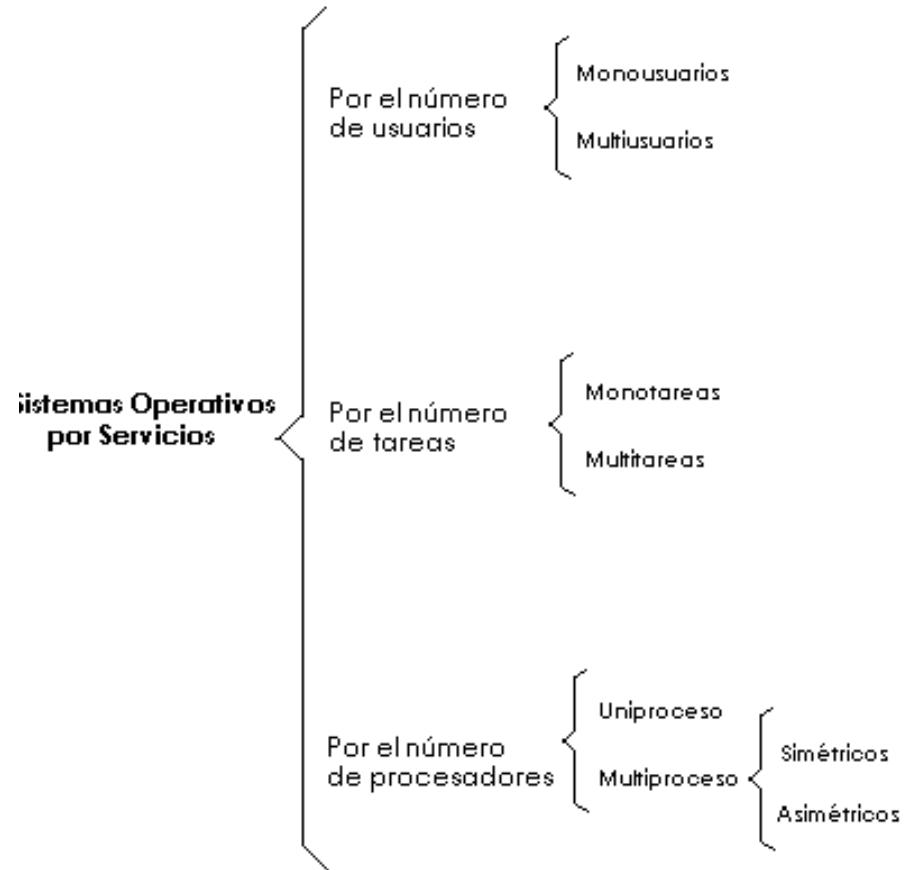


Procesador  
Memoria  
Periféricos (Entrada/salida)  
Discos  
Pantallas  
Teclado, ratón  
Impresoras

# Conceptos clave



- Por lo tanto el sistema operativo:
  - Es un programa o conjunto de programas (kernel + programas de utilidades adicionales).
  - Actúa de intermediario entre el hardware y los programas de aplicación y usuarios.
- Clasificación:



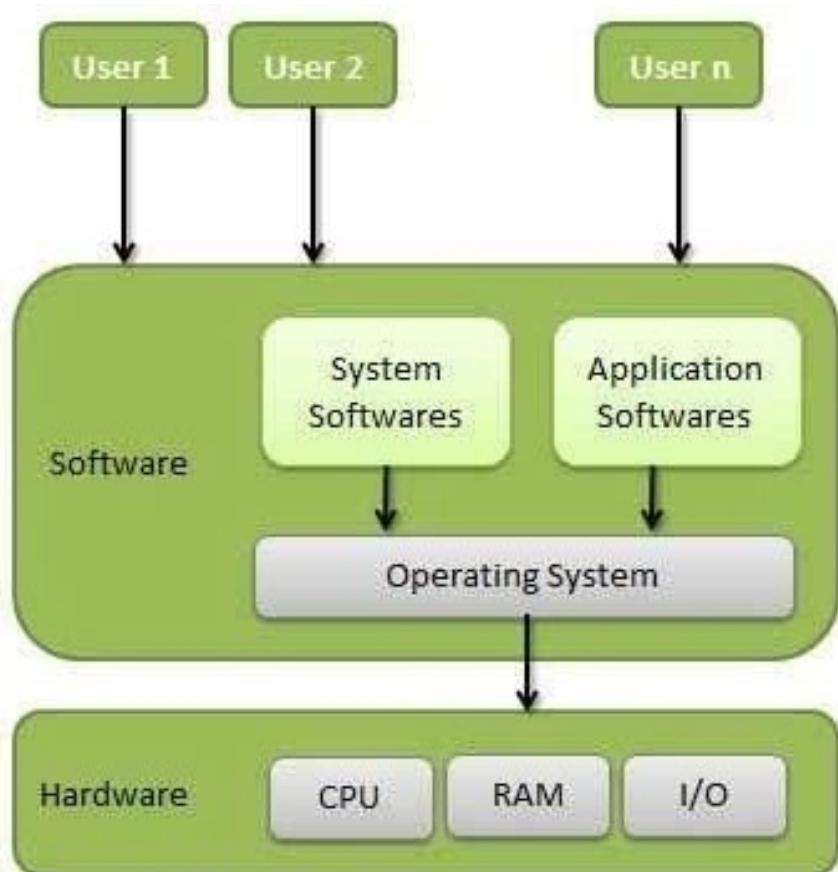
# Conceptos clave



## Funciones de los Sistemas Operativos

- Asignación de tiempo de CPU:
  - Planificar las diversas actividades.
- Control de recursos:
  - Asignar recursos de forma racional. Por ejemplo: división de la memoria del ordenador entre los programas.
- Control de entrada/salida:
  - Gestionar los datos que se envían desde y hacia los periféricos.
- Control de los errores y protección:
  - Informar de las situaciones anómalas
- Interfaz con el usuario:
  - Proporcionar acceso a los recursos (de comandos, gráfica,...)
- Control estadístico y del rendimiento:
  - Calcular datos de uso de CPU y memoria (clave en el uso compartido de sistemas).

# Conceptos clave



- Un sistema operativo realiza básicamente las siguientes operaciones:
  - Gestión del procesador.
  - Gestión de la memoria.
  - Gestión de las operaciones de entrada/salida sobre los periféricos.
  - Gestión del almacenamiento en disco.

# Gestión del procesador



- Implica básicamente la **asignación de tiempo de procesador (CPU)** a los programas, de forma que se optimice el uso del procesador.
- Un programa en ejecución se denomina **proceso**, luego el SO se encarga de planificar procesos.
- Existen tres modelos o paradigmas para la planificación de procesos:
  - **Monoprogramación**: no se inicia un proceso hasta que no finaliza otro
  - **Multiprogramación**: se pueden ejecutar varios programas a la vez, pero sobre la misma CPU
  - **Multiprocesamiento**: se pueden ejecutar varios procesos a la vez en diferentes CPUs

# Gestión del procesador

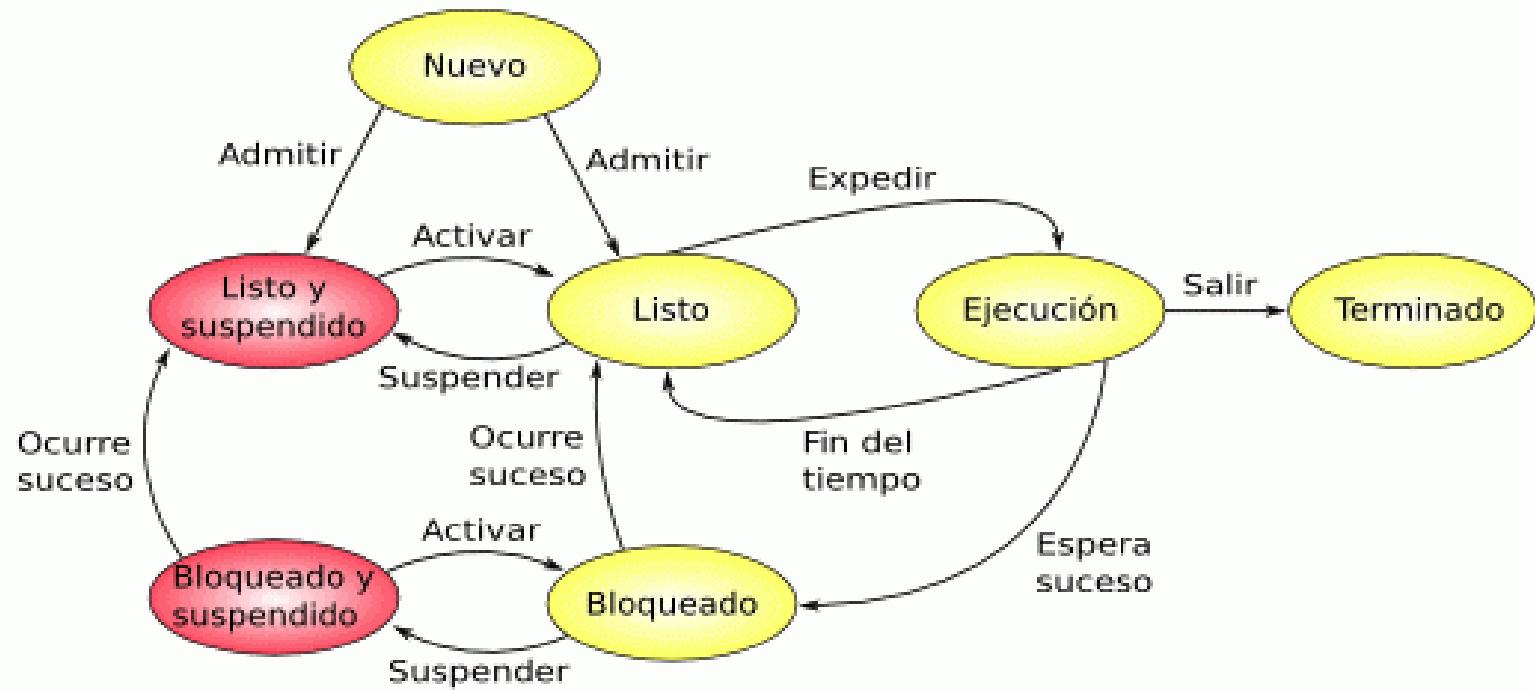


- Evidentemente, un sistema **monoprogramado** es incómodo para el usuario (imagina un ordenador en el que no puedes abrir dos programas al mismo tiempo) y poco eficiente.
- Los **procesos activos** (que están en ejecución) no están todo el tiempo ejecutando instrucciones; en muchos momentos se detienen esperando, por ejemplo, que llegue un dato de un periférico (como un teclado). En esos momentos, el procesador no está siendo utilizado, por lo que se pierde eficiencia en el uso del mismo.
- Un sistema **multiprogramado** resuelve ambos problemas: permite tener varios programas “abiertos a la vez”, y aprovecha los momentos en que unos procesos están esperando un evento para que otros utilicen el procesador.

# Gestión del procesador



- La **planificación de procesos** consiste, por tanto, en asignar tiempo de CPU a cada proceso activo.
- La siguiente figura muestra un ejemplo de cómo se produce la planificación de procesos y del **estado** en que puede encontrarse cada uno.



# Gestión del procesador



## Planificación de procesos, explicación del diagrama de estados:

- **Nuevo**: un proceso nuevo es un programa que no ha iniciado su ejecución. Es el estado inicial de cada proceso.
- Cuando un usuario “abre un programa” (lo lanza a ejecución) el proceso pasa de Nuevo a **Listo** (está preparado para ejecutarse y en memoria principal) o **Listo y suspendido** (está preparado para ejecutarse pero permanece de momento en el disco). El pasar a uno u otro estado depende, entre otras cosas, de la carga del procesador y los criterios del planificador. Así como el paso de Listo a Listo y suspendido y viceversa.
- El estado **Ejecución** se corresponde con un proceso que se está ejecutando en la CPU en ese instante (en cada procesador, sólo puede haber un proceso en ejecución simultáneamente).

# Gestión del procesador



**Planificación de procesos**, explicación del diagrama de estados (cont):

- Un proceso deja de estar **En ejecución** en dos situaciones: o bien se acaba el tiempo que le había asignado el procesador, pasando al estado **Listo** nuevamente, donde esperará otro turno de ejecución, o bien se para esperando algún evento (normalmente, la comunicación con un periférico de entrada/salida), pasando al estado **Bloqueado**. También puede pasar al estado **Terminado** si se finaliza ese proceso.
- Del estado **Bloqueado** puede pasar al estado **Listo**, cuando se produce el evento que esperaba, o al **Bloqueado suspendido**, si se devuelve a disco por decisión de planificador hasta que se produzca dicho evento.

**Planificación de procesos. Multiprocesamiento.**

- Un sistema operativo con **multiprocesamiento** es aquel que tiene varios procesadores (o núcleos de ejecución). El sistema operativo planifica para cada procesador o núcleo siguiendo un algoritmo como el explicado para un solo procesador.

# Gestión de la Memoria



- La memoria principal de la mayoría de los ordenadores es mucho mas pequeña de lo que sería necesario para manejar todos los programas y datos.
- El **módulo de gestión de memoria** es el encargado de asignar ciertas porciones de la memoria principal a los diferentes programas, mientras el resto de programas y datos se mantienen en los dispositivos de almacenamiento masivo (discos).
- La forma mas común de gestión de memoria es crear una **memoria virtual** utilizando los dispositivos de almacenamiento masivo. La memoria virtual permite dos cosas:
  - Contar con una memoria para ejecución de procesos superior a la memoria real del sistema.
  - Proteger el espacio de memoria de unos procesos frente a otros.

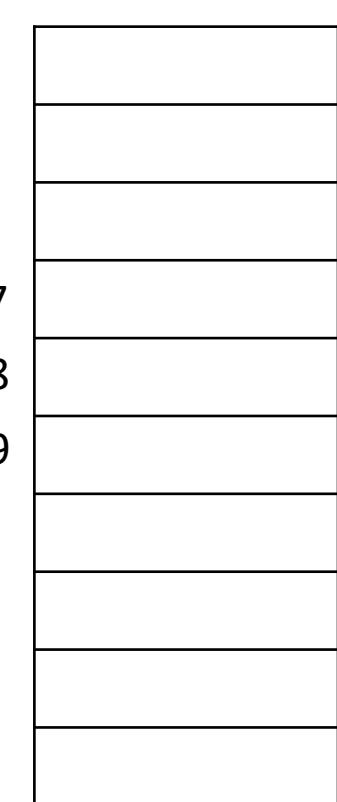
# Gestión de la Memoria



## Memoria virtual

- Amplía la memoria para ejecución de procesos sin añadir más memoria física.
  - Esto se consigue utilizando el disco duro (o sistema de almacenamiento masivo o secundario) como una extensión de la memoria RAM.
  - La memoria RAM no es mas que una colección de celdas o casillas cada una de las cuales puede almacenar un byte de información.
  - Cada celda está identificada por un número que se denomina **dirección**.

Direcciones físicas



RAM

# Gestión de la Memoria



## Memoria virtual

- Amplía la memoria para ejecución de procesos sin añadir más memoria física (cont)
  - Cuando se utiliza memoria virtual, los procesos no usan direcciones físicas para acceder a la memoria, sino **direcciones virtuales**, que son traducidas en tiempo de ejecución a direcciones físicas
  - Las direcciones virtuales suelen ser mayores, dando un acceso a una cantidad de memoria superior a la real

Direcciones virtuales

↓  
127347  
650823  
499701



Direcciones físicas

0737
0738
0739

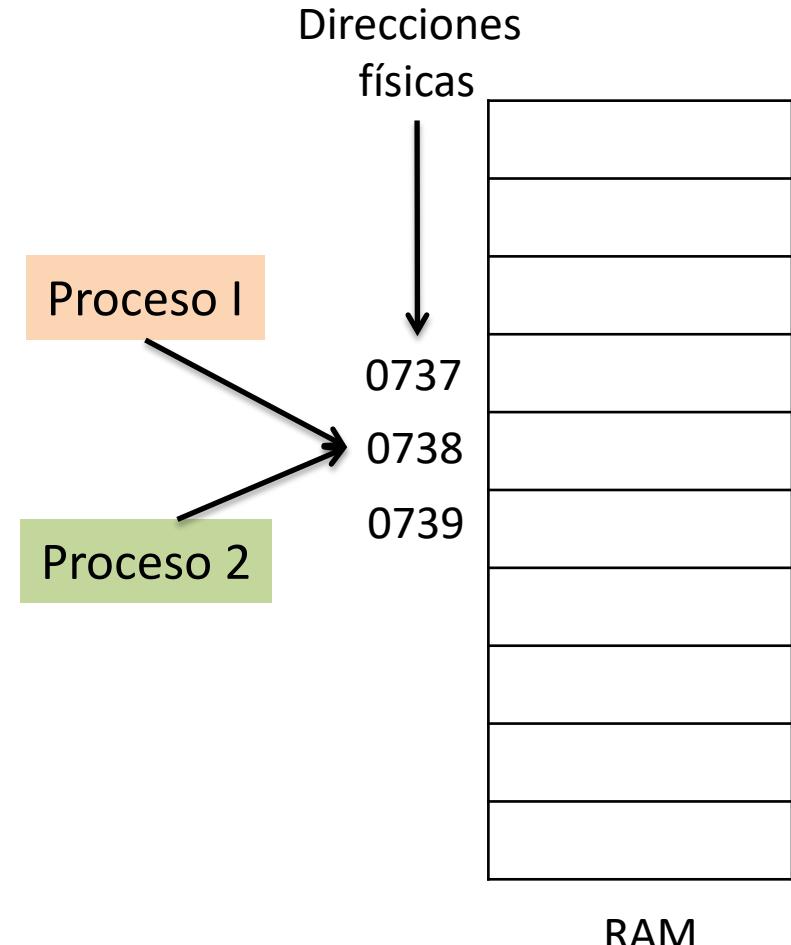
RAM

# Gestión de la Memoria



## Memoria virtual

- Protege el espacio de direcciones de un proceso frente a otro
  - Cuando se genera un programa, este utiliza la memoria sin ser consciente de que va a tener que compartirlo con otros; por lo tanto, en los **sistemas multiprogramación** podría haber **conflicto entre dos procesos activos** que podrían estar utilizando las mismas posiciones para guardar sus datos

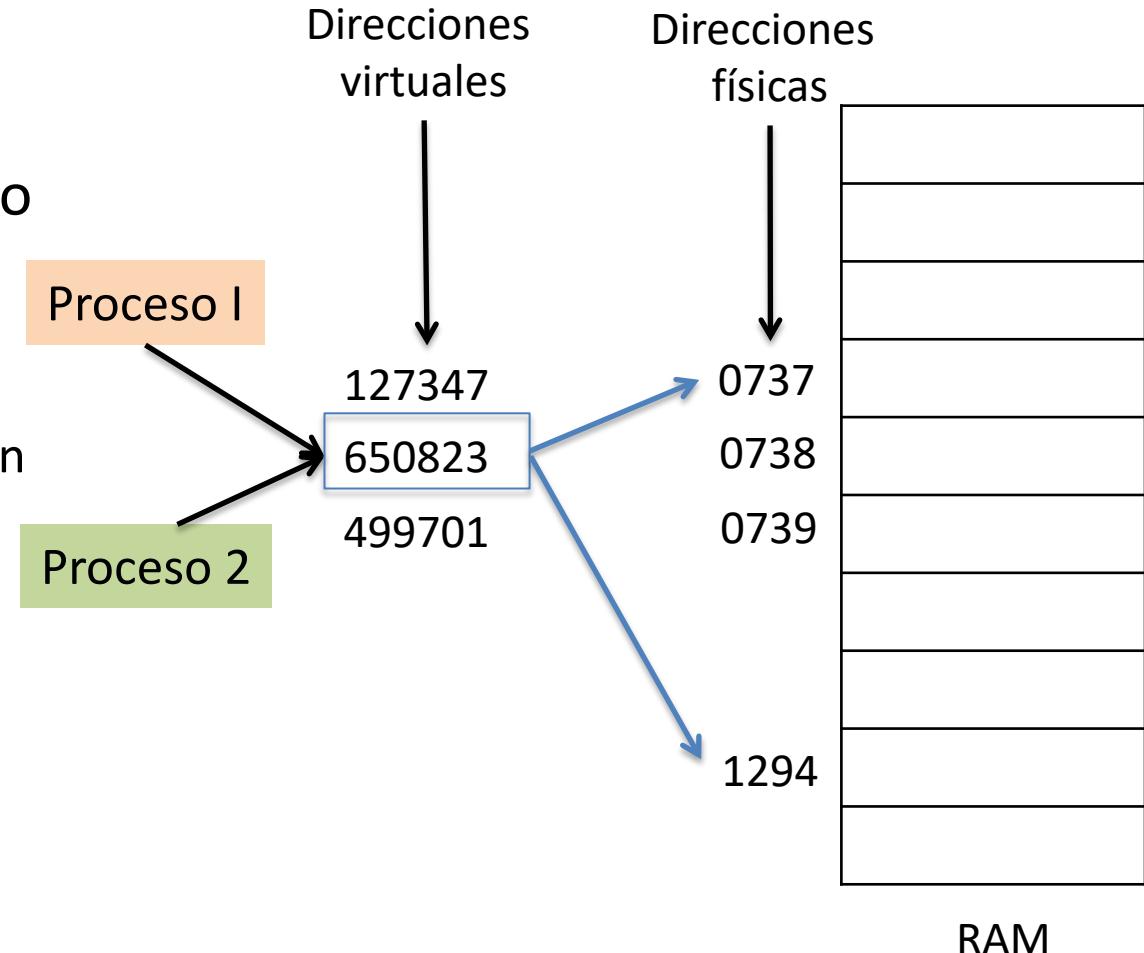


# Gestión de la Memoria



## Memoria virtual

- Protege el espacio de direcciones de un proceso frente a otro (cont)
  - Esto no ocurre con memoria virtual, porque aunque los procesos hagan referencia a la misma dirección, esta será traducida a direcciones físicas diferentes



# Gestión de la Memoria



## Memoria virtual

- La asignación de memoria a procesos se realiza dividiendo la memoria en bloques.
- Existen tres enfoques:
  - Por **páginas**: bloques de igual tamaño para todos los procesos.
  - Por **segmentos**: bloques con distinto tamaño según el proceso.
  - Por **segmentos paginados**: Los segmentos de tamaño distinto se dividen a su vez en páginas de mismo tamaño.
- A cada proceso se le asignan páginas/segmentos diferentes, consiguiendo con esto espacios protegidos para cada uno.

# Gestión de la entrada/salida



- El SO se suele encontrar con problemas derivados de las diferentes velocidades de funcionamiento de los dispositivos. Esto es tratado por este módulo presentándolo como una cuestión independiente del dispositivo.

Por ejemplo: en las salidas es frecuente la utilización de spoolers, los datos se almacenan temporalmente en una cola situada en un dispositivo de almacenamiento masivo hasta que el periférico requerido quede libre.

# Gestión de los dispositivos de almacenamiento masivo



- Los datos y programas de un dispositivo de almacenamiento masivo se mantienen en ficheros.
- El SO supervisa la creación, actualización y eliminación de estos ficheros.
- Mantiene un directorio con todos los ficheros que existen en el sistema en cada momento.
- Cada fichero está dotado de un conjunto de privilegios de acceso, que indican la extensión con la que pueden compartir la información contenida en el fichero. El sistema operativo vigila que estos privilegios no sean violados.



# Fundamentos de Ingeniería Informática

---



## TEMA 7

### Redes



# Objetivos

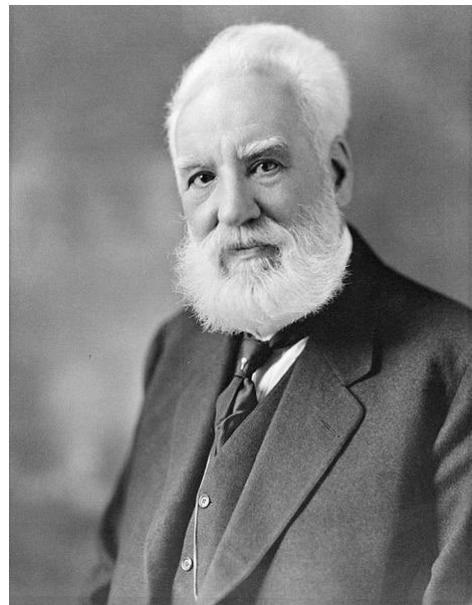
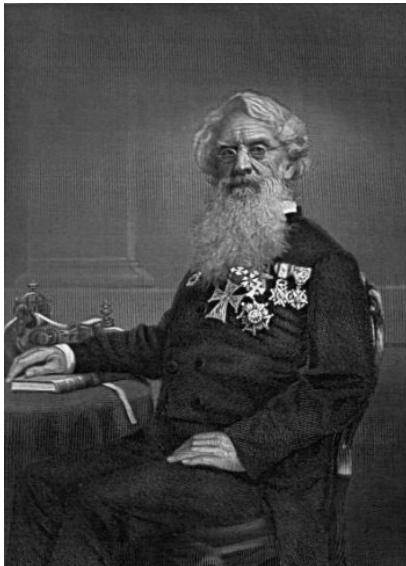


- Definir conceptos básicos relacionados con las redes y TCP/IP.
- Distinguir entre redes LAN, MAN y WAN, y las tecnologías utilizadas comúnmente en cada una de ellas para la transmisión de información.
- Contrastar el modelo OSI con el modelo TCP/IP.
- Diferenciar nombre, dirección IP y dirección física y explicar los procesos de traducción entre los tres.
- Explicar cómo se produce el envío de información a través de internet.

# Orígenes



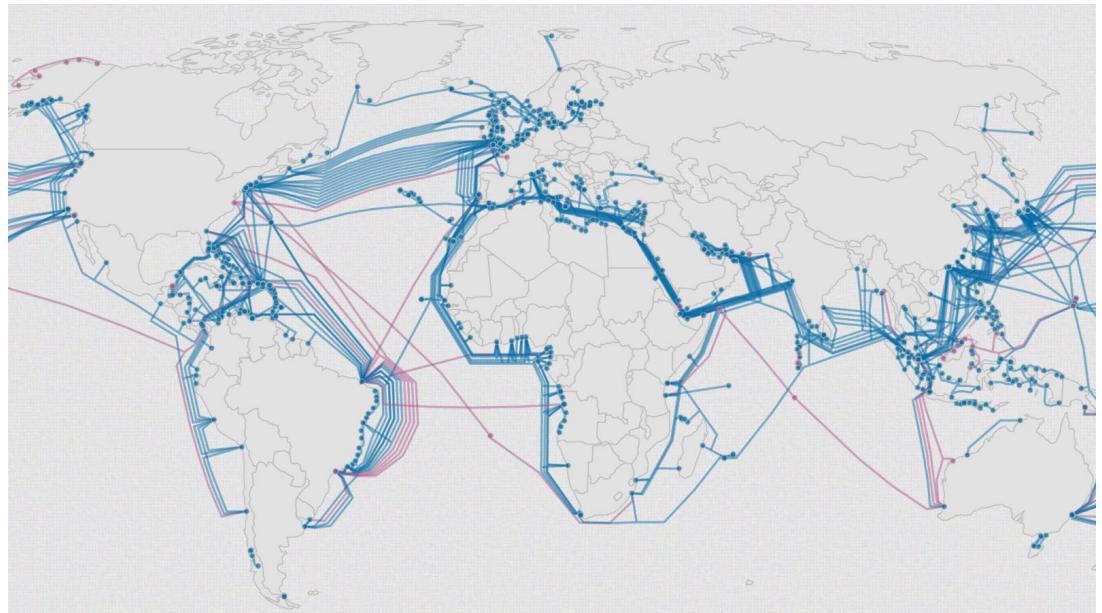
- **Telégrafo** (Samuel Morse, 1832 - 1844): Transmisión de mensajes.
- **Teléfono** (Alexander Graham Bell, 1876): Transmisión de voz.



# Actualidad

---

- Miles de millones de dispositivos a nivel mundial conectados entre sí a través de Internet, compartiendo:
  - Recursos Hardware.
  - Software y aplicaciones.
  - Información: mensajes de texto, documentos, voz, imágenes, vídeos...



# Actualidad



## THE INTERNET IN 2023 EVERY MINUTE



Created by: eDiscovery Today & LTMG

# Alcance de las redes



**LAN:** redes de área local

Alcance



Velocidad

**WAN:** redes de área extensa

**MAN:** redes de área metropolitana



# Medios físicos para la transmisión

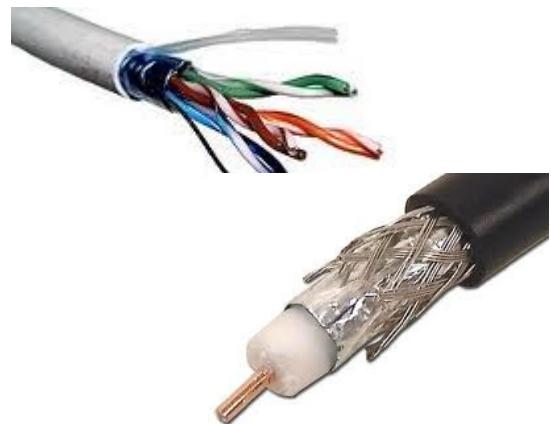


- LAN

Alcance: de unos metros a unos pocos kilómetros

Velocidad: de 10 a 100 Mbps

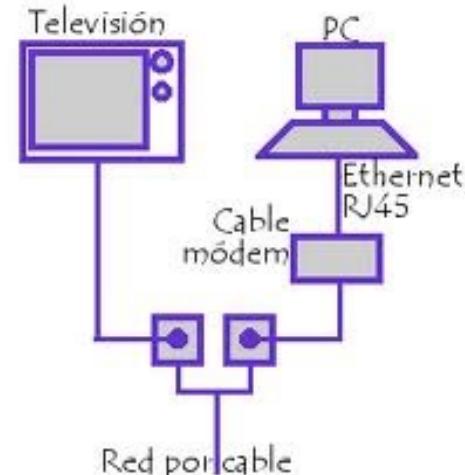
- Cable
  - Par trenzado (LAN pequeñas)
  - Cable coaxial (LAN grandes)
- Inalámbricas
  - Infrarrojos
  - Radiofrecuencia
    - Bluetooth (IEEE 802.15)
    - Wifi (IEEE 802.11)



# Medios físicos para la transmisión



- MAN
  - Cable
    - Fibra óptica
    - Red telefónica
      - Módem
      - Banda ancha: ADSL
    - Red de televisión por cable
      - Cable-módem
  - Inalámbricas
    - Wimax (IEEE 802.16)





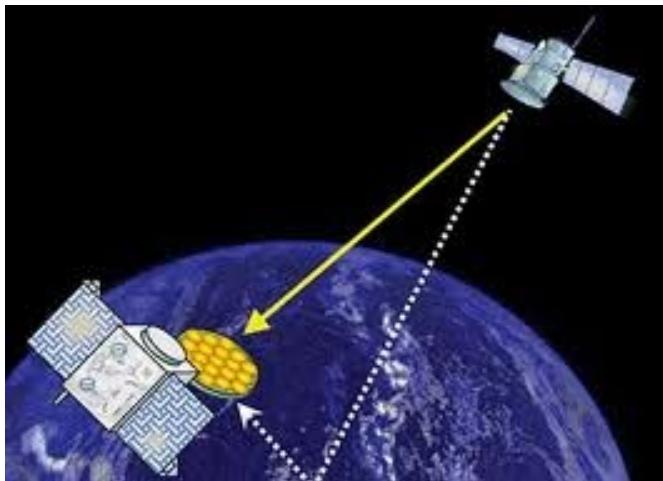
# Medios físicos para la trasmisión

- WAN

Integran múltiples redes

Utilizan redes de comunicación públicas

- Redes de telefonía
- Satélites



# Funcionamiento de una red de ordenadores: Modelo OSI



- La implementación de todas las tareas que implica la transmisión de información por redes se organiza en funciones o **protocolos** organizados por niveles, de manera que cada nivel implementa sus funciones basándose en los protocolos del nivel inferior. Así se reduce la complejidad del diseño.
- El **MODELO OSI (Open System Interconnection)** es un modelo teórico que define siete capas, y es utilizado como referencia para la definición de arquitecturas de interconexión de sistemas.
- Fue desarrollado en 1984 por la **ISO (International Standard Organization)**

## LA PILA OSI

**Nivel de Aplicación**  
Servicios de red a aplicaciones

**Nivel de Presentación**  
Representación de los datos

**Nivel de Sesión**  
Comunicación entre dispositivos de la red

**Nivel de Transporte**  
Conexión extremo-a-extremo y fiabilidad de los datos

**Nivel de Red**  
Determinación de ruta e IP  
(Direccionamiento lógico)

**Nivel de Enlace de Datos**  
Direccionamiento físico (MAC y LLC)

**Nivel Físico**  
Señal y transmisión binaria

Fuente:

<http://3con14.es/i2013/index.php/temas/hardware-tema-1/14-dispositivos-modelos-conceptuales.html>

# Especificaciones para las capas física y de enlace del modelo OSI

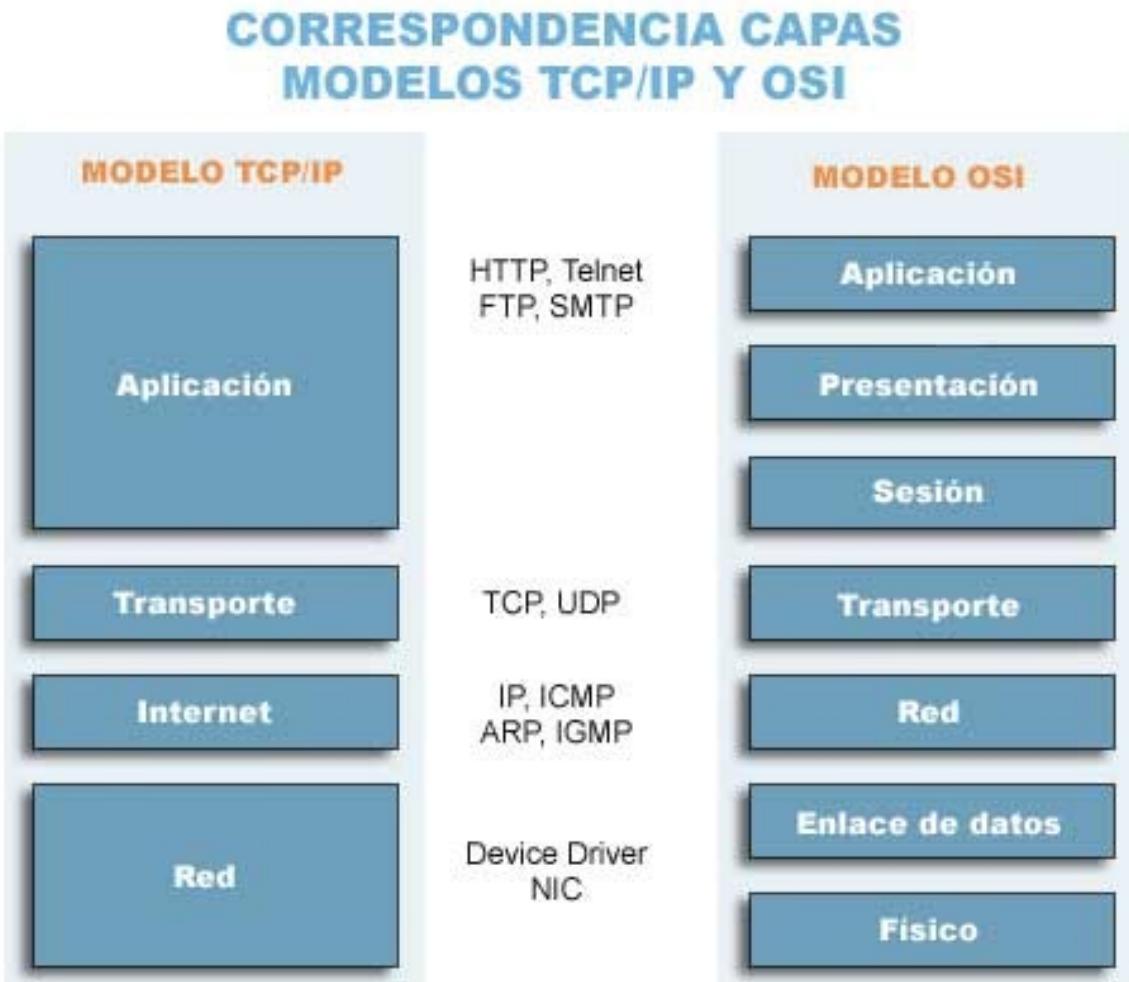


- Ethernet
  - Estándar (IEEE 802.3) para LAN que define características del cableado a nivel físico y formatos de tramas para el nivel de enlace.
  - Puede aplicarse sobre par trenzado, cable coaxial o fibra óptica.
  - Existen diferentes tecnologías ethernet que ofrecen velocidades de transmisión distintas.
- Otras redes:
  - Token Ring (estándar IEEE 802.5).
  - FDDI (Fiber Distributed Data Interface).

# Protocolos de Internet: TCP/IP



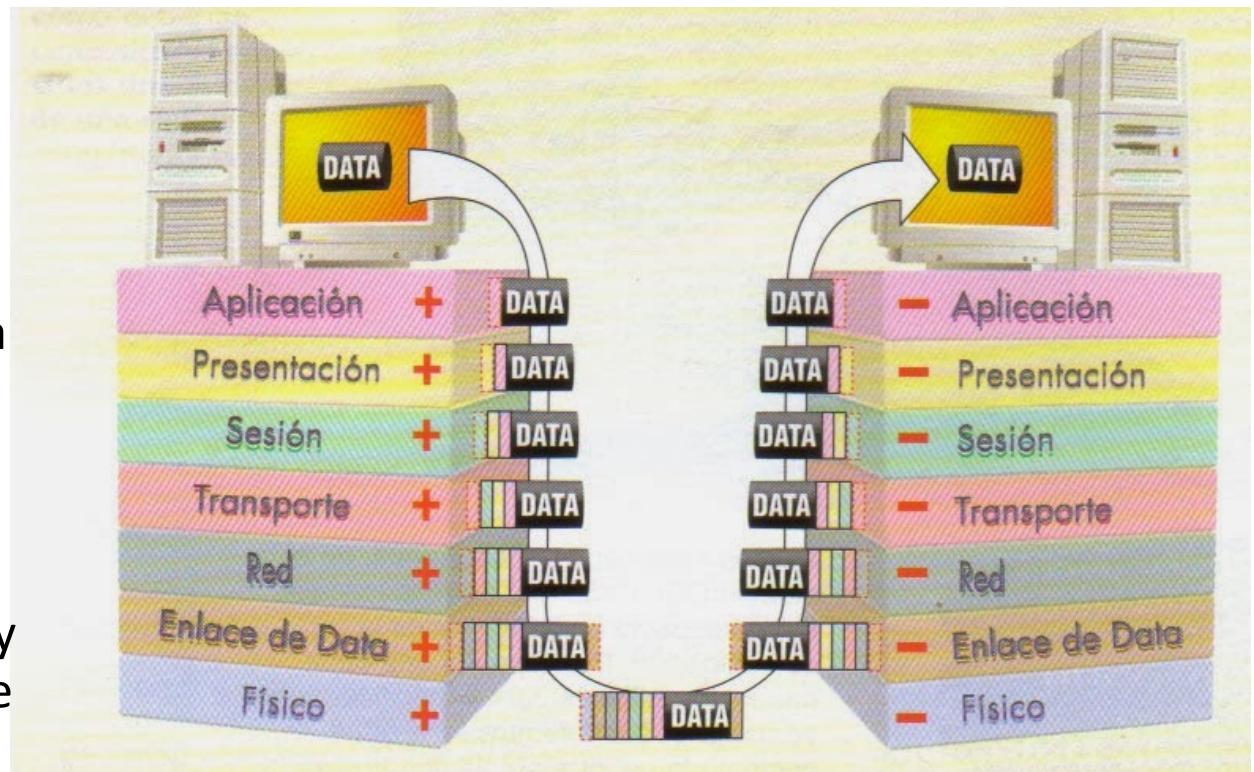
- TCP/IP es un conjunto de protocolos (entre los que se encuentran los protocolos TCP e IP, que le dan nombre) organizados en 4 niveles
- Surge en los años 70, después de que la agencia norteamericana DARPA pusiera en marcha la red ARPANET, considerada el origen de Internet
- Las primeras conexiones de ordenadores en ARPANET fueron a través de líneas telefónicas



# Funcionamiento del modelo OSI: encapsulamiento de los datos



- Cada protocolo añade a los datos que le llegan del protocolo anterior información de control, denominada cabecera (Header).
- La información final que se transmite (colección de bits) comprende los datos y todas las cabeceras de todos los protocolos por los que han pasado.

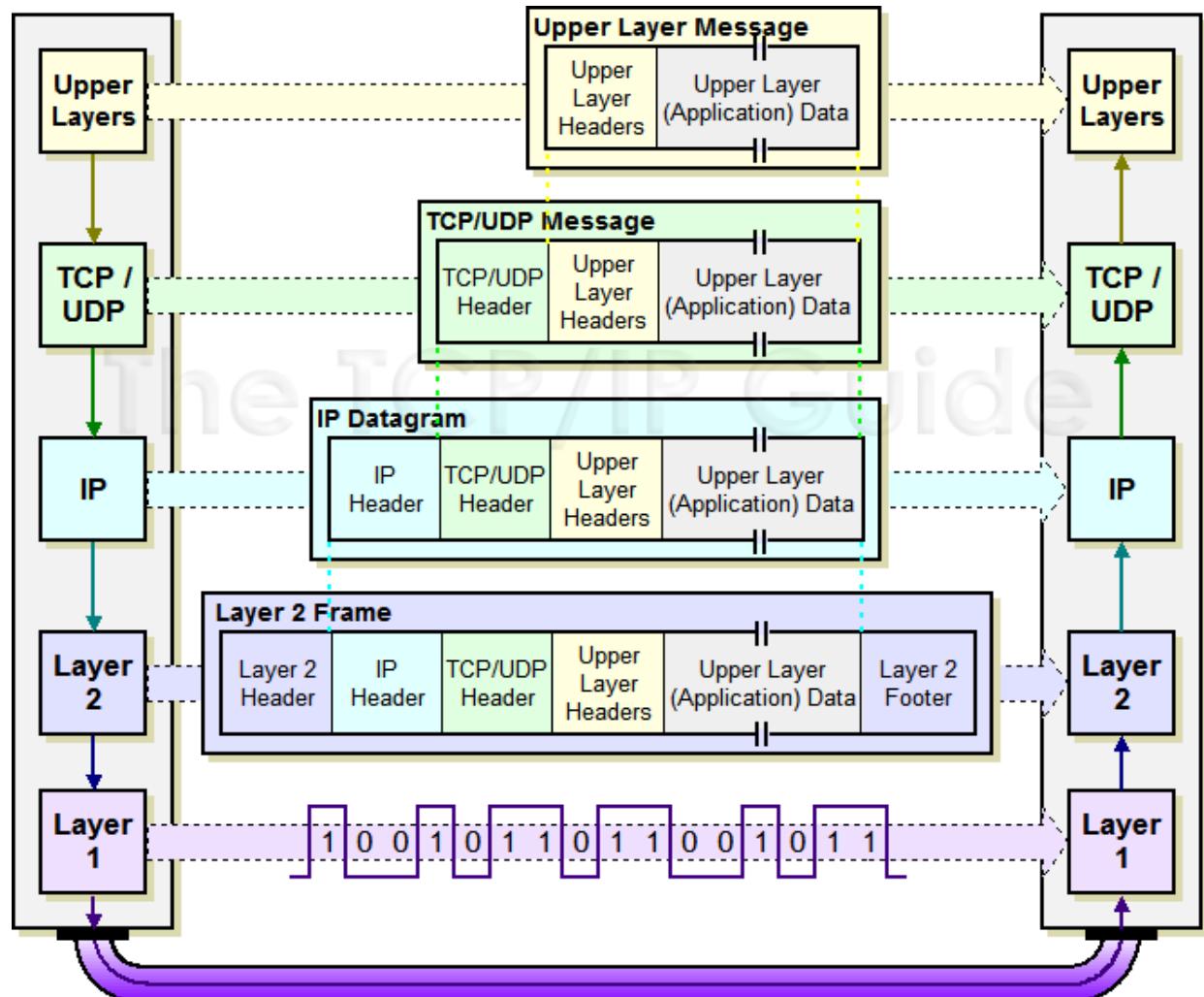


Fuente de la imagen:  
<https://vane395.blogspot.com.es/2013/11/el-modelo-osi-tras-el-crecimiento-en-la.html>

# Funcionamiento de TCP/IP: encapsulamiento de los datos



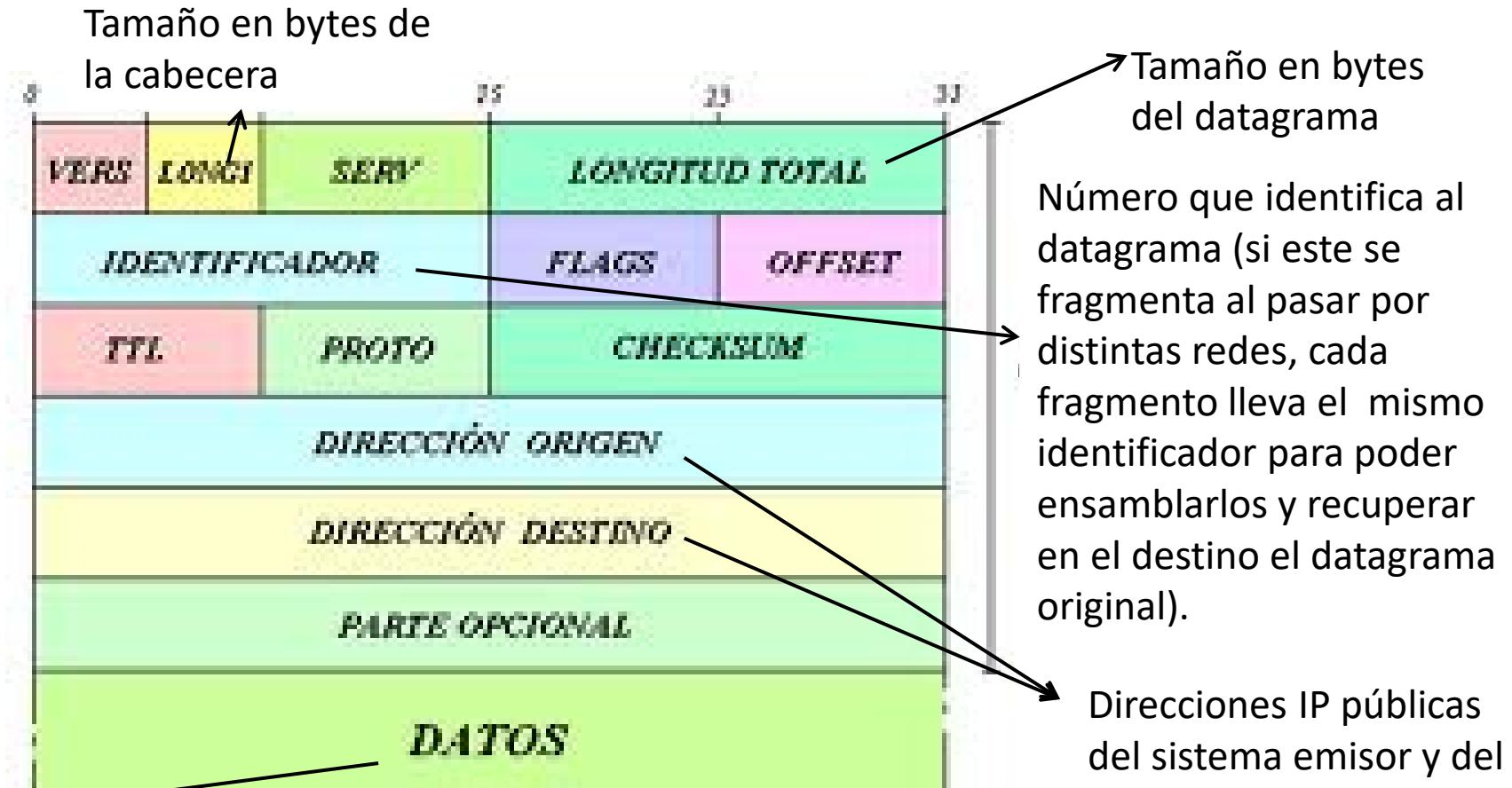
- Como en el modelo OSI, cada nivel añade una cabecera.
- A nivel de IP los paquetes se denominan **datagramas** y a nivel de Red, **tramas**.



Fuente de la imagen:

[http://www.tcpipguide.com/free/t\\_IPDatagramEncapsulation.htm](http://www.tcpipguide.com/free/t_IPDatagramEncapsulation.htm)

# Funcionamiento de TCP/IP: formato de un datagrama (cabecera + datos)



Contiene la información que viene de los protocolos de la capa superior (TCP, UDP,...), que contienen a su vez una cabecera + los datos provistos por la correspondiente aplicación

# Identificación de dispositivos en una red

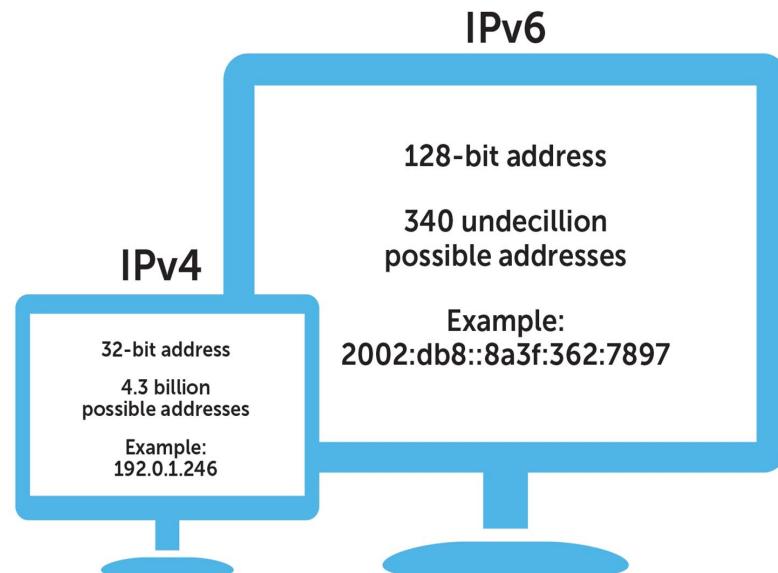


- **Nombre:** es traducido a dirección IP por el servicio **DNS**
  - Existe también un servicio de traducción de nombres a IP dinámicas, denominado **DNS dinámico**. Esto permite utilizar software de servidor en un PC con IP variable.
- **Dirección IPv4** (32 bits). Ej: 173.194.34.240
  - **IPv6** (128 bits). Nueva versión de dirección IP planteada para resolver el problema del agotamiento de las direcciones IP, dado que cada vez hay más máquinas conectadas a internet.
- **Dirección física:** dirección asociada al dispositivo de red
  - Ej. MAC (48 bits). Ej: 00-25-34-E2-2F-11 (única para cada adaptador de red, se usa en redes Ethernet y Wifi, entre otras)

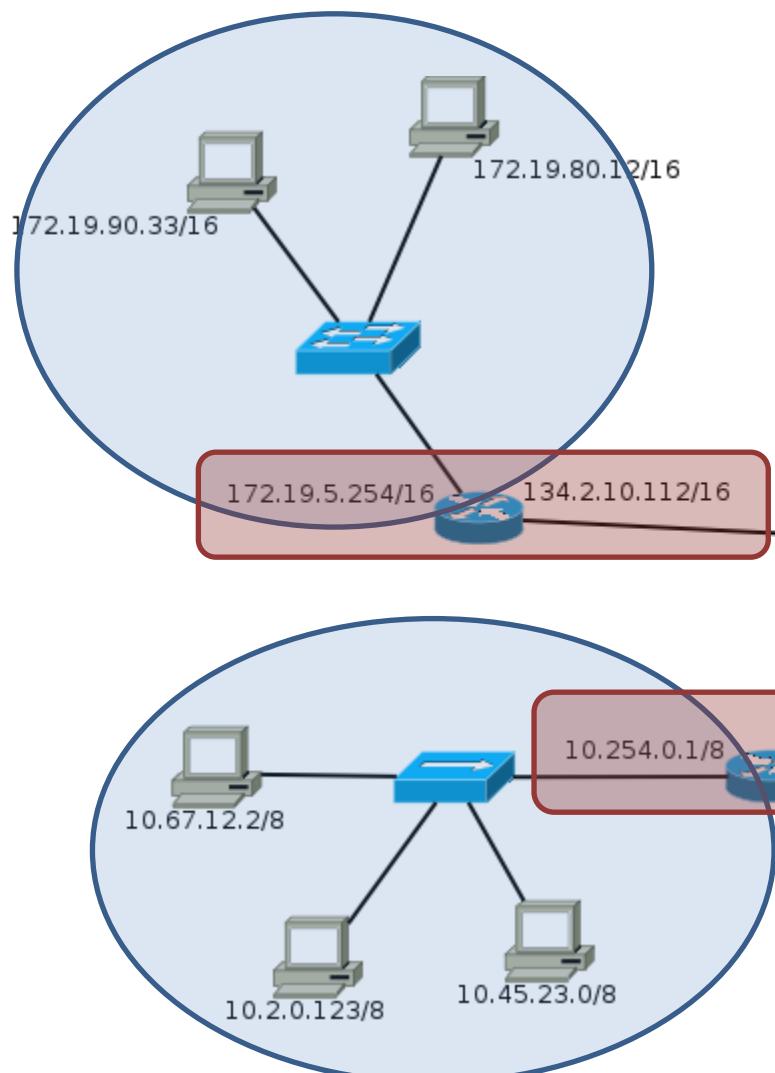
# Identificación de dispositivos en una red



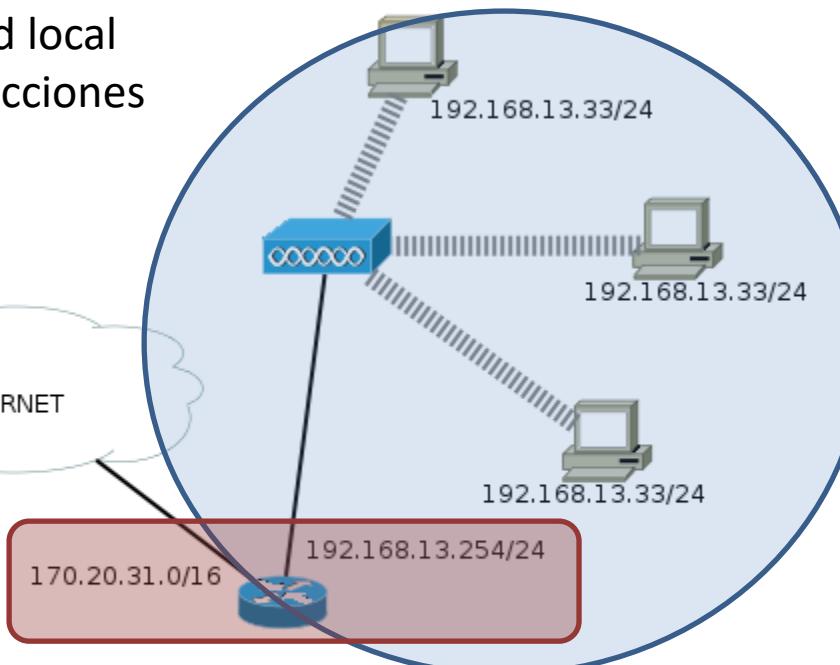
- Direcciones IPv4
  - Existen direcciones públicas y direcciones privadas.
    - Rango de direcciones privadas:  
10.x.x.x 1, de 172.16.x.x.a  
172.31.x.x, 192.168.x.x
  - Las direcciones privadas se utilizan dentro de redes locales y han de ser traducidas a direcciones públicas si se desea conectarse a internet. De esta traducción se encarga el servicio NAT (Network Address Translation), realizado por el router o proxy que da salida a internet desde dicha red local.



# Ejemplo de redes locales conectadas a Internet

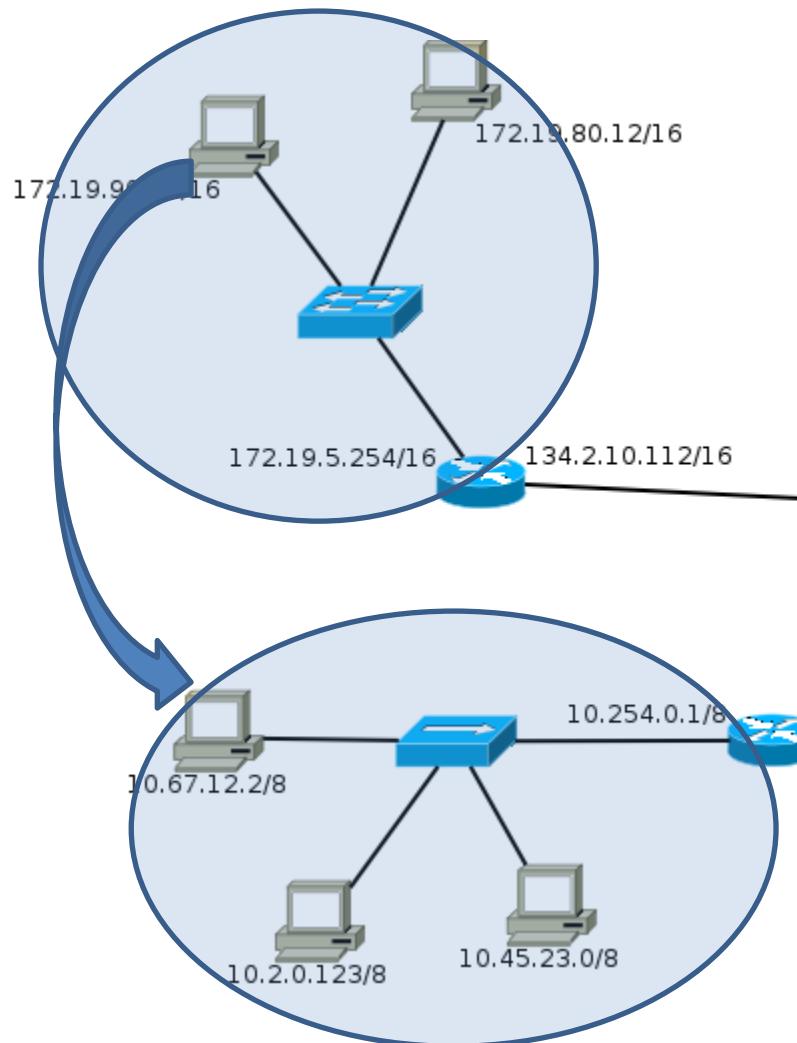


Los host conectados en cada red local tienen direcciones privadas



Los routers tienen dos interfaces de red, y por tanto, dos direcciones IP: una privada para conectarse a la red local y otra pública para conectarse a internet. Además de proporcionar servicios de enrutado, también proporcionan NAT.

# Envío de paquetes por Internet



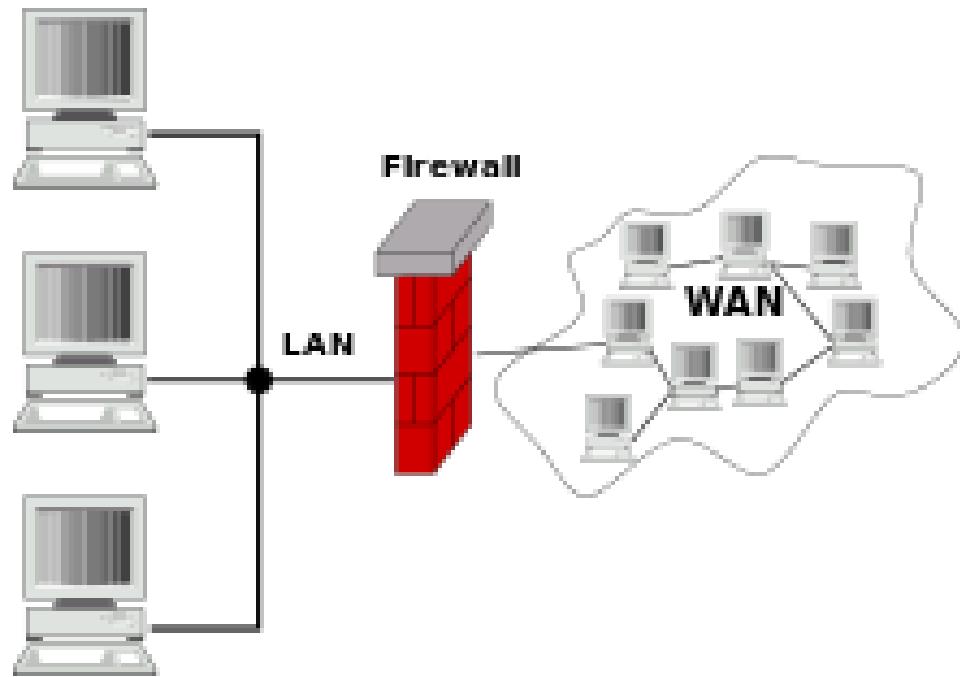
El envío de datos por internet se origina en un host de una determinada red. Si el host destino pertenece a esa misma red (puede saberse por la dirección IP de dicho destino y la máscara de subred), el envío se produce directamente a través de la dirección física de dicho host. Si no, los datos serán enviados de red en red a través de diferentes routers, hasta alcanzar la red de destino y una vez allí, el host concreto al que se dirigen los datos.

Cada paquete IP que se envía por internet, aún perteneciendo a los mismos datos de origen, puede seguir una ruta diferente hasta alcanzar el destino.

# Firewall



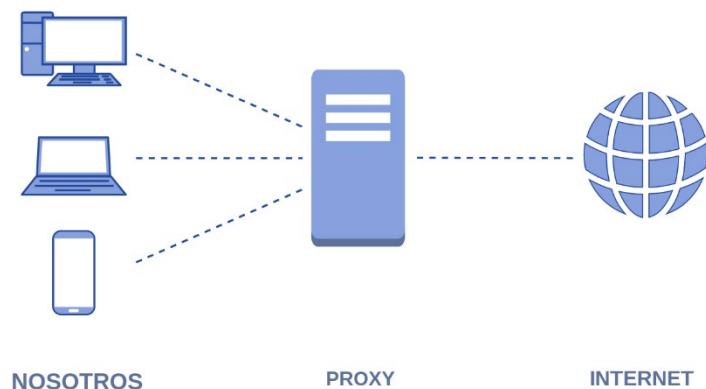
- Sistema que protege los accesos a una red, prohibiendo accesos no autorizados.



# Proxy

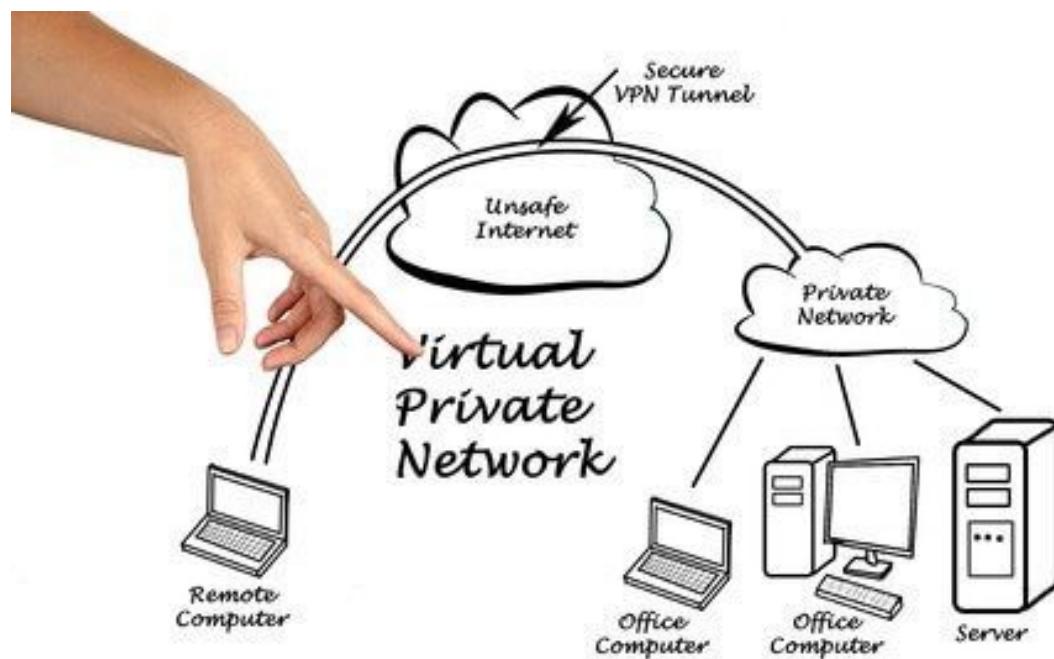


- Sistema que actúa como representante de otro, permitiendo la conexión de éste a otra red externa.
  - Proporciona caché.
  - Control de acceso y seguridad (en combinación con el firewall).
  - Servicio de traducción de direcciones (NAT).





Una red privada virtual (VPN) extiende una red privada a través de una red pública y permite a los usuarios enviar y recibir datos a través de redes públicas o compartidas como si sus dispositivos estuvieran conectados directamente a la red privada.



# Fundamentos de Ingeniería Informática

## Módulo II - Unidad 8 Nuevas Tendencias I

Profesores: Héctor Molina García

# ¿Qué es una tecnología de tendencia ?

Una **tendencia** es un cambio o desarrollo hacia algo nuevo o diferente. Esto significa que la tecnología de tendencia cambiará nuestro futuro.

- Inteligencia Artificial (IA).
- Aprendizaje automático (ML).
- Visión Artificial.
- Computación.
- Sistemas de control.
- Internet de las Cosas (IoT).
- Blockchain.



# Tecnologías en tendencia - La transformación digital

La tecnología está cambiando nuestros comportamientos.



La tecnología está cambiando nuestros comportamientos.



# Tecnologías en tendencia - La transformación digital

La tecnología está cambiando nuestros comportamientos.



La tecnología está cambiando nuestros comportamientos.

Dear Santa,  
How are you? I'm good.  
Here is what I want for  
Christmas.  
<http://www.amazon.com/gp/product/B0032HF60>  
Mref=59\_hps\_bw\_g21\_ir03?pf\_rd\_m=ATVPDKIKXODER&pf\_rd\_Scenter=3&pf\_rd\_id=9XW442FH1K03Y78MWQNM&pf\_rd\_t=101&pf\_rd\_i=1328901542&pf\_rd\_d=i=16579

Querido hijo,

Esta semana cambiaremos todos los días la contraseña del Wifi,

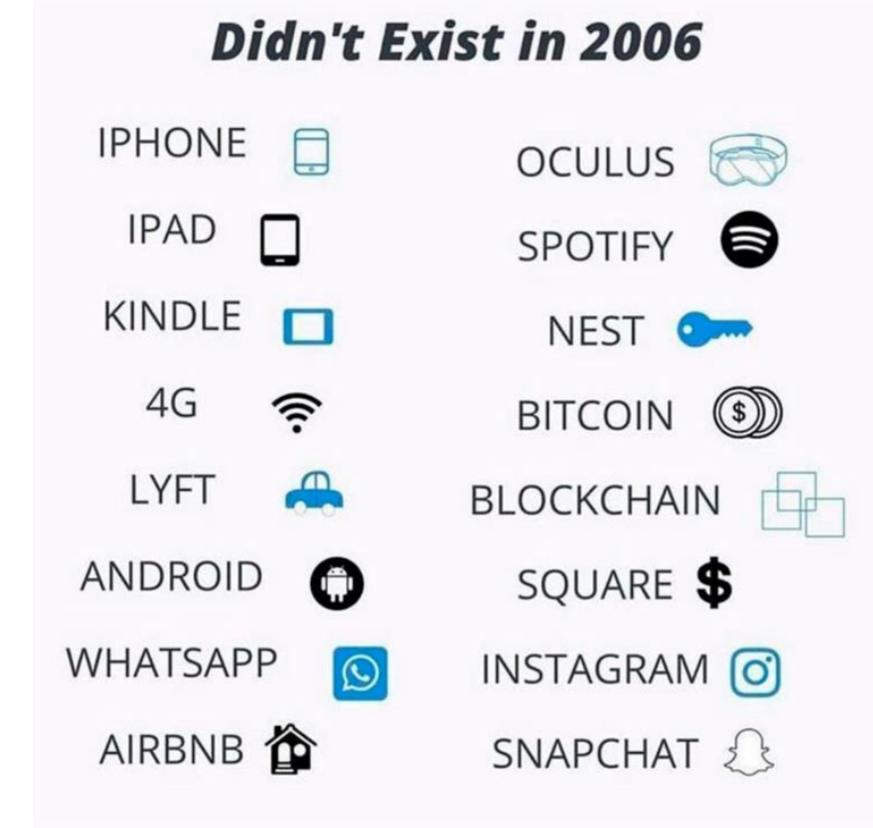
Para conseguir la de hoy.  
Tienes que:

- Limpiar tu habitación
- Lavar los platos
- Tirar la basura

~~Mama' y papa'~~ B

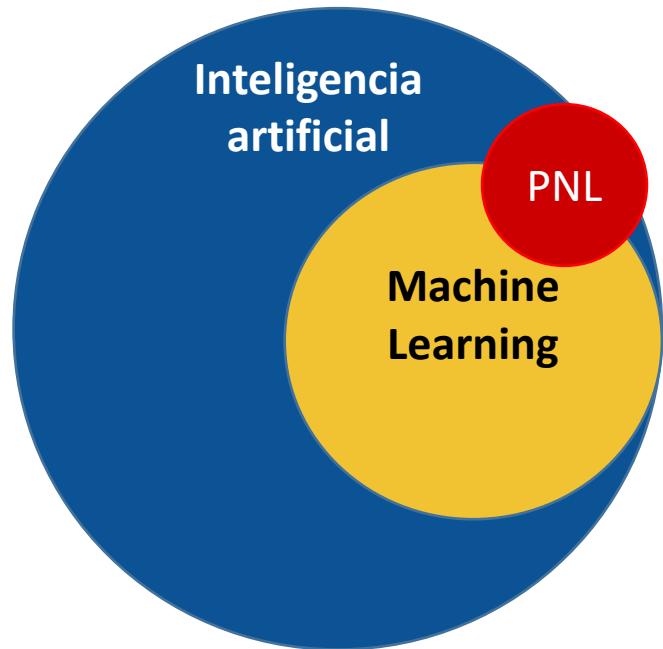


# Tecnologías en tendencia - La transformación digital



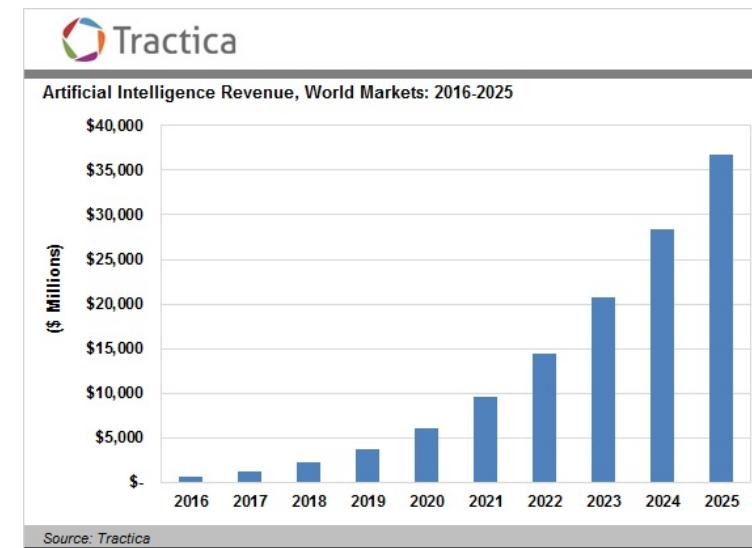
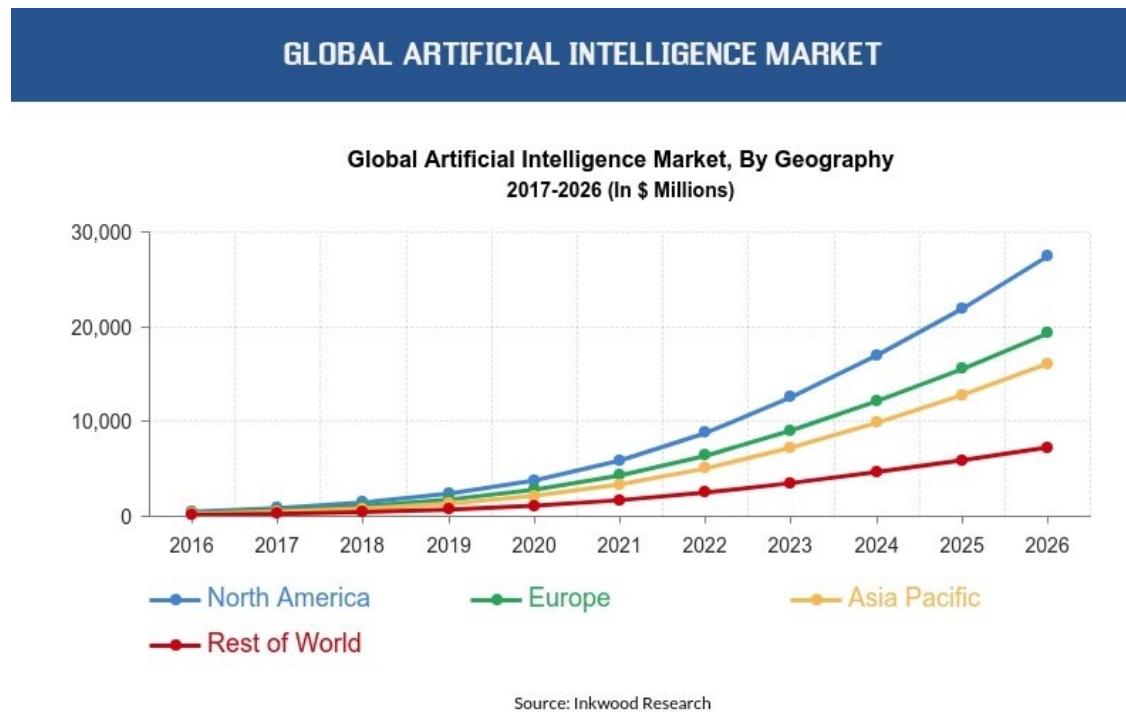
# Inteligencia artificial

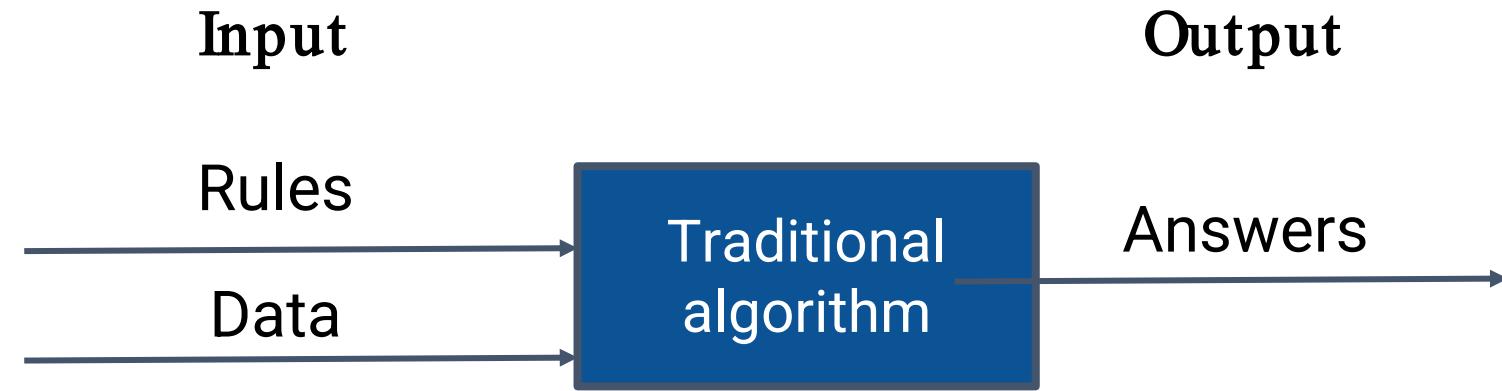
# Inteligencia Artificial (IA)



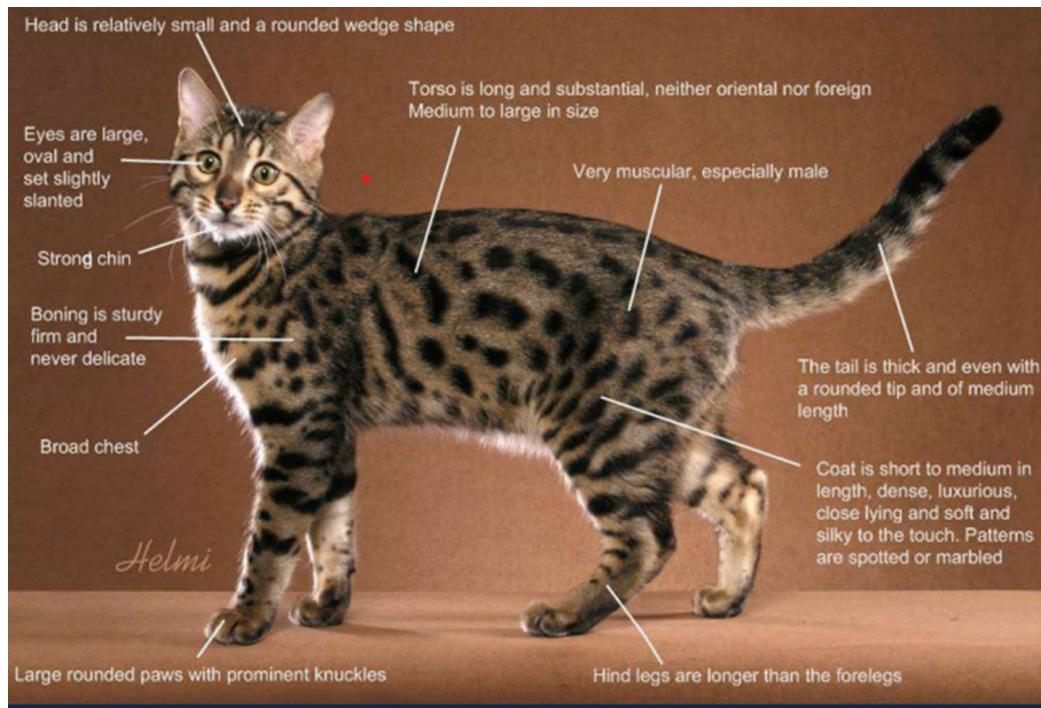
# Inteligencia Artificial (IA)

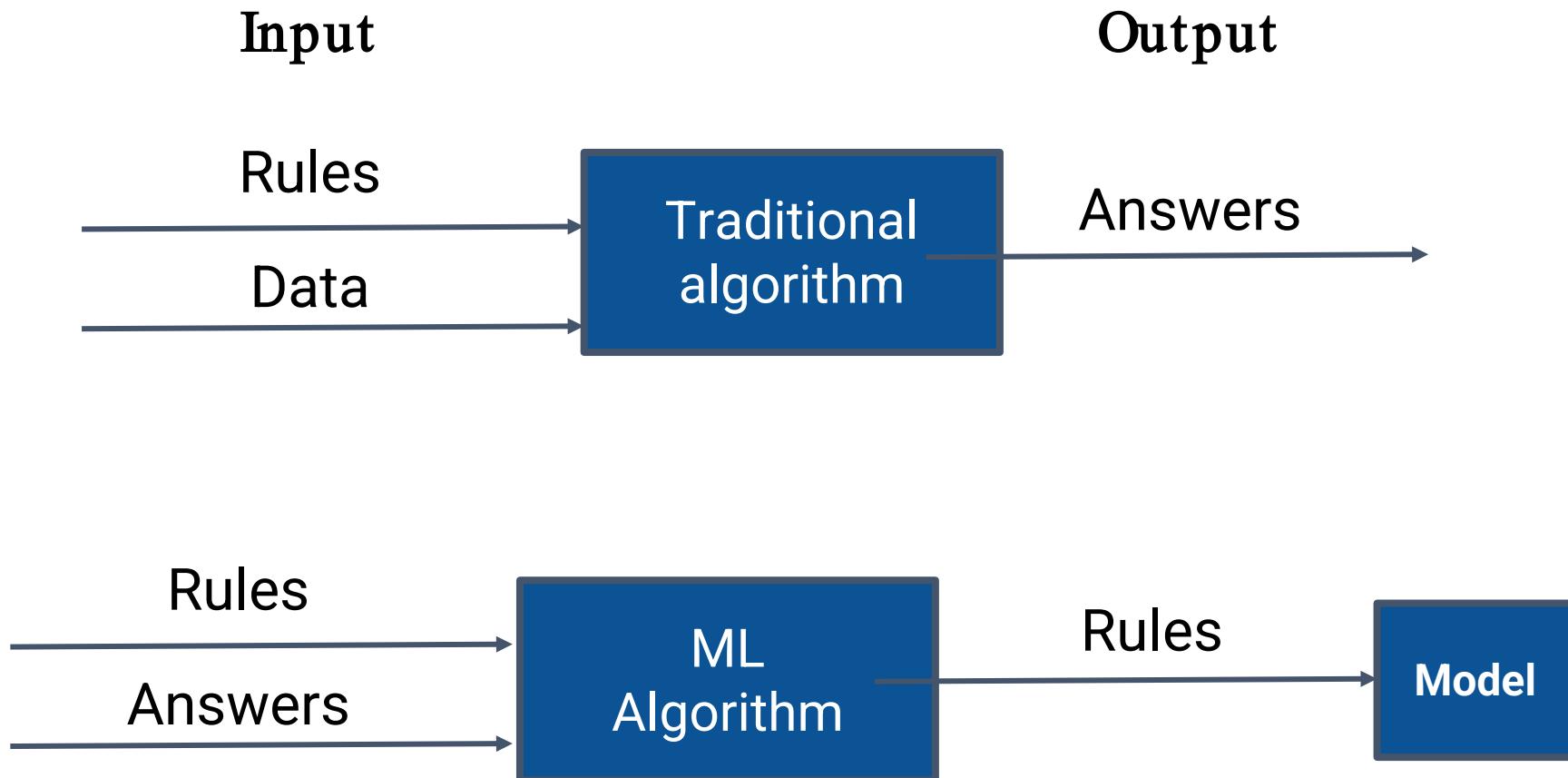
La inteligencia artificial (IA) es la inteligencia demostrada por las máquinas, a diferencia de la inteligencia natural que muestran los animales y los humanos.





Los algoritmos tradicionales funcionan mediante el uso de reglas definidas.



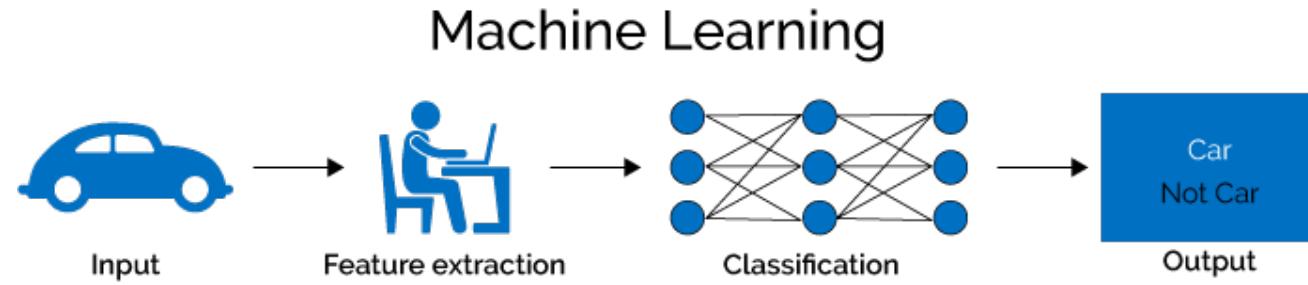


# Inteligencia artificial - Aprendizaje automático

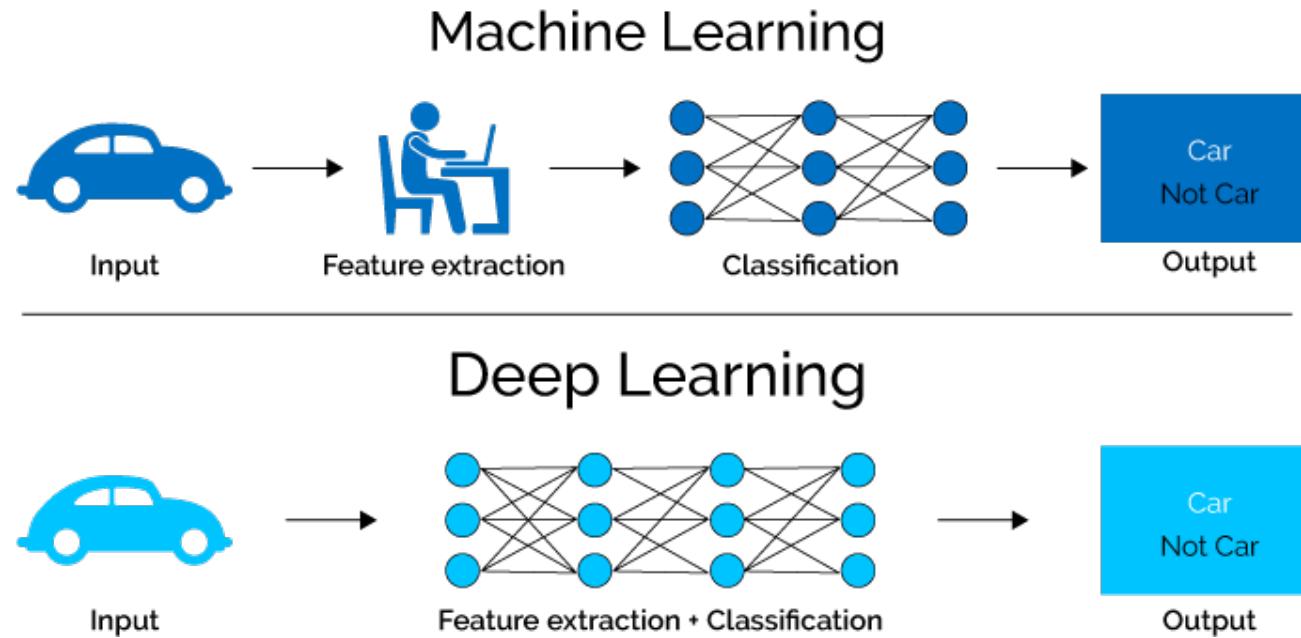
Los algoritmos de ML funcionan mediante el uso de ejemplos que intentan capturar el conocimiento que reside dentro de ellos.



El aprendizaje automático (ML) es el estudio de algoritmos informáticos que mejoran automáticamente a través de la experiencia (ejemplos de datos).



El aprendizaje automático (ML) es el estudio de algoritmos informáticos que mejoran automáticamente a través de la experiencia (ejemplos de datos).



## AlphaGo

AlphaGo es el primer jugador en derrotar a un jugador de Go profesional humano, el primero en derrotar a un campeón mundial de Go, y es posiblemente el jugador de Go más fuerte del mundo.

- Dos jugadores que juegan por turnos.
- Piedras blancas y negras.
- Modelos basados en la interacción hombre-máquina.

El jugador combina un árbol de búsqueda avanzada con redes neuronales profundas. Estas redes toman una descripción de la placa como entrada y la procesan a través de varias capas diferentes que contienen millones de neuronas.



Hay de 10 a 170 configuraciones de placa posibles en Go, mucho más que el número de átomos en el universo conocido.

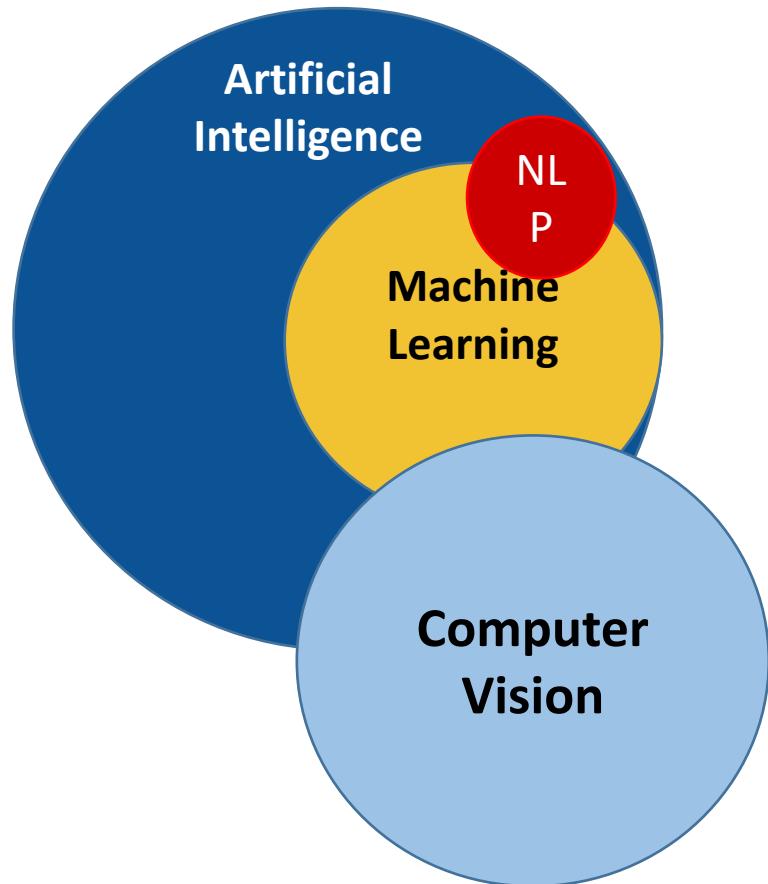


# ALPHAGO



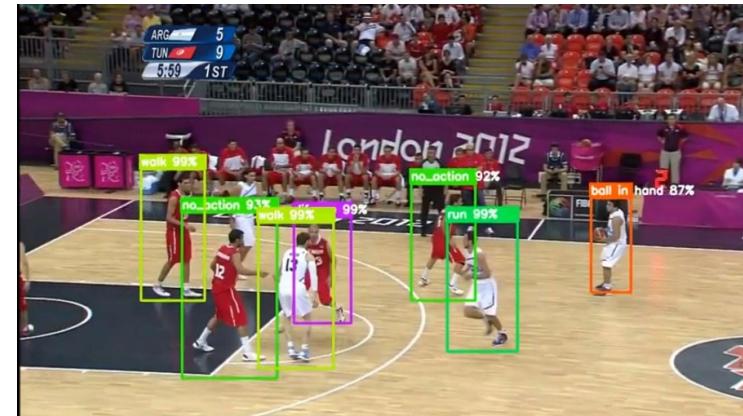
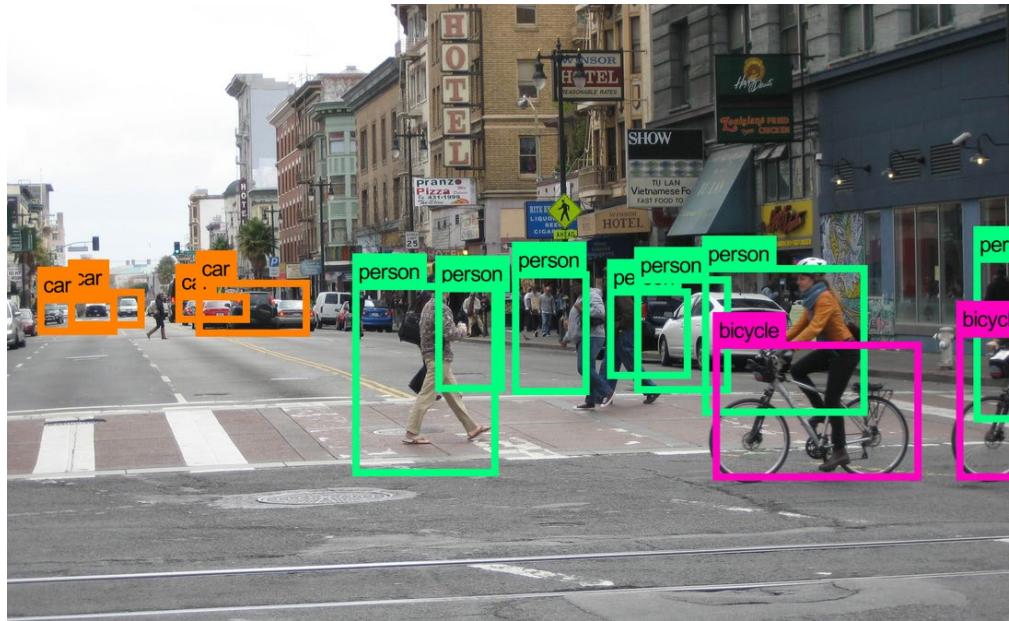
# Visión artificial

# Visión artificial

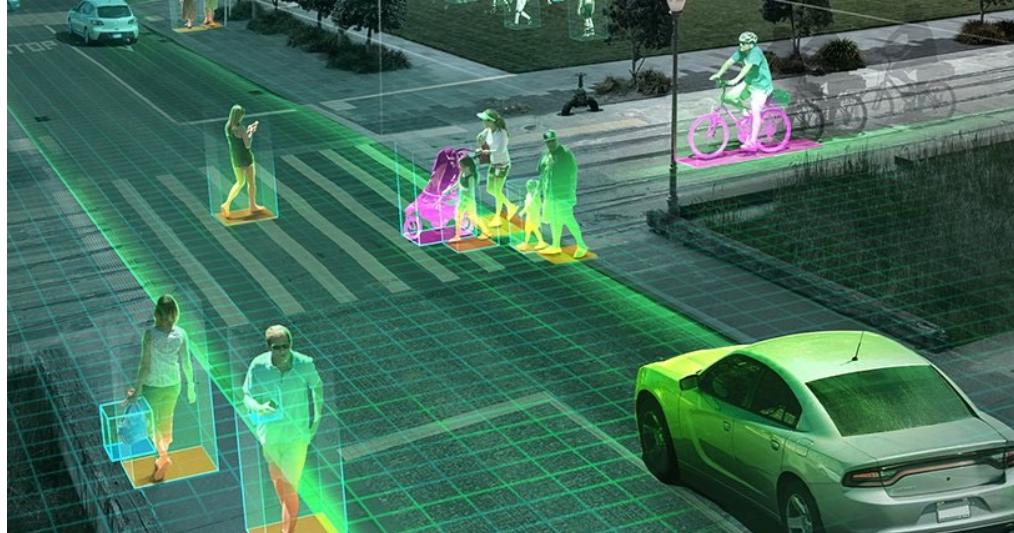
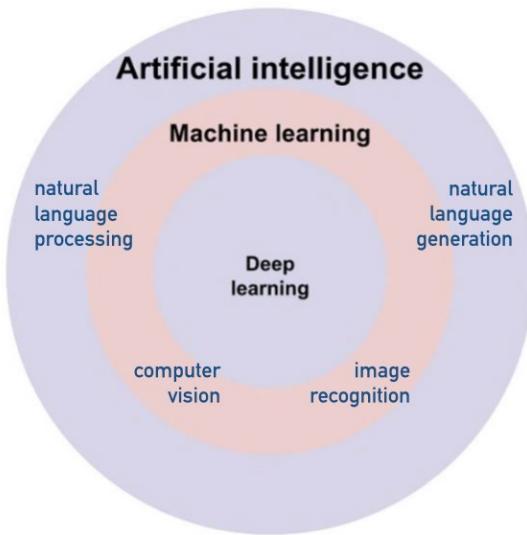


# Visión artificial

La visión artificial es un campo de la inteligencia artificial que entrena a las computadoras para interpretar y comprender el mundo visual mediante el uso de imágenes y videos de cámaras.



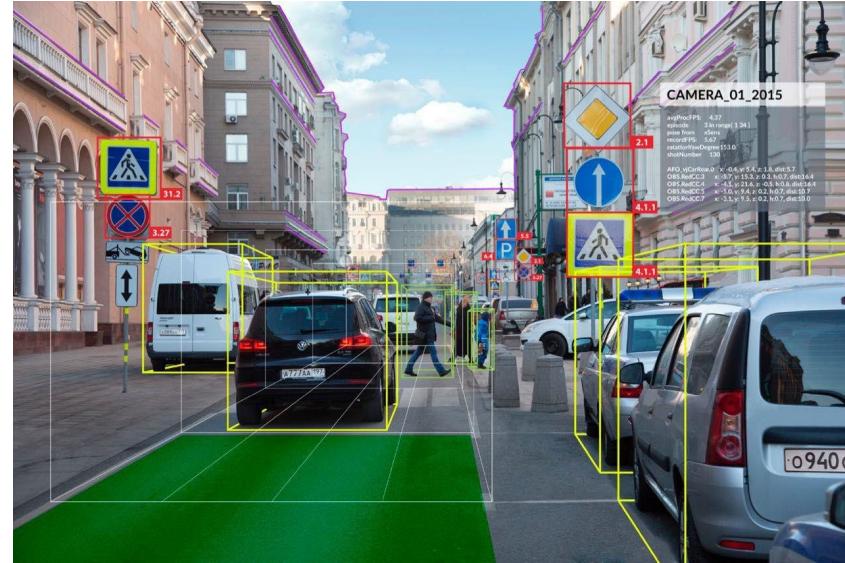
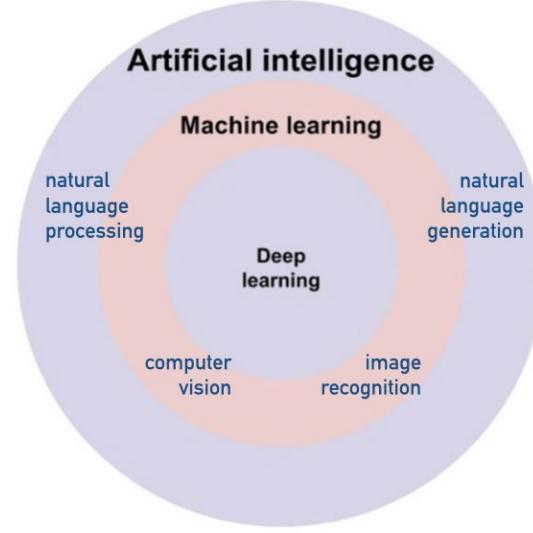
# Visión artificial



La visión artificial, en el contexto de la visión por computadora, es la capacidad de las máquinas para comprender (incluida la capacidad de inferir algo al respecto) la imagen de entrada y su contenido. La visión artificial utiliza algoritmos de procesamiento de imágenes para resolver algunas de sus tareas.

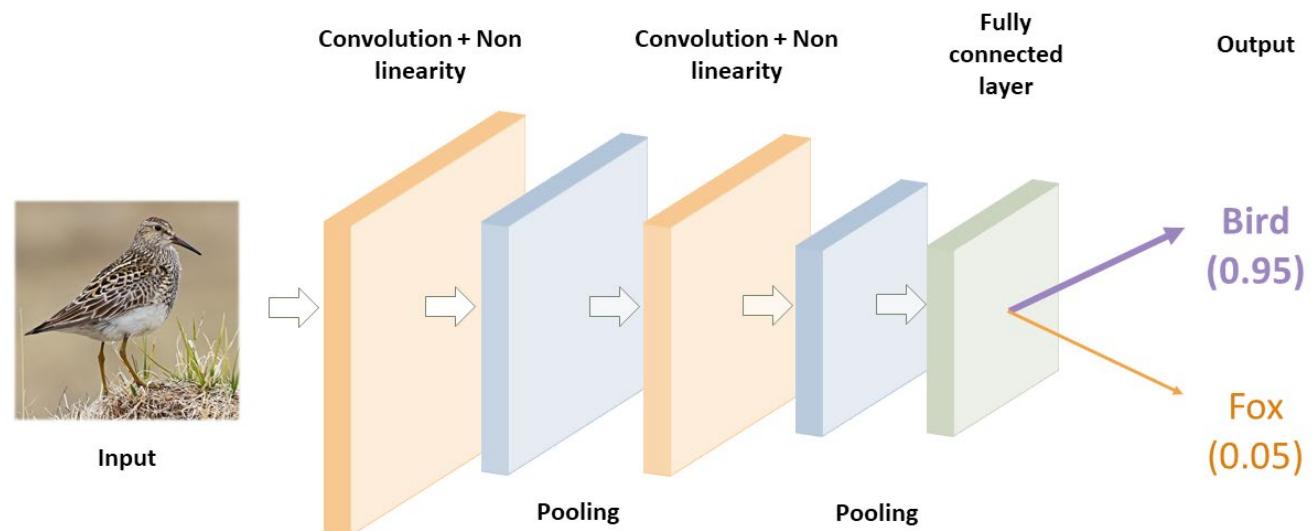


# Visión artificial



El reconocimiento de imágenes, en el contexto de la visión artificial, es la capacidad de las máquinas para identificar objetos, lugares, personas, ..., cualquier cosa que esté en la imagen de entrada. El reconocimiento de imágenes utiliza algoritmos de procesamiento de imágenes para resolver algunas de sus tareas.

Una red neuronal convolucional (ConvNet / CNN) es un algoritmo de aprendizaje profundo que puede tomar una imagen de entrada, asignar importancia (características y sesgos aprendibles) a varios aspectos / objetos en la imagen y poder diferenciar uno del otro.



# Visión artificial



¿Fue hecho por un humano?

# Visión artificial

**real picture**



Deep Neural Network



**Input picture**



¿Fue hecho por un humano?

**NO**



Universidad  
Francisco de Vitoria  
**UFV** Madrid

¿Qué más podemos hacer usando la visión artificial?

Deep fake

Art generation

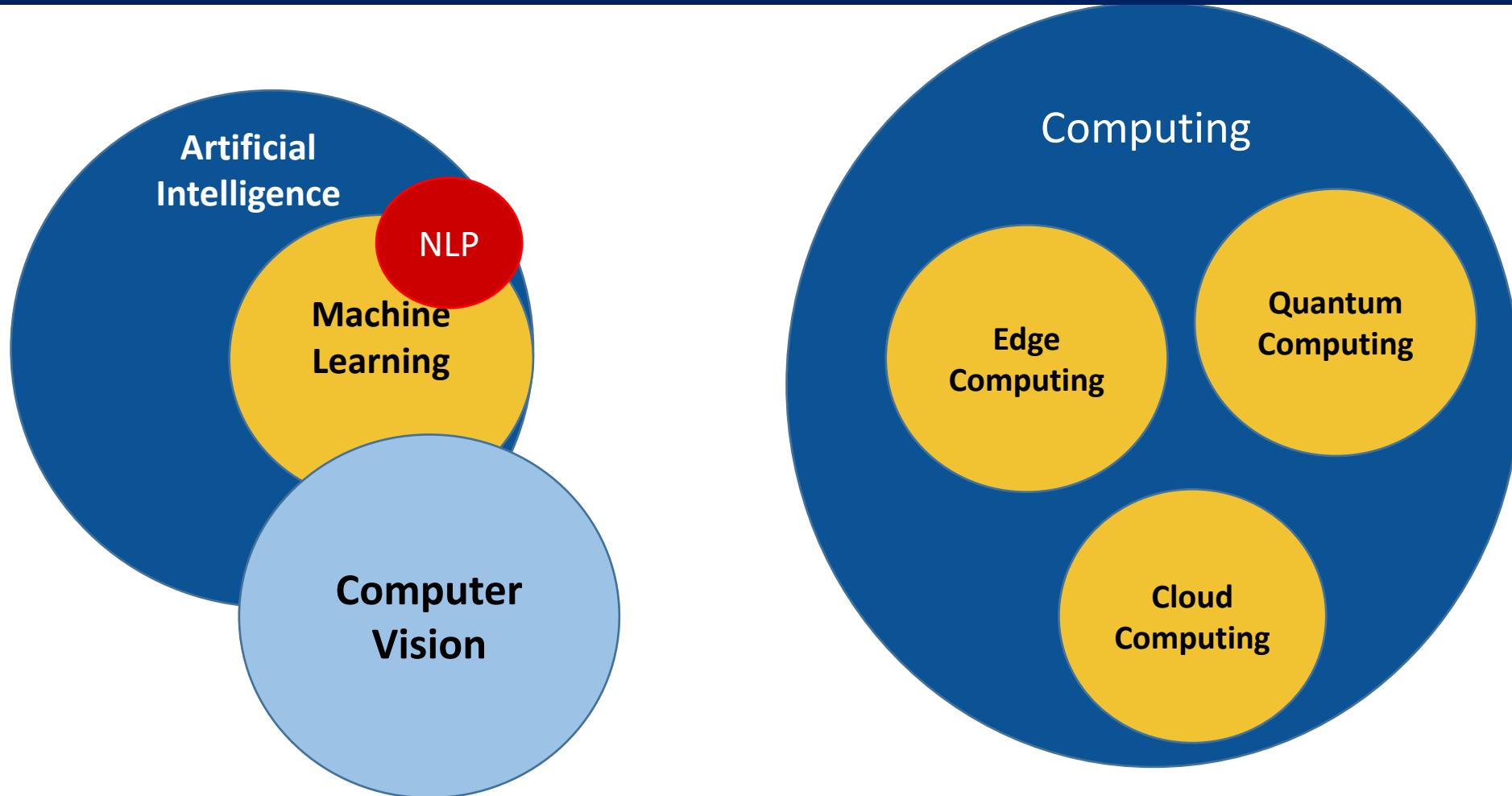


Emoji Generation

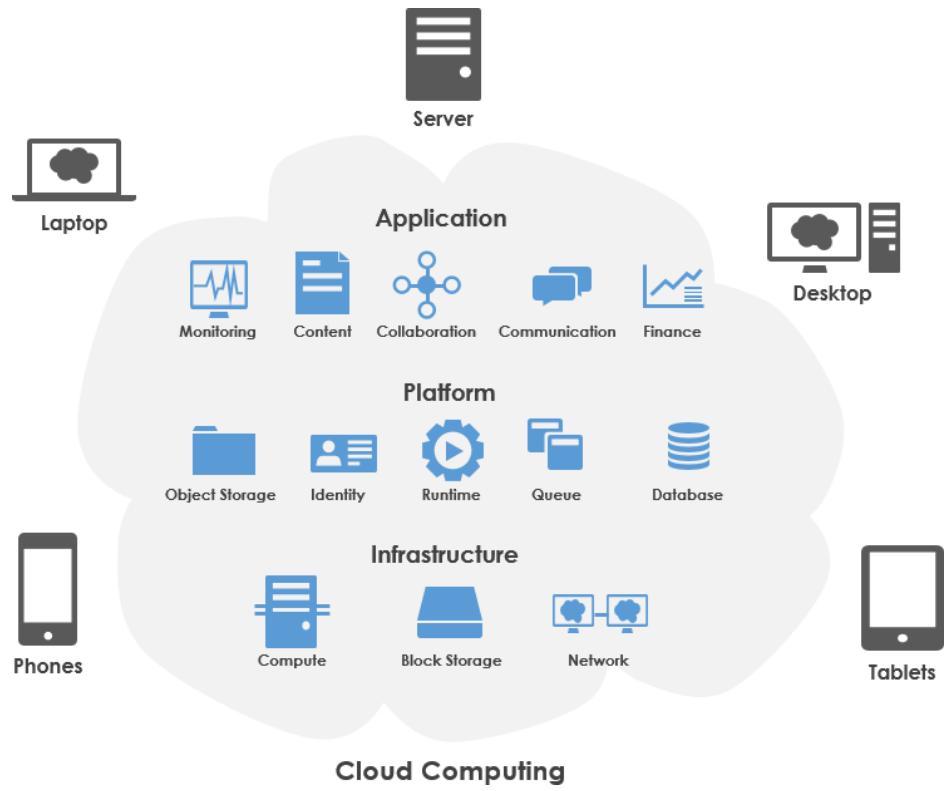


# Computación - Nube

# Computación



# Computation - Cloud computing



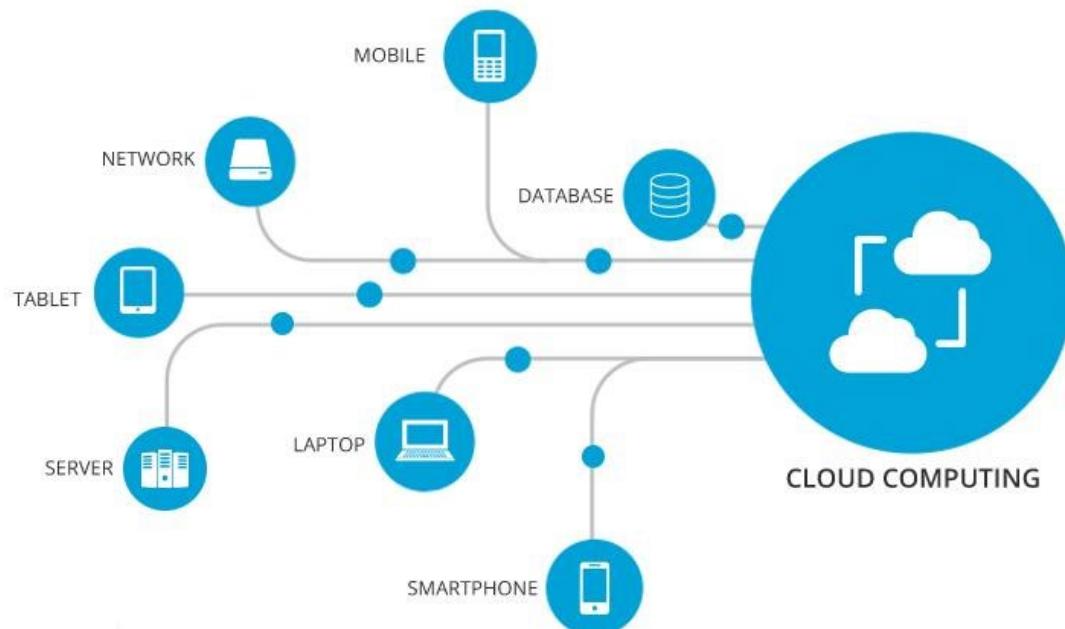
Alibaba Cloud



## Computation - Cloud computing

La computación en la nube es la disponibilidad bajo demanda de los recursos del sistema informático, especialmente el almacenamiento de datos y la potencia informática, sin una gestión activa directa por parte del usuario.

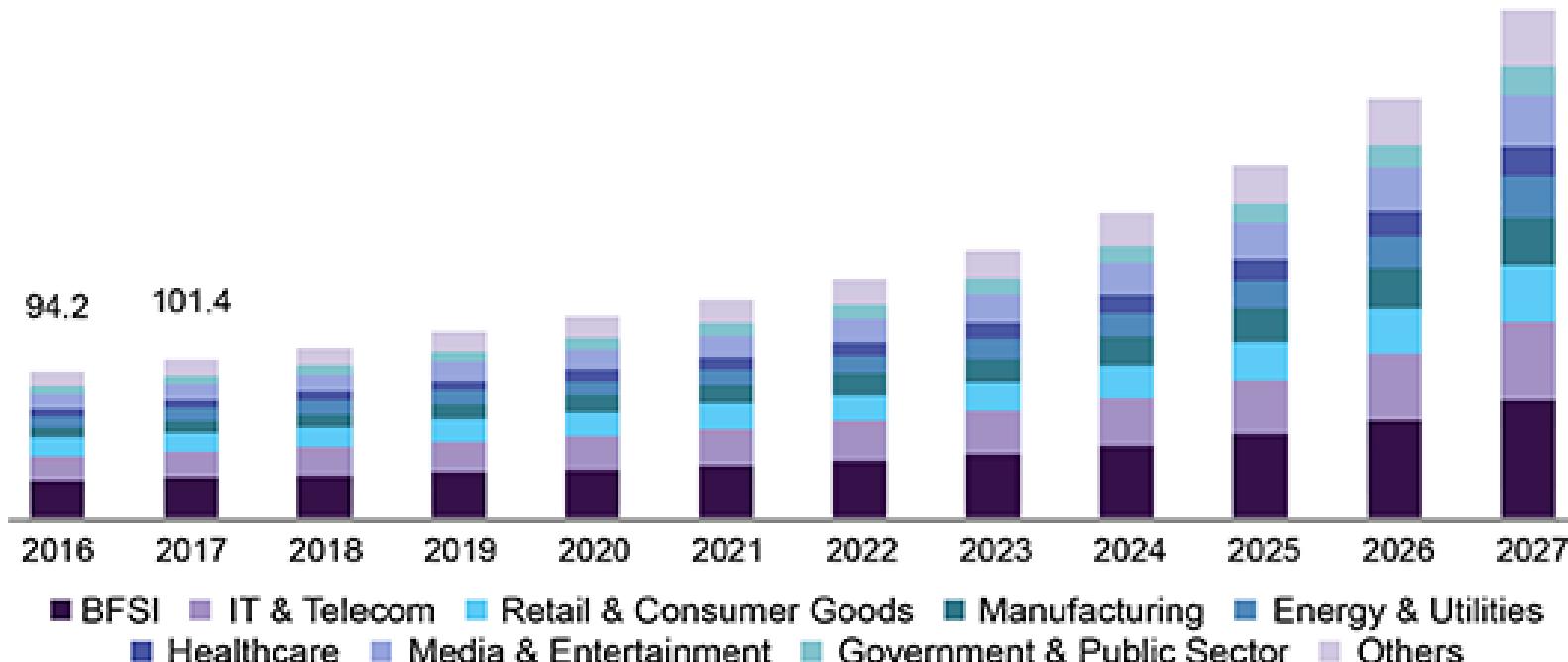
- Pago por uso.
- Escala global.
- Gran rendimiento.



# Computation - Cloud computing



U.S. cloud computing market size, by end use, 2016 - 2027 (USD Billion)



Source: [www.grandviewresearch.com](http://www.grandviewresearch.com)

# Computation - Cloud computing

## Public Cloud

Typically have massive amounts of available space, which translates into easy scalability. Recommended for software development and collaborative projects.

## Private Cloud

Usually reside behind a firewall and are utilized by a single organization. Recommended for businesses with very tight regulatory requirements

## Hybrid Cloud

Combine public clouds with private clouds to allow the two platforms to interact seamlessly. Recommended for businesses balancing big data analytics with strict data privacy regulations.

## Community Cloud

A collaborative, multi-tenant platform used by several distinct organizations to share the same applications. Users are typically operating within the same industry or field.

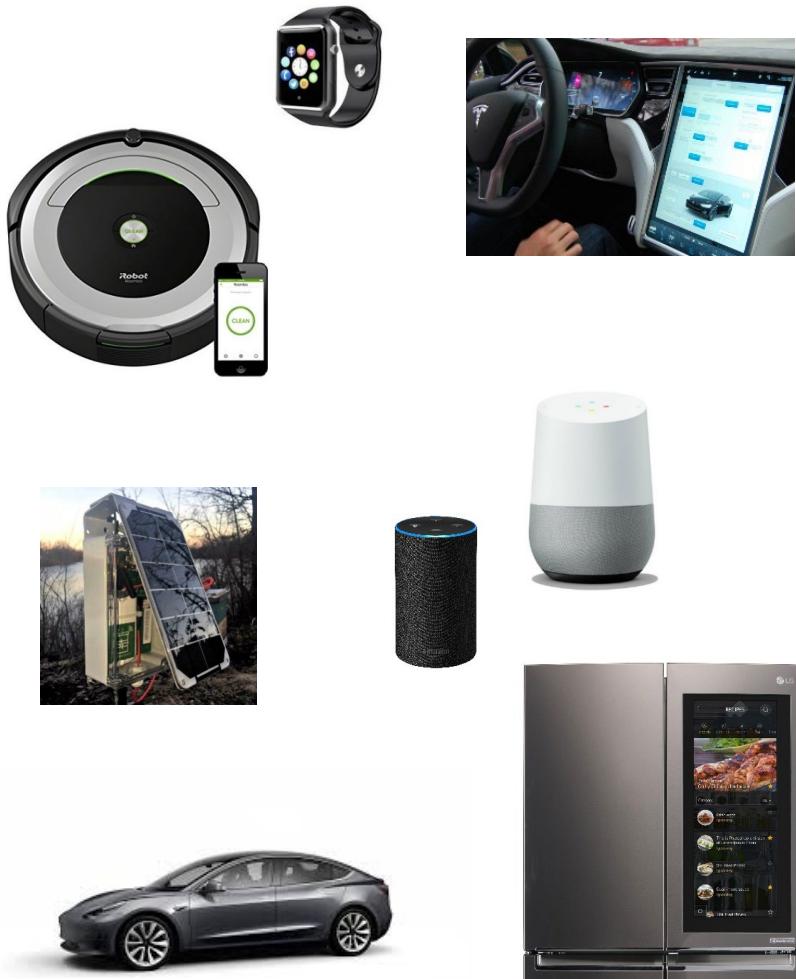
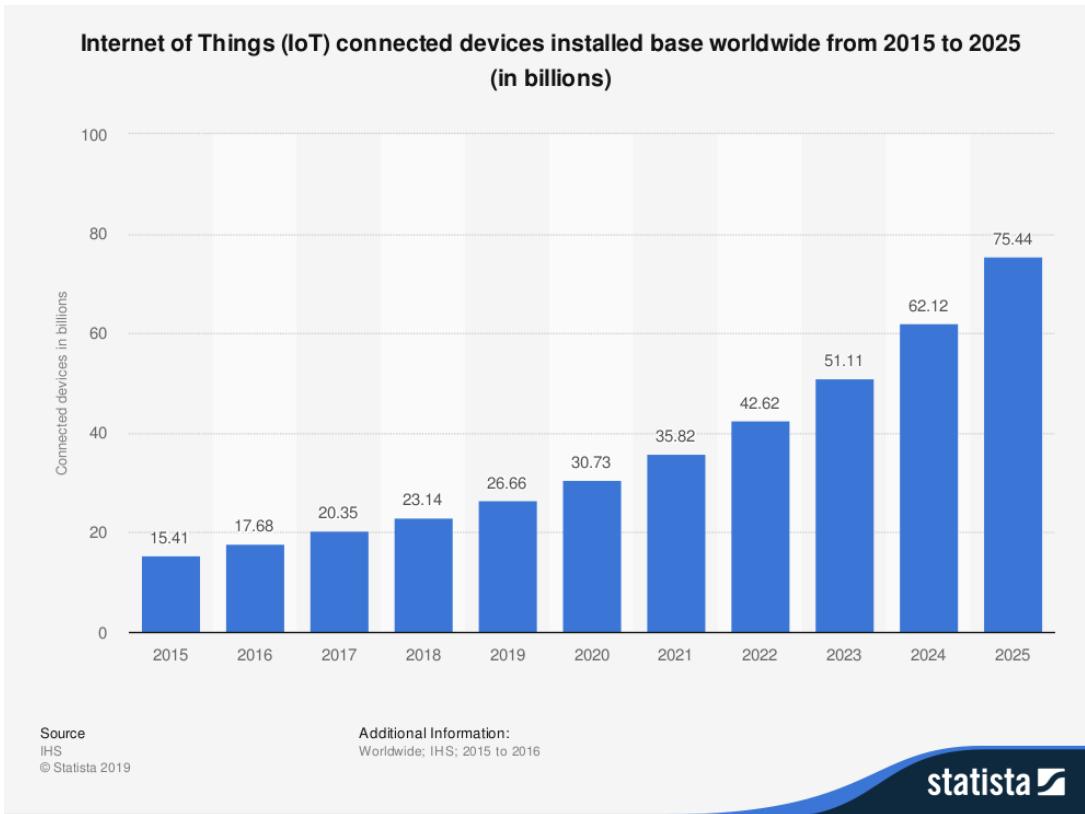


## Types of Cloud Deployment



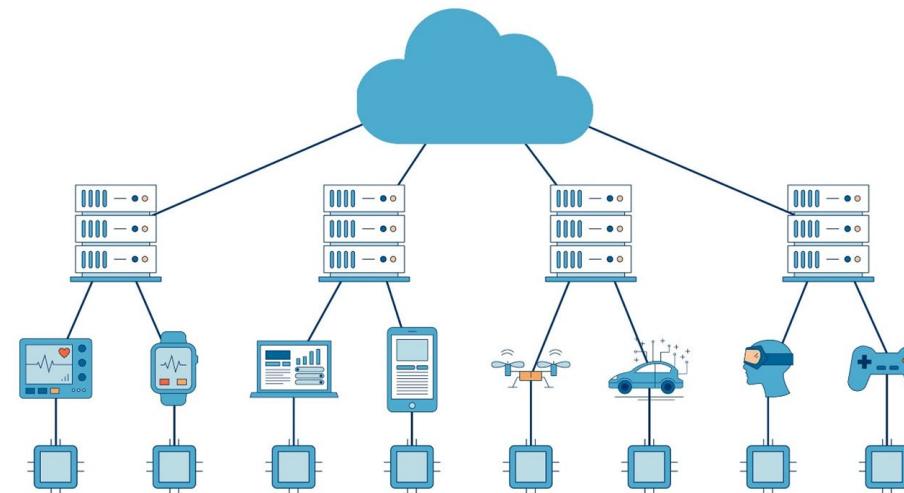
# Computation - Edge

# Computation - Edge computing



## Computation - Edge computing

Edge computing es un paradigma de computación distribuida que acerca la computación y el almacenamiento de datos a la ubicación donde se necesita, para disminuir los tiempos de respuesta, ahorrar ancho de banda y mantener la privacidad.



# Computation - Edge computing

Intel NCS 2



Price: \$79.99

Coral Edge TPU  
Accelerator



Price: \$74.99

Jetson Nano  
Nvidia



Price: \$99.00

Coral Edge  
TPU Board



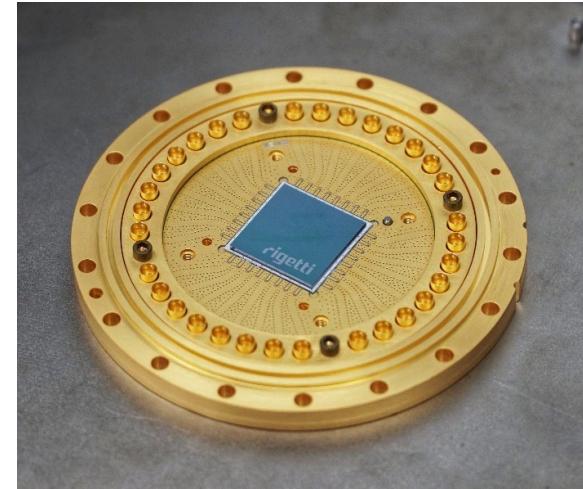
Price: \$140.99

Nuevos dispositivos baratos para ejecutar Machine Learning Inference en el perímetro

# Computación Cuántica

# Computation - Quantum computing

Las computadoras cuánticas son máquinas que utilizan las propiedades de la física cuántica para almacenar datos y realizar cálculos.



La unidad básica de memoria en las computadoras cuánticas es un bit cuántico o qubit.

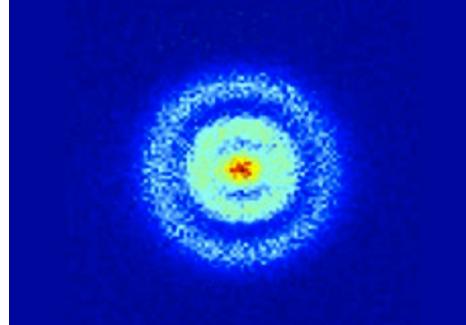


La computación cuántica utiliza los fenómenos cuánticos como la superposición y el entrelazamiento para realizar cálculos.

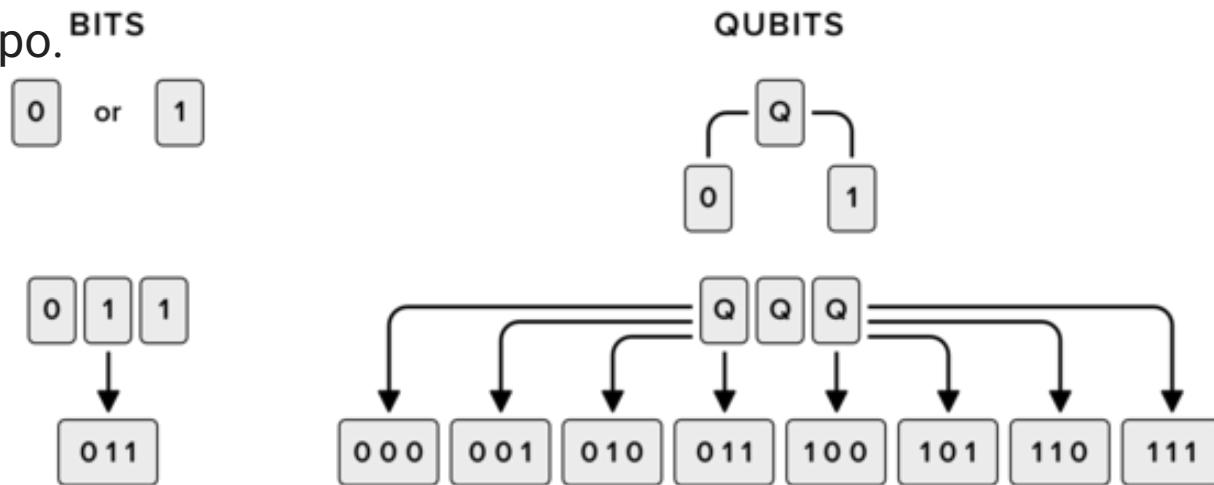
- La **superposición cuántica** es una propiedad de los sistemas cuánticos para estar en múltiples estados al mismo tiempo hasta que se mide.
- El **entrelazamiento cuántico** es un fenómeno físico que ocurre cuando un par o un grupo de partículas se genera, interactúa o comparte proximidad espacial de tal manera que el estado cuántico de cada partícula del par o grupo no puede describirse independientemente del estado de los demás, incluso cuando las partículas están separadas por una enorme distancia.

**Los qubits** se hacen usando sistemas físicos, como el espín de un electrón o la orientación de un fotón.

- Una computadora tradicional necesita tres bits para representar cualquier número entero entre 0 y 8
- Una computadora cuántica de tres qubits puede representar todos los números entre 0 y 8 al mismo tiempo.

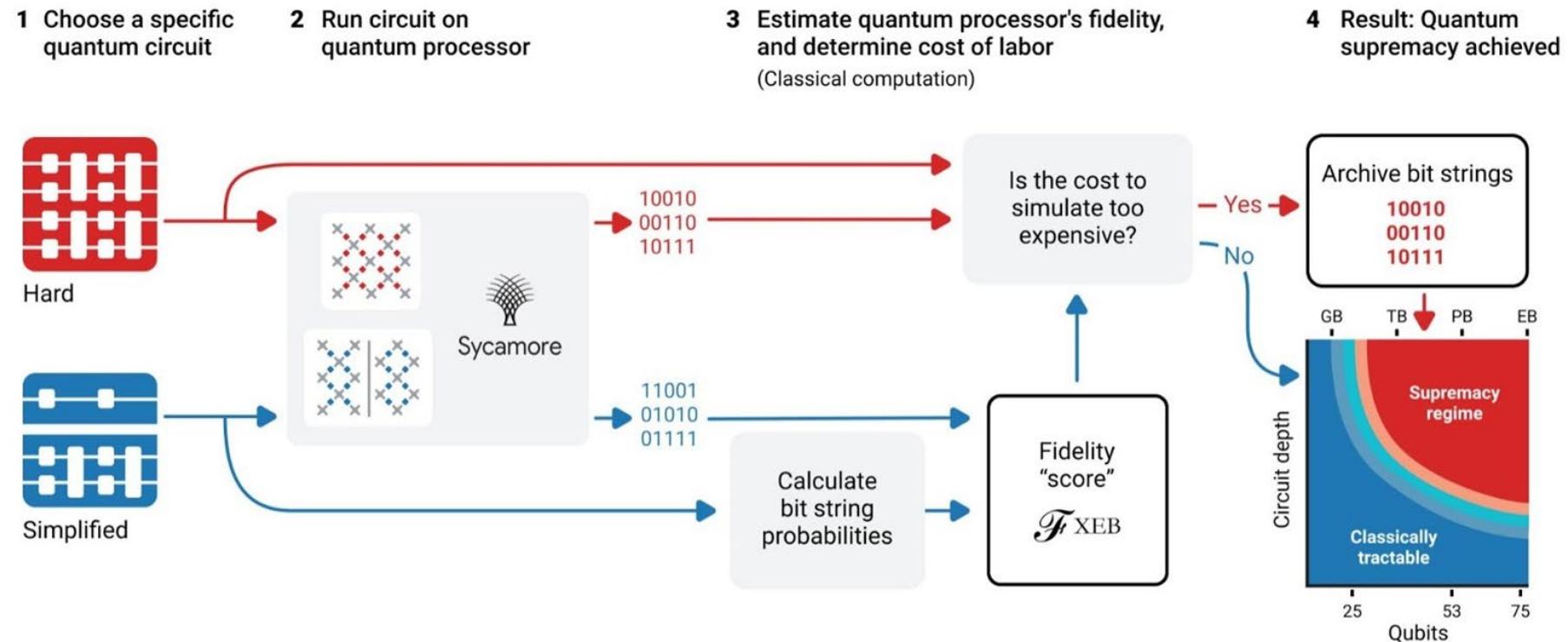


Hydrogen Atom



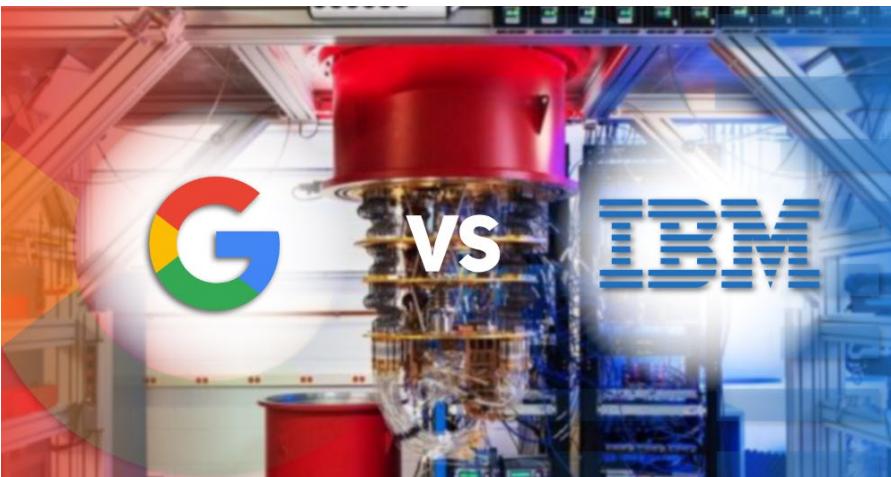
# Computation - Quantum computing

Las operaciones en qubits se realizan utilizando una combinación de multiplicación de matrices, números complejos y operadores de lógica cuántica en lugar de los operadores funcionales de verdad simples del álgebra booleana.



# Computation - Quantum computing

**La supremacía cuántica o ventaja cuántica** es el objetivo de demostrar que un dispositivo cuántico programable (computadora cuántica) puede resolver un problema que ninguna computadora clásica puede resolver en un tiempo factible.



Quantum Supremacy: A Test on the IBM Quantum Computer

Yash Palan,<sup>1,\*</sup> Bikash K. Behera,<sup>2,3,†</sup> and Prasanta K. Panigrahi<sup>3,‡</sup>

<sup>1</sup>Department of Physics, Indian Institute of Science Education and Research Bhopal, Bhopal, Madhya Pradesh, India  
<sup>2</sup>Rikash's Quantum (OPC) Pvt. Ltd., Rajindri, Mohanpur 241246, Nadia, West Bengal, India

<sup>a</sup>Bikash's Quantum (OPC) Pet. Ltd., Balindi, Mohanpur 741246, Nadia, West Bengal, India  
<sup>b</sup>Department of Physical Sciences,  
University of Calcutta, 32, College Street, Kolkata, 700073, West Bengal, India

*Indian Institute of Science Education and Research Kolkata, Mohanpur 741246, West Bengal, India*

The news of achieving quantum supremacy by Google AI has received critical acclaim by a number of researchers in the field of quantum computing. Here, we implement

cross entropy bench marking procedure on the IBM quantum computer and report the results obtained. The backend used for this purpose is IBM Qurene. Through this experiment, we observe an exponential decay in the fidelity. Noticing that the observations are similar to ones obtained by Google AI, we conclude that by increasing the number of qubits, it is possible to achieve quantum supremacy on IBM's quantum computer.

## 1. INTRODUCTION

Ever since Deutsch's quantum algorithm was made public, the idea of quantum computation was made more plausible. This sparked a great interest in the possibility of quantum computation, which has led to many developments in the field. One such development is the search for quantum algorithms that can be used to solve practical problems like factoring large numbers or solving systems of linear equations. Another development is the search for quantum error correction codes that can be used to protect quantum information from decoherence and noise. The field of quantum computation has also led to the development of new mathematical tools and techniques, such as quantum probability theory and quantum logic, which have applications in various fields of science and technology. In addition, the search for quantum algorithms has led to the development of new quantum computing hardware, such as quantum computers based on superconducting qubits, optical qubits, and ion traps. These developments have opened up new possibilities for quantum computation and have paved the way for future breakthroughs in the field.

Many approaches have been undertaken for the creation of quantum computers. One approach is to use standard quantum computation using trapped cold atoms in a periodic potential.<sup>1</sup> The advantage of such a system is that it is relatively simple to implement. Another proposal is using optics for creation of a quantum computer.<sup>2</sup> Other possibilities include using molecular quantum computation,<sup>3</sup> or biological computation<sup>4</sup>, quantum computation using quantum optics<sup>5,6,7</sup> and many others. The present day quantum computer is based on optical components with as much as 53 qubits.<sup>8</sup> However, a large part of the development of physical realizations is in the area of quantum optics.

The birth of the idea of quantum supremacy lies in the work of Shor [1] who showed that a quantum computer can solve the problem of Prime Factorization and Discrete Logarithms in a Quantum Computer.<sup>1</sup> In this paper, we outlined the basic idea of quantum computation and quantum supremacy. Here, we see one of the first instances where a quantum computer is able to provide a speedup when compared to a classical algorithm. We also show that an algorithm is able to provide an exponential speedup when compared to the best classical algorithm. On a best case scenario, a quantum computer can cross entropy in 2 steps a square root speedup in comparison to the best classical algorithm. This is a significant cross entropy in the field of quantum computation. This is done by calculation of the fidelity and plotting its variation with respect to time and the number of qubits. In addition, we refer to the time required for a circuit to repeat. Using this we verify that the results obtained by Google and IBM are correct. Hence, it is believed that for 53 qubits or higher, quantum supremacy can be achieved even by the IBM quantum computer.

we arrive at the equation for cross entropy metric which is used for benchmarking purpose. In section 3,

# Fundamentos de Ingeniería Informática

## Module II - Unit 8 Nuevas tendencias II

Profesor: Héctor Molina García



Universidad  
Francisco de Vitoria  
UFV Madrid

*Grado en Ingeniería Informática  
Escuela Politécnica Superior*

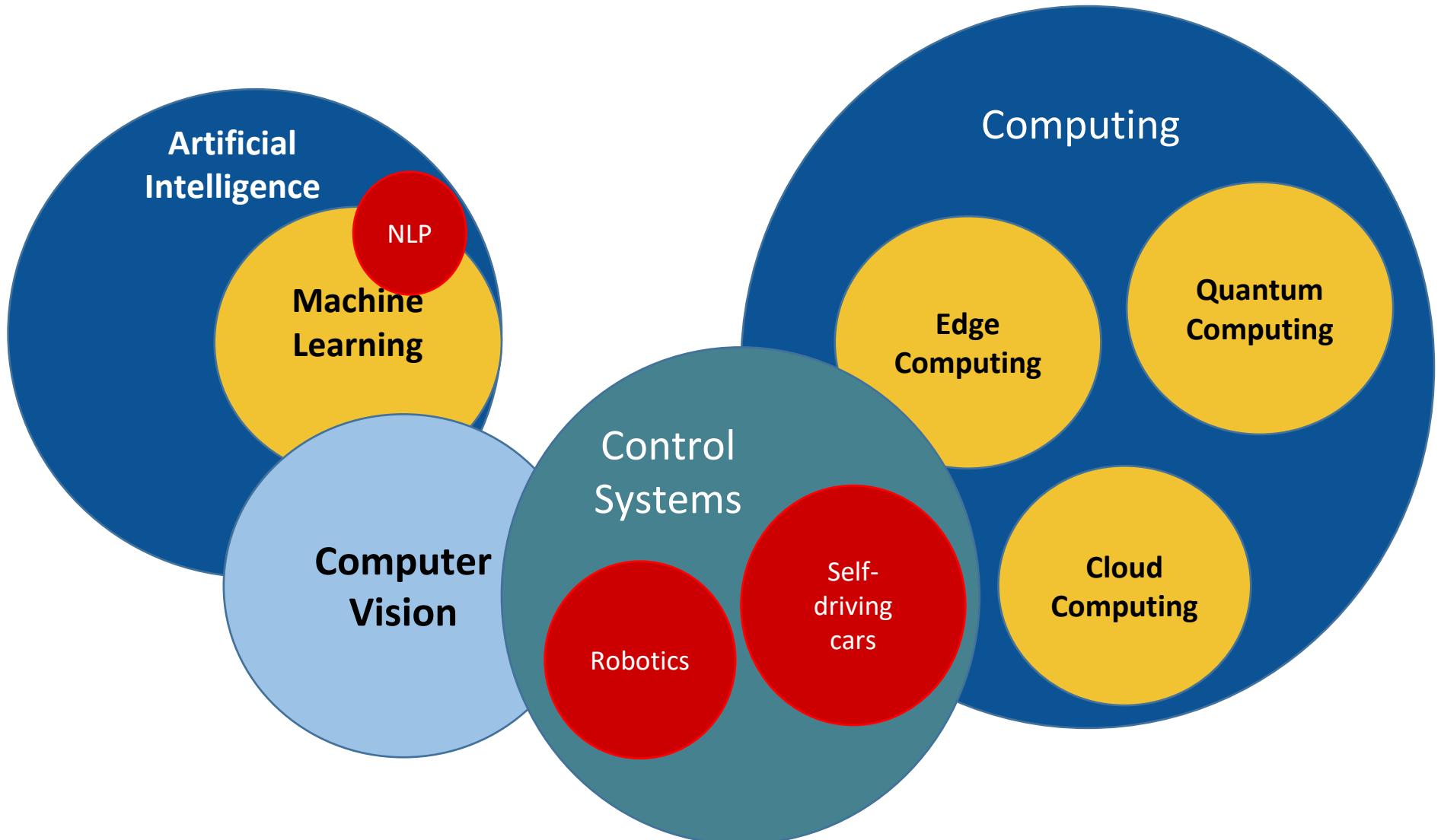
# Trend technologies

Una tendencia es un cambio o desarrollo hacia algo nuevo o diferente.  
Esto significa que la tecnología de tendencia cambiará nuestro futuro.

- Inteligencia Artificial (IA).
- Aprendizaje automático (ML).
- Visión Artificial.
- Computación.
- **Sistemas de control.**
- **Internet de las Cosas (IoT).**
- **Blockchain.**
- **5G**

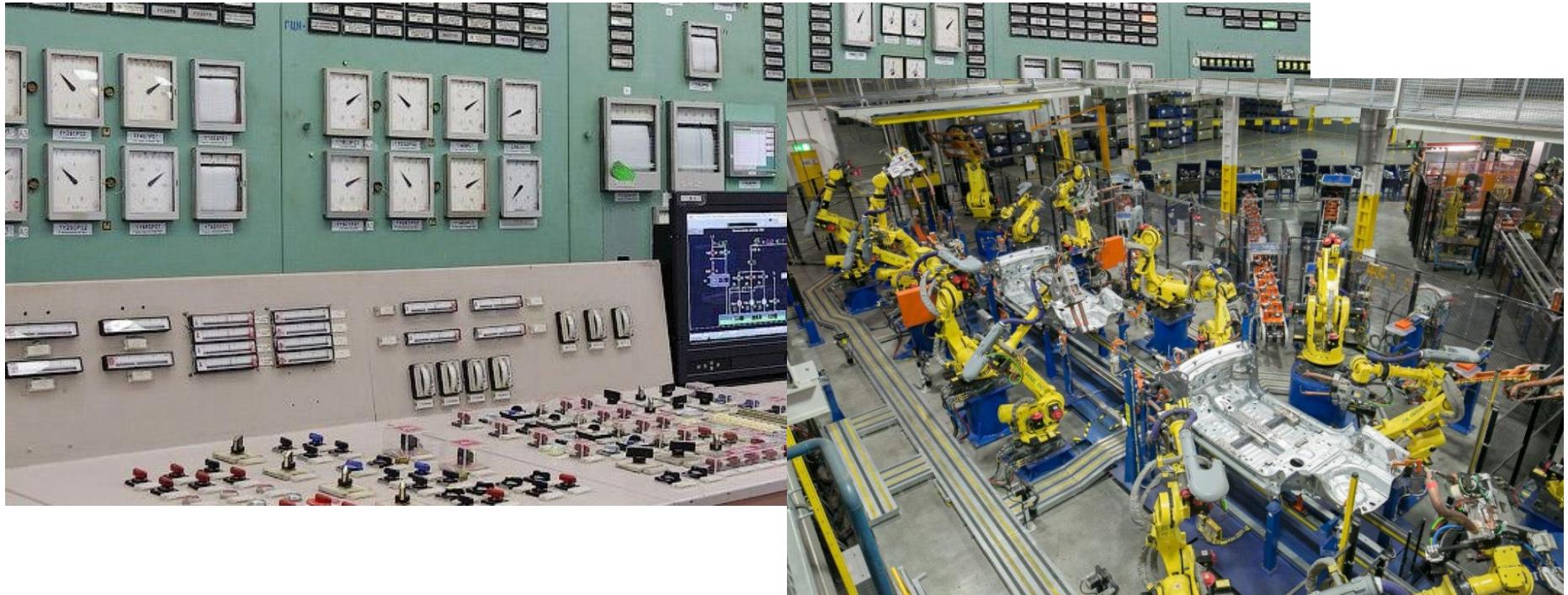


# Sistema de control



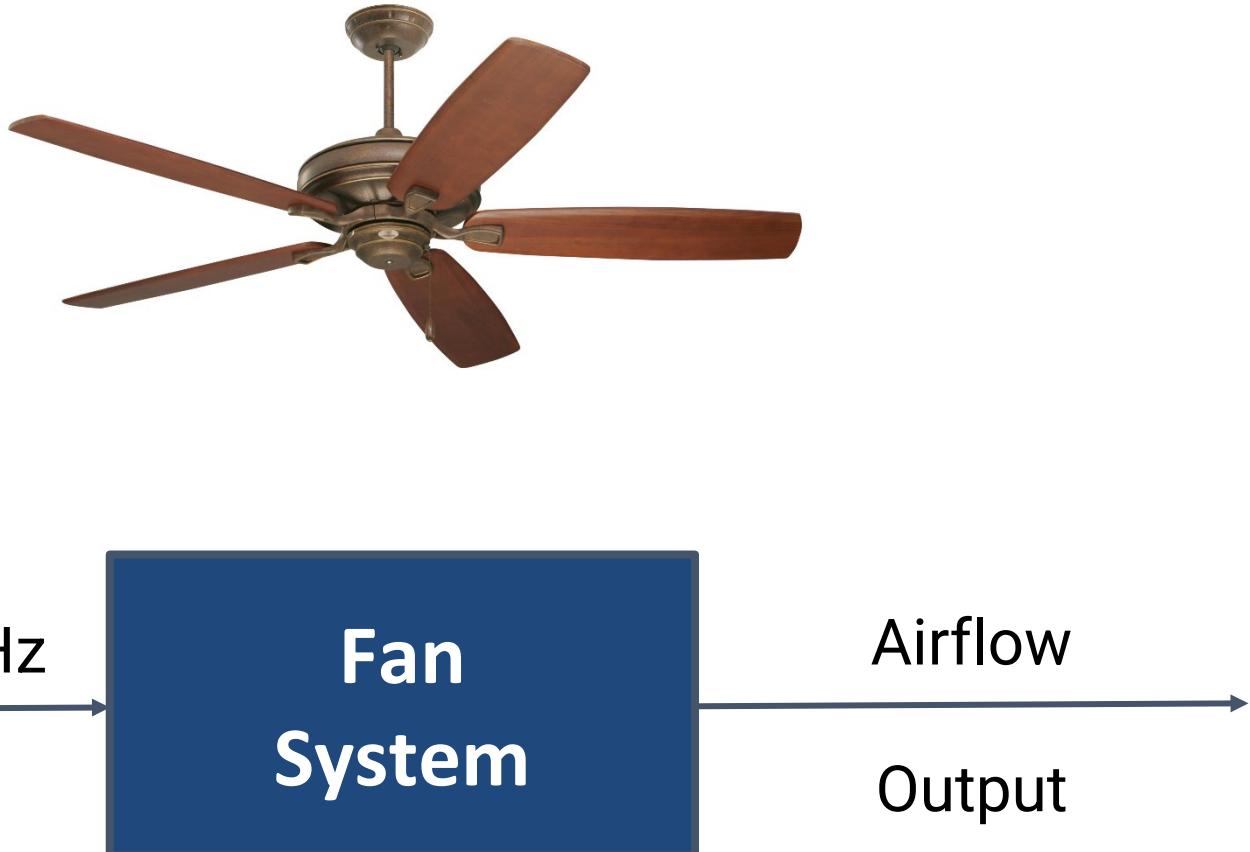
# Control Systems - Robotics

Control: Es el proceso para regular u ordenar un sistema con el fin de alcanzar un objetivo deseado.

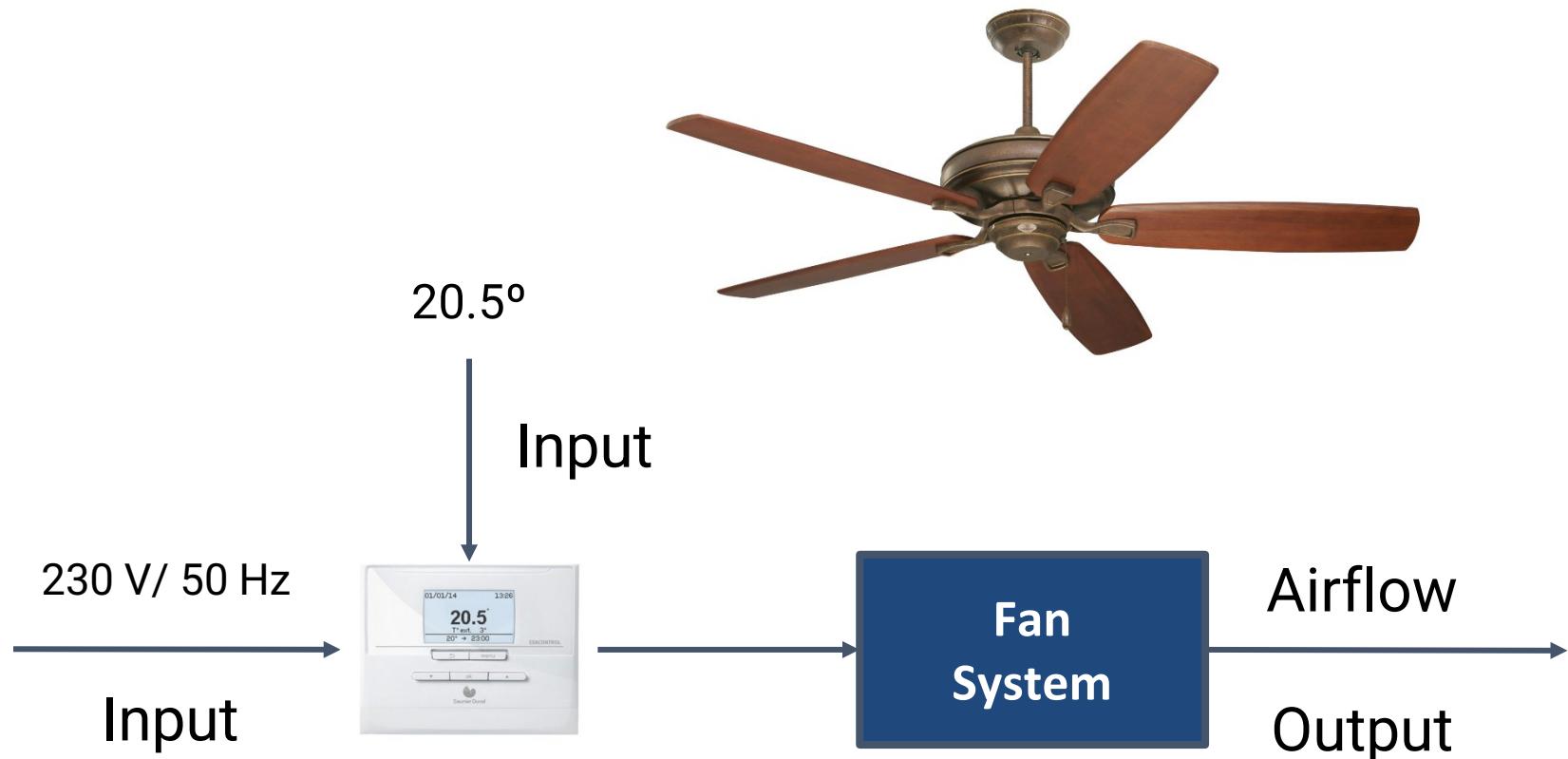


Citroën C3 production line

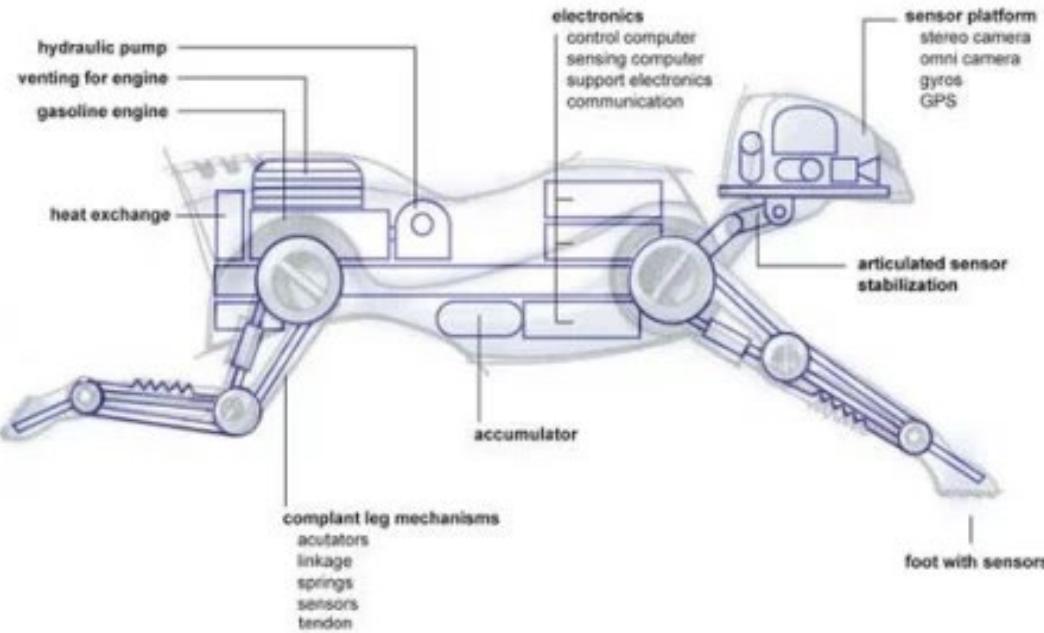
# Control Systems - Robotics



# Control Systems - Robotics



# Control Systems - Robotics



© 2003 Boston Dynamics

Diseño inicial del perro robótico (Boston Dynamics)



# Control Systems - Robotics

Un sistema de control es un sistema o subsistema que está compuesto por un conjunto de componentes (sensores, actuadores y/o acondicionador) que regulan el comportamiento de un sistema (o de ellos mismos) para lograr un objetivo específico.



# Control Systems - Robotics

Un sensor es un dispositivo capaz de capturar información del entorno mediante el uso de un transductor.

- **Sensores propioceptivos:** Son sensores que miden los valores internos del sistema. Por ejemplo, velocidad del motor, carga de la rueda, ángulos de articulación del brazo del robot, voltaje de la batería.

- **Sensores exteroceptivos:** Son sensores que miden la información del entorno. Por ejemplo: distancia (GPS), proximidad (sonar, láser, etc.), actile (parachoques, presión, etc), visual (cámaras, etc.) y acústica (micrófonos, etc.).



# Control Systems - Robotics

Un **actuador** es un dispositivo capaz de transformar algún tipo de energía mediante el uso de un transductor con el fin de producir otro tipo. Los actuadores se utilizan para producir variaciones en el medio ambiente

- Locomoción (Motores).
- Servomotor.
- Visual (Pantallas, leds)
- Acústico (altavoces)



# Control Systems - Robotics

Un robot humanoide es un robot que se asemeja al cuerpo humano en forma.



NASA's R5 robot

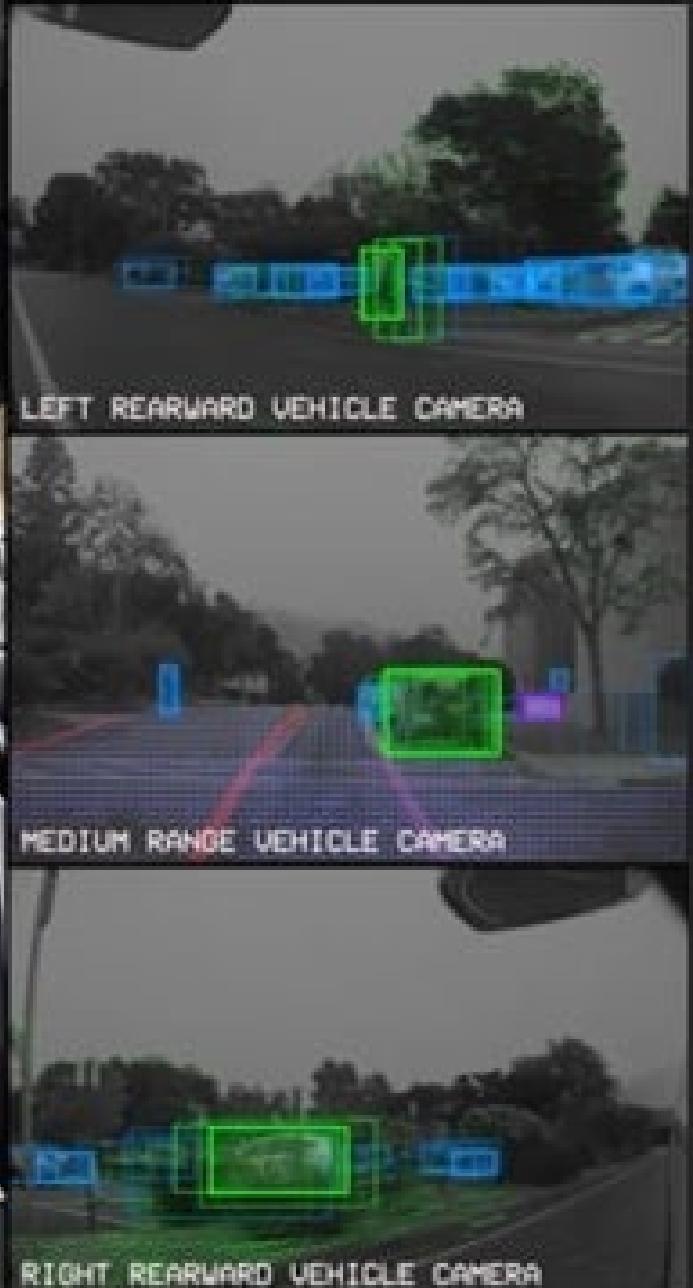


Spot and Atlas (Boston Dynamics)

# Control Systems - Robotics



# Self-driving cars



Universidad  
Francisco de Vitoria  
UFV Madrid

# ¿Esto es un coche autónomo?

# Control Systems - Self-driving cars

Los sistemas de control en los coches autónomos pueden ser:

- **Driver-in-Control.** El vehículo tiene sofisticadas funciones semiautomáticas para la asistencia al conductor, pero el conductor sigue siendo el responsable final y tiene el control.
- **Vehicle-in-Control.** Cuando el vehículo es totalmente autónomo y no requiera conductor. Es probable que el vehículo esté restringido a ciertos lugares y condiciones.

# Control Systems - Self-driving cars

## SAE Levels of Driving Automation Simplified - Two Matter Most

Level 2: Driver-in-Control and Level 4: Vehicle-in-Control

AUTOPilot REVIEW



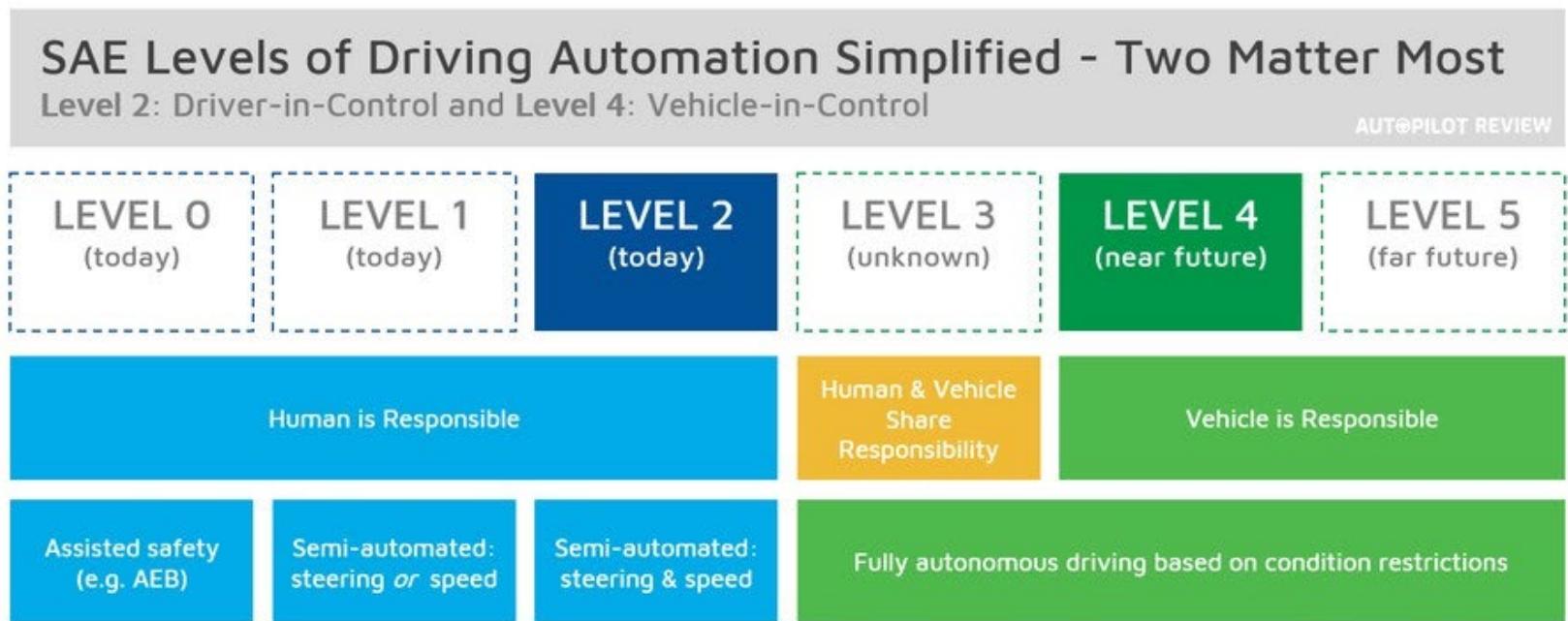
# Control Systems - Self-driving cars

El modo de conducción define quién es responsable del comportamiento del sistema de control.



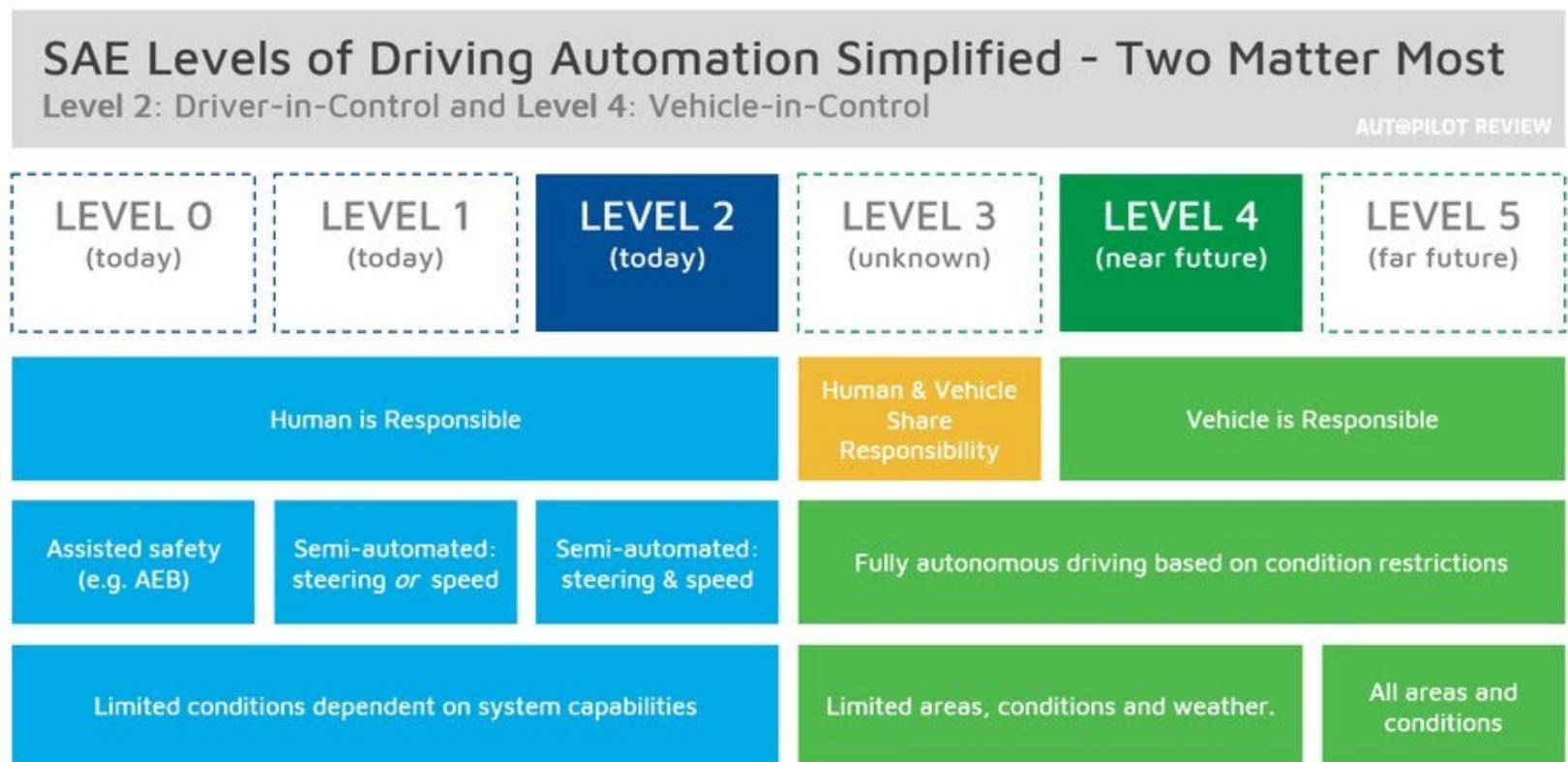
# Control Systems - Self-driving cars

La automatización define cuáles son las características del sistema de control.



# Control Systems - Self-driving cars

Las condiciones definen la configuración del entorno donde funciona el sistema de control autónomo.

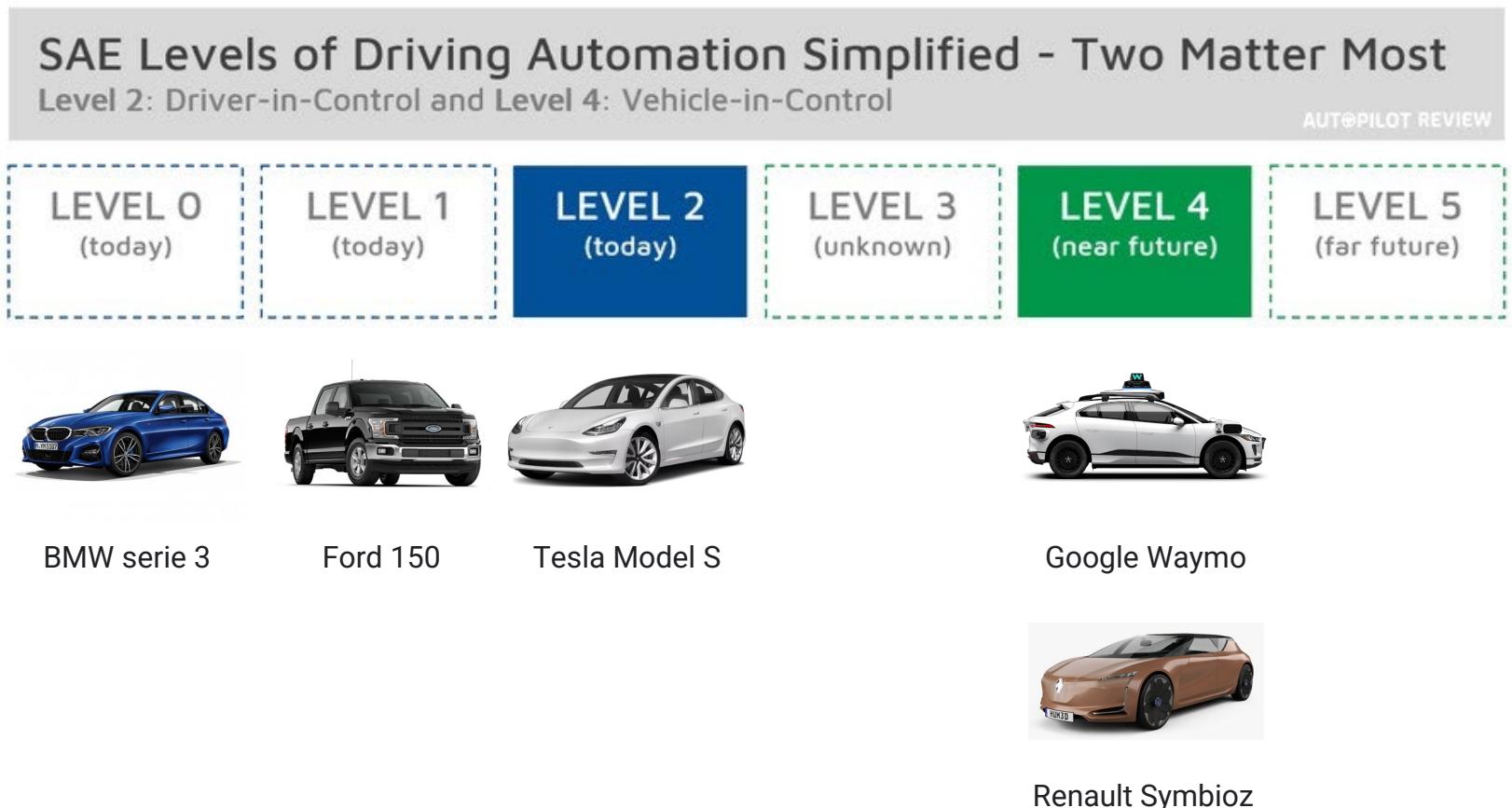


# Control Systems - Self-driving cars



# Control Systems - Self-driving cars

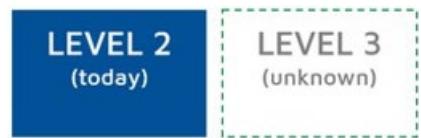
¿Qué niveles están disponibles ahora?



# ¿Coches autónomos o coches Self driving cars?

# Control Systems - Self-driving cars

Un **automóvil autónomo** es un vehículo capaz de detectar su entorno y operar sin participación humana. No se requiere que un pasajero humano tome el control del vehículo en ningún momento, ni se requiere que un pasajero humano esté presente en el vehículo.

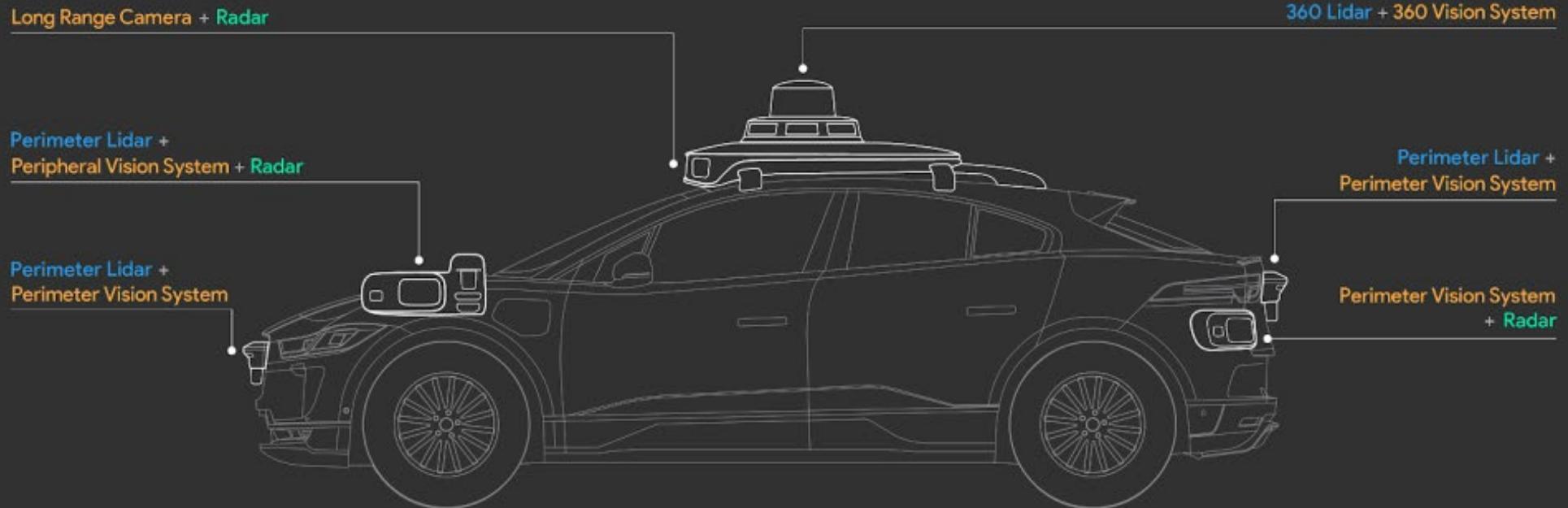


Un **self-driving car** es un vehículo capaz de detectar su entorno y operar en él, pero los conductores humanos deben tomar el control en situaciones peligrosas.



# Google self-driving car

# Control Systems - Self-driving cars



# Control Systems - Self-driving cars

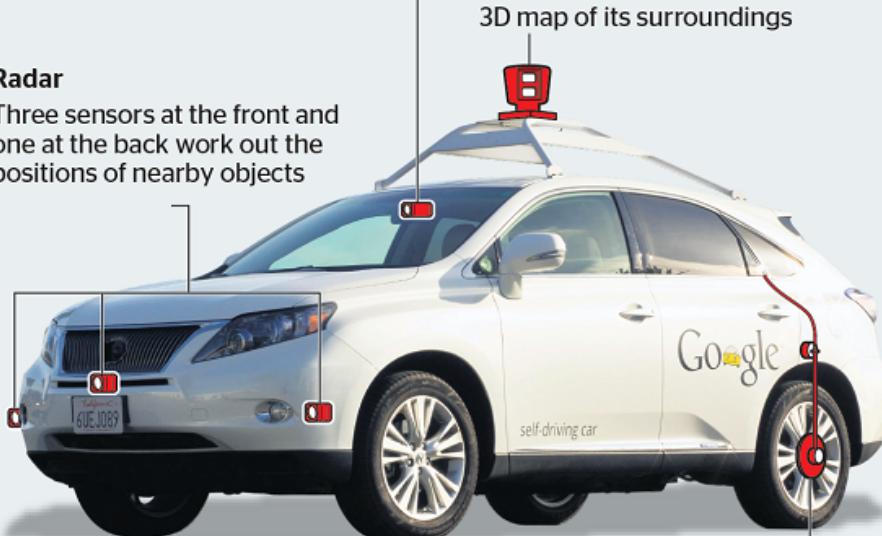


## Look – no driver

**Video camera**  
Detects traffic lights, oncoming vehicles and other obstacles

**Radar**  
Three sensors at the front and one at the back work out the positions of nearby objects

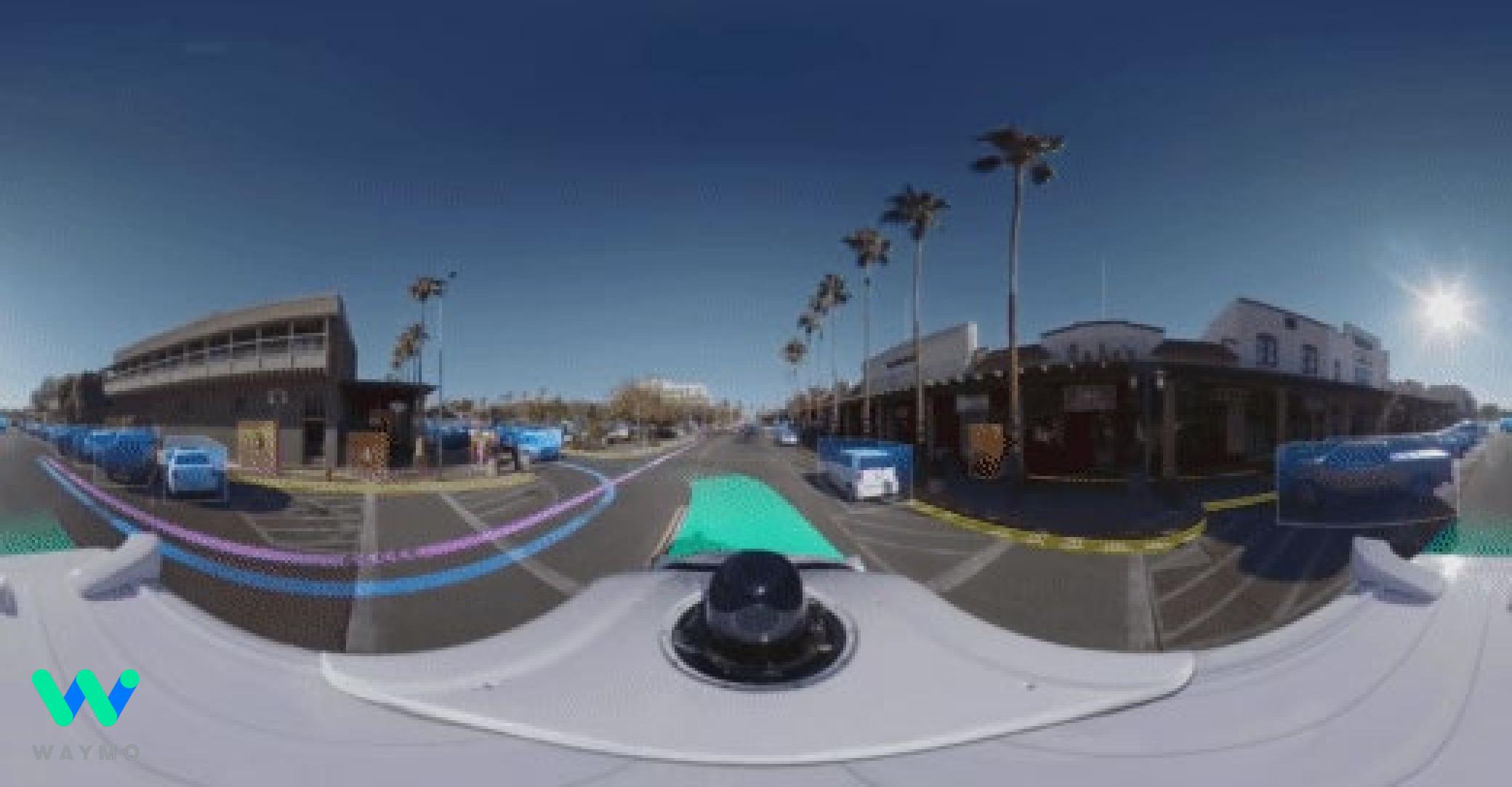
**Lidar**  
A rotating sensor on the roof scans 200ft in all directions to create a 3D map of its surroundings



**Position estimator**  
A sensor on the left rear wheel measures the car's movements so that its position can be mapped with accuracy

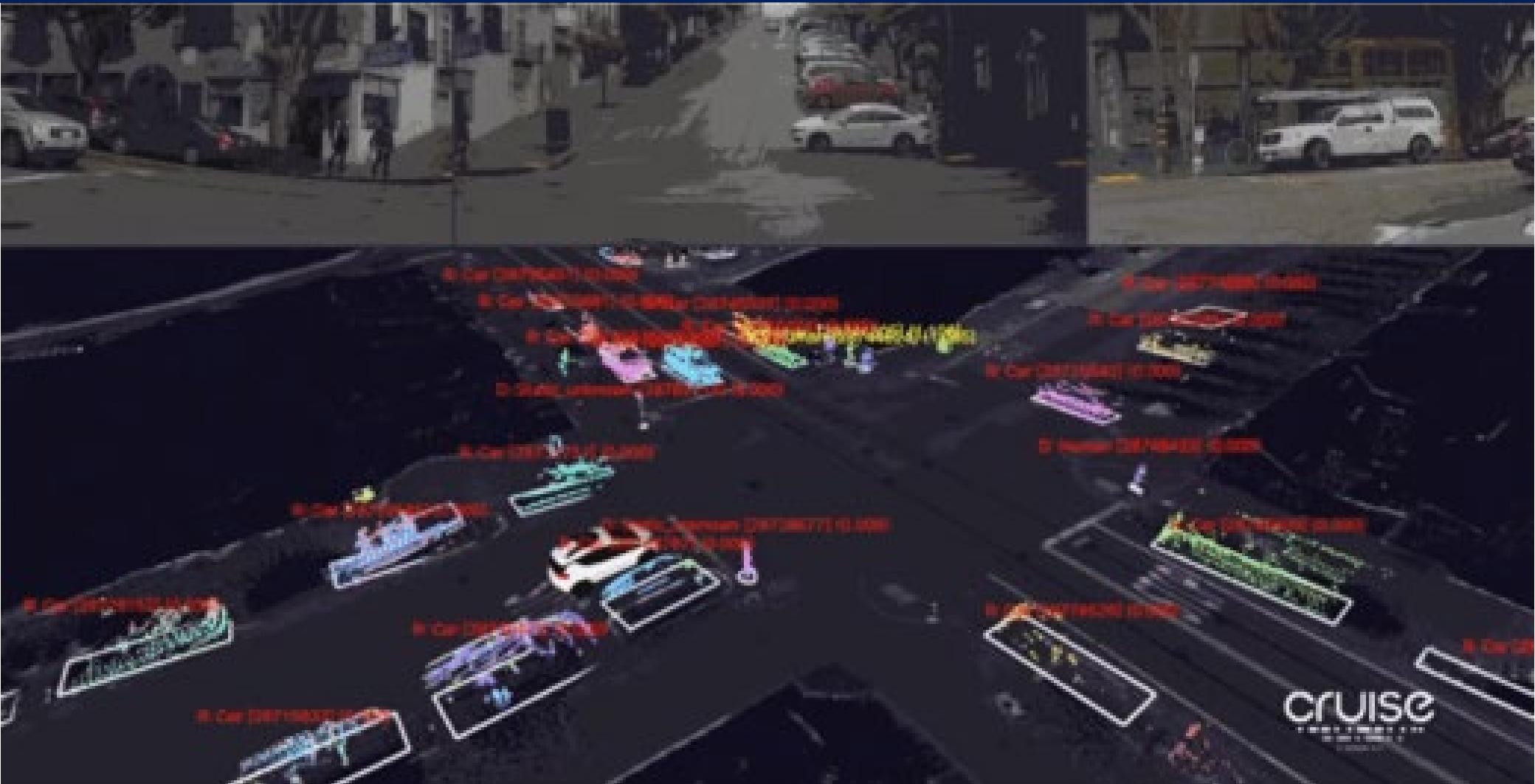
- El sistema lidar se encuentra encima del automóvil, siempre girando como R2D2, atrapando objetos y ayudando a la IA a descubrir dónde están las cosas.
- Los sensores de radar alrededor del automóvil detectan objetos para que el automóvil pueda evitarlos.
- Cámara de vídeo a nivel humano para detectar semáforos, señales u objetos.
- Estimadores de posición para medir los movimientos del automóvil.

# Control Systems - Robotics



Universidad  
Francisco de Vitoria  
**UFV** Madrid

# Control Systems - Robotics



# Control Systems - Self-driving cars

El botón e-stop: Este es un stop de pánico y permite detener el sistema de conducción autónoma.

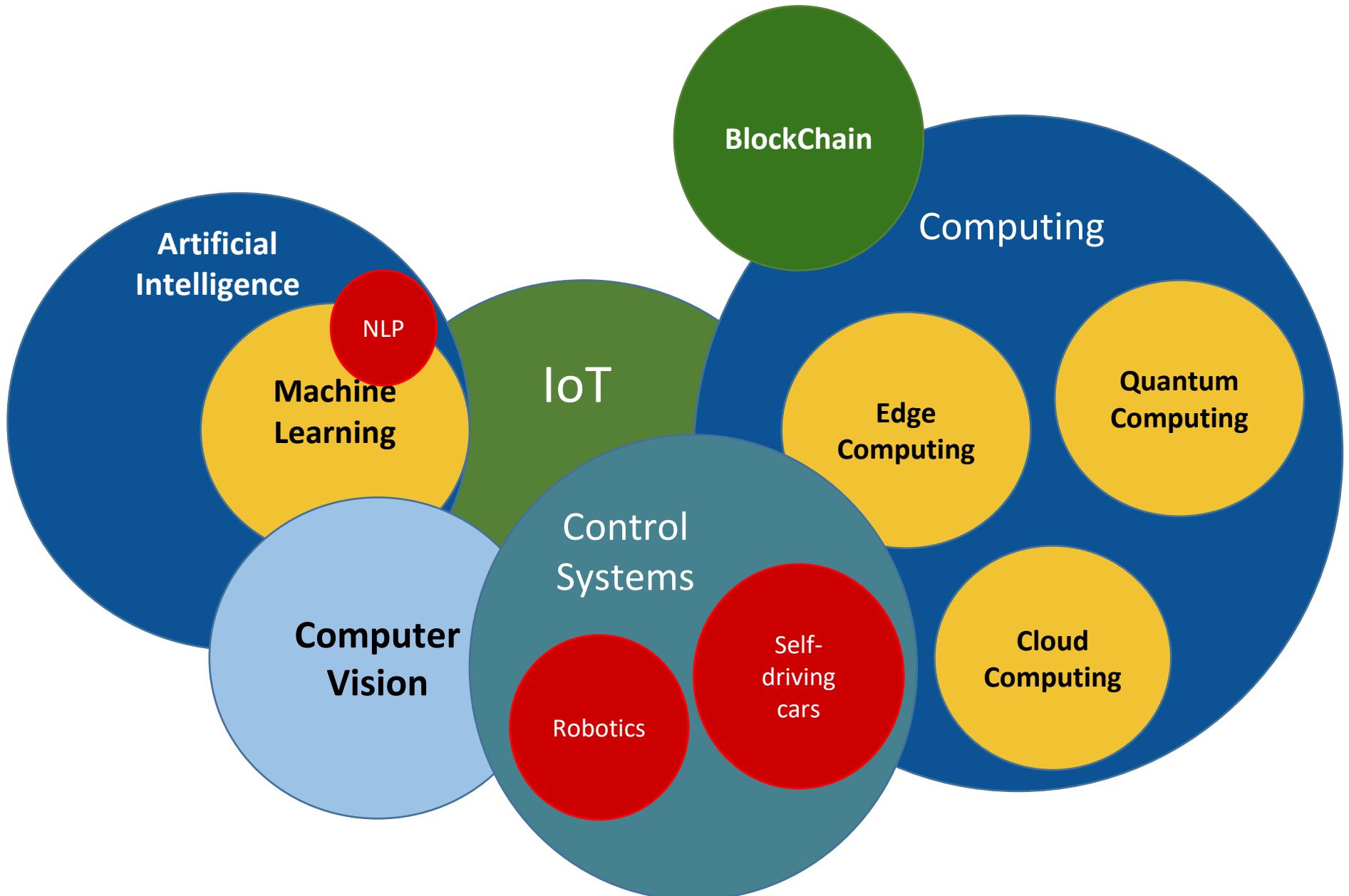


# Control Systems - Self-driving cars

Otto era una startup de San Francisco que fabrica kits de software y hardware para vehículos autónomos. Fue adquirida por Uber.



# Internet of Things



# IoT - Internet of Things

El Internet de las cosas (IoT) describe la red de objetos físicos (cosas) que están integrados con sensores, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet.



# IoT - Internet of Things

Las tecnologías IoT se dividen en dos tipos:

- **Consumer IoT (CIoT)** se centra en la comodidad para clientes individuales, mientras que IIoT se centra fuertemente en el sector industrial, mejorando la eficiencia y la seguridad.
- **Industrial IoT (IIoT)** se centra en ofrecer servicios a organizaciones (smart factory) o entre organizaciones (cadena de suministro minorista) para mejorar la eficiencia, las operaciones, ahorrar energía o incluir IA a los dispositivos. IIoT es el tipo de IoT más establecido y maduro.

# IoT - Internet of Things

**Consumer IoT (CIoT)** se utiliza para crear aplicaciones como:

- Los gadgets son dispositivos de consumo, por ejemplo, móviles, refrigeradores, gafas, etc.
- La cantidad de datos y las tasas son comparativamente bajas.
- Las aplicaciones no son muy críticas.
- Las aplicaciones CiOT se consideran centradas en el consumidor.



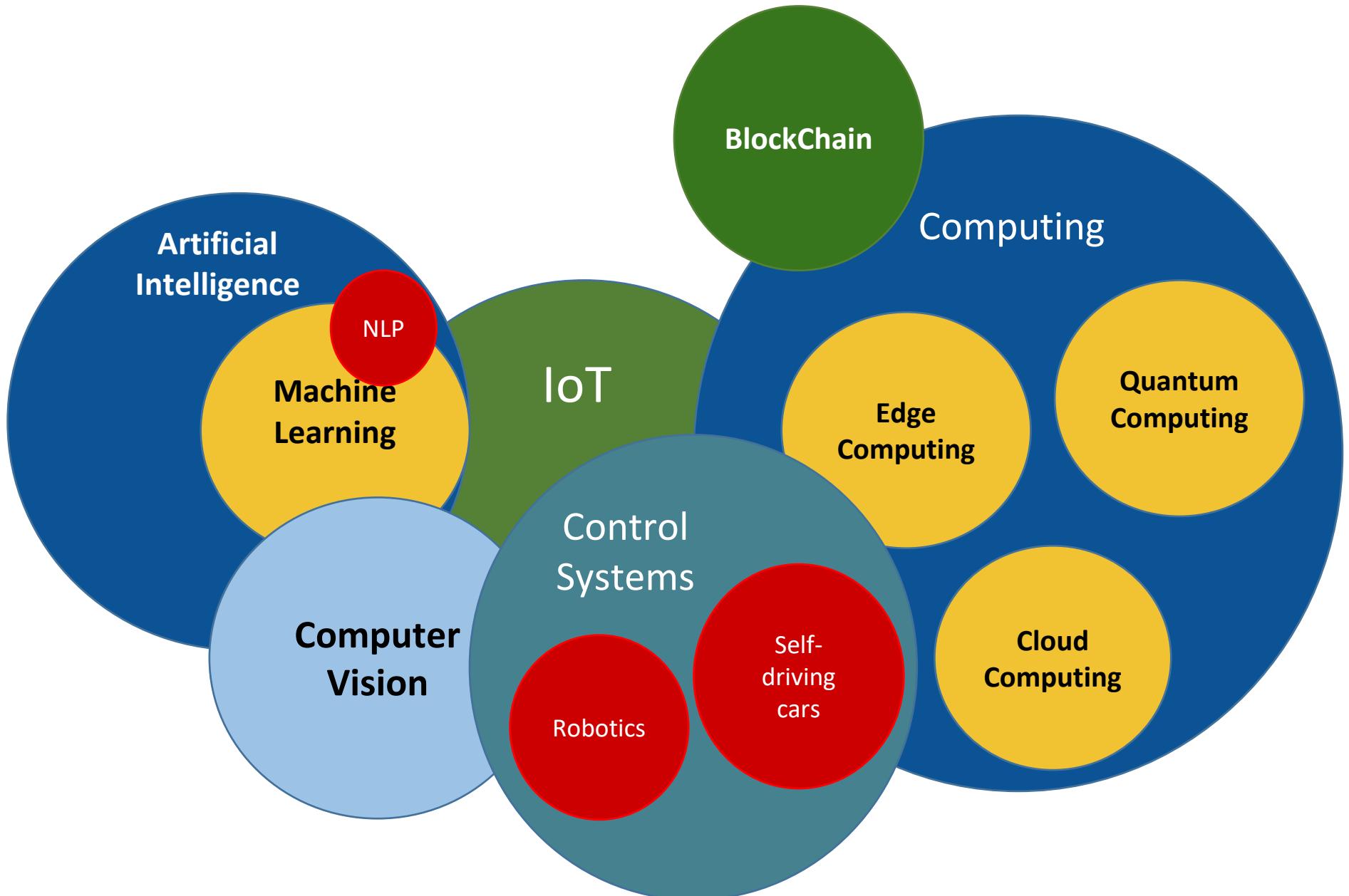
# IoT - Internet of Things

**Industrial IoT (IIoT)** se utiliza para crear aplicaciones como:

- Los dispositivos IIoT funcionan en entornos industriales, de transporte, energéticos o médicos.
- La cantidad y las tasas de datos tienden a variar de sostenidas a relativamente altas.
- Las aplicaciones son críticas para la seguridad, y se consideran centradas en el sistema.



# Blockchain



# Blockchain

**Blockchain** o cadena de bloques es una **base de datos distribuida**, formada por cadenas de bloques que **evitan su modificación**. Cuando se publica, se genera un sello de tiempo y se enlaza con el bloque anterior.

Se podría decir que blockchain es un libro de cuentas de una enorme BBDD donde se registran todas las transacciones y cuenta con el consenso de todo el mundo.



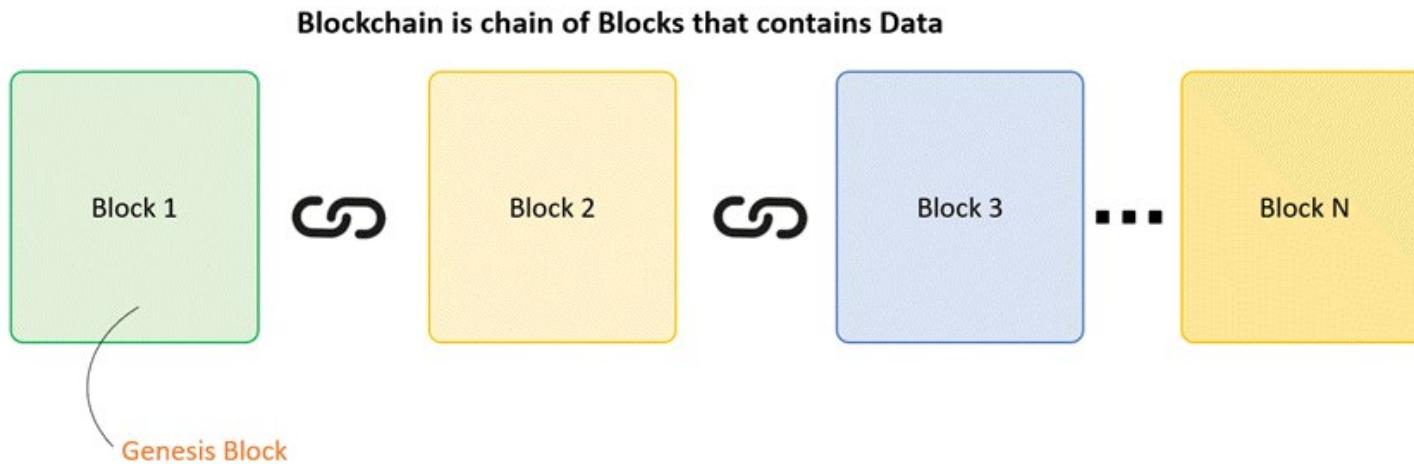
## A really important thing



- Blockchain no es Bitcoin, pero es la tecnología detrás de Bitcoin.
- Bitcoin es el token digital y blockchain es el registro para realizar un seguimiento de quién posee los tokens digitales.
- No puedes tener Bitcoin sin blockchain, pero puedes tener blockchain sin Bitcoin.

# Blockchain

**Blockchain** es una cadena de bloques que contienen información. Los datos almacenados dentro de un bloque dependen del tipo de blockchain.



- El primer bloque de la cadena se llama bloque Génesis.
- Cada nuevo bloque de la cadena está vinculado al bloque anterior.

# Blockchain

From: Sam

To: Peter

Amount: 200



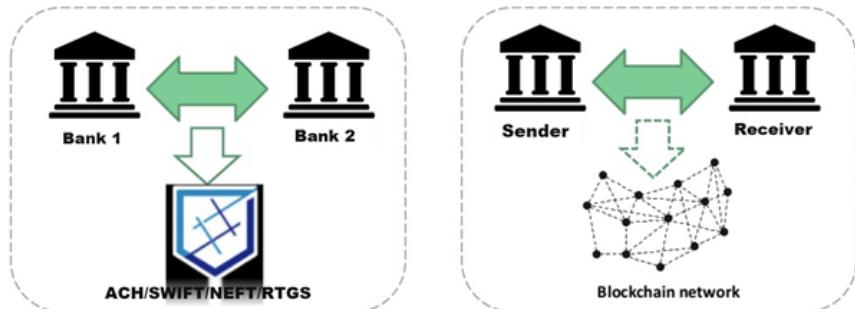
Por ejemplo, un bloque de Bitcoin contiene información sobre:

- Remitente (De)
- Receptor (Para)
- Número de bitcoins a transferir (Cantidad)

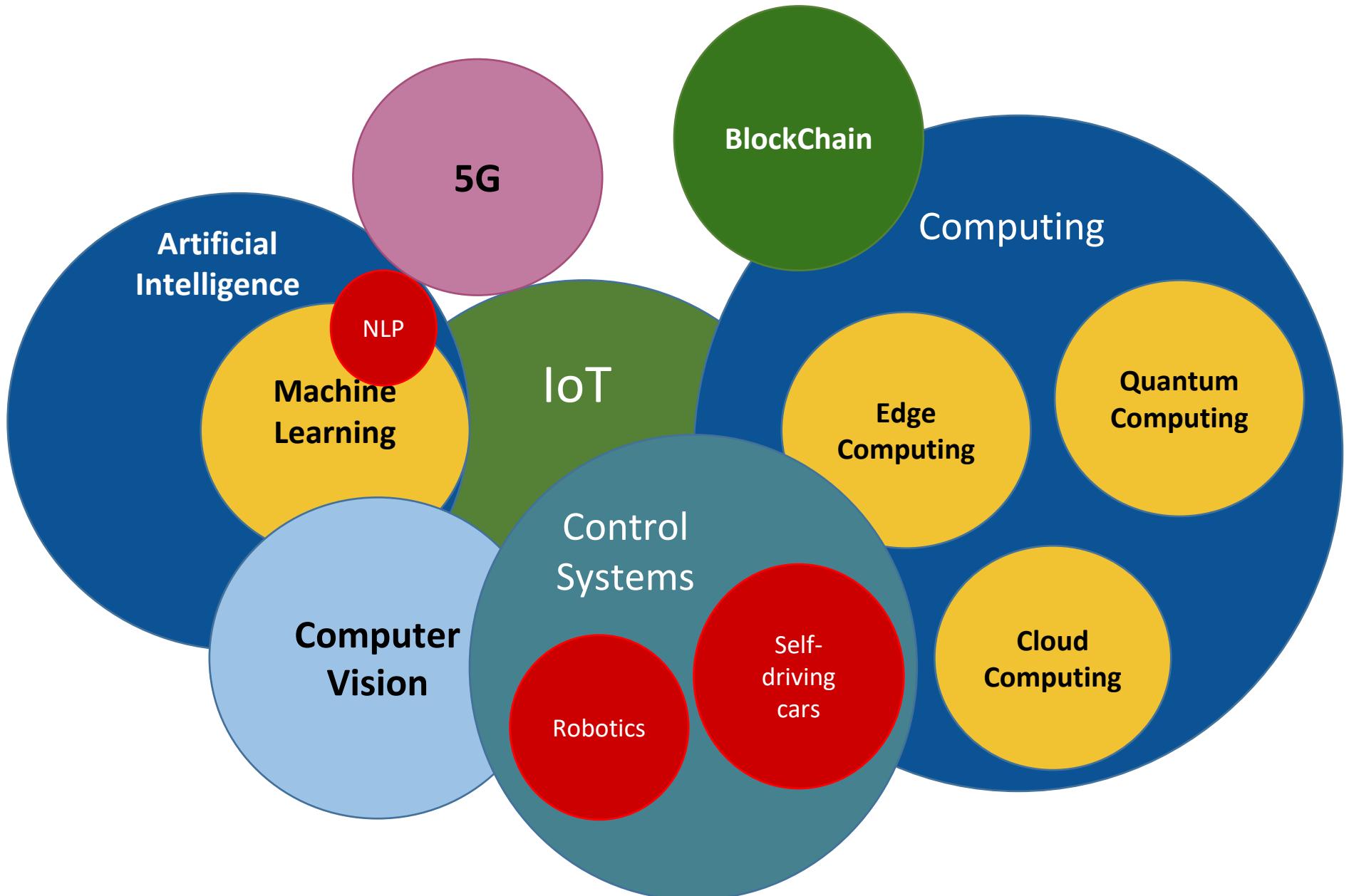
## Bitcoin Block Example

# Blockchain

Una **criptomoneda** es un medio de intercambio como las monedas tradicionales como el EURO, pero está diseñada para intercambiar la información digital a través de un proceso posible gracias a ciertos principios de criptografía. Una criptomoneda es una moneda digital y se clasifica como un subconjunto de monedas alternativas y monedas virtuales.



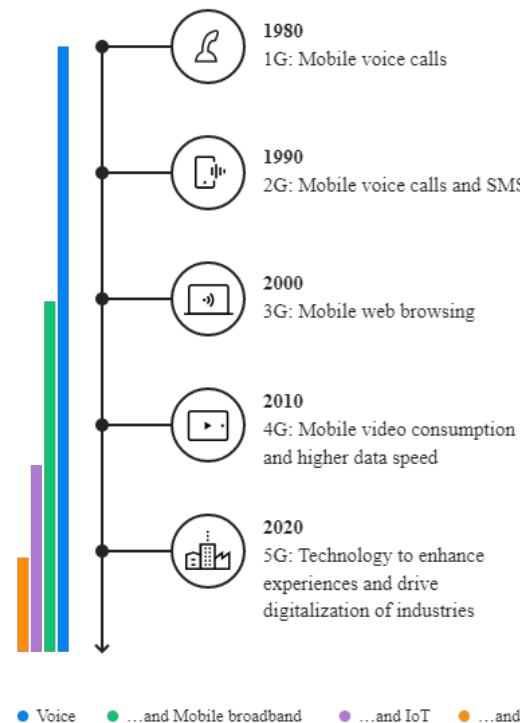
# 5G



# 5G

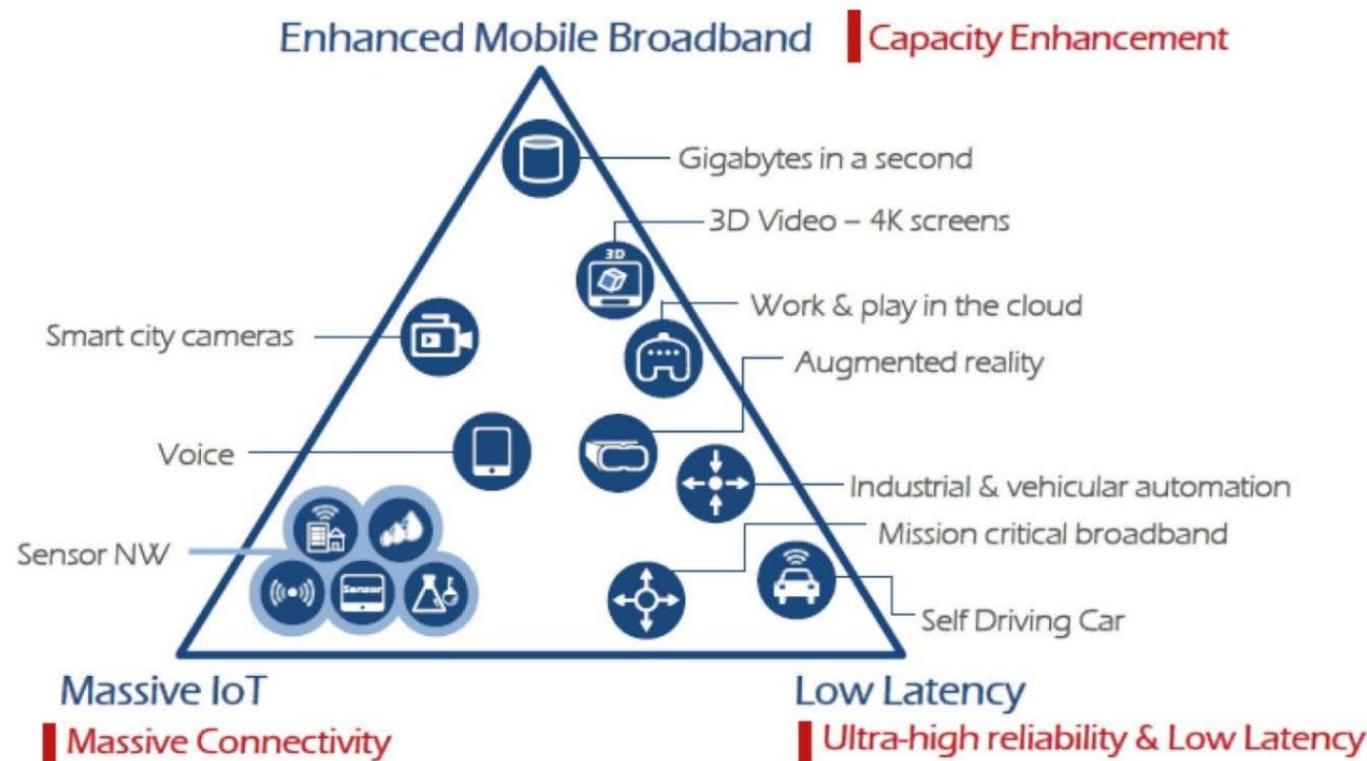
5G es la quinta generación de redes telefónicas que es 100 veces más rápida que 4G. Esto significa que las velocidades de descarga 5G pueden alcanzar los 10 gigabits por segundo. Aplicaciones 5G en un futuro próximo:

- Vehículos autónomos.
- Complex Artificial Intelligence.
- Virtual Reality (VR) and Augmented Reality (AR).
- Complex IoT like drones.
- Mobile Services Resembling Broadband Perspectives.
- Dedicated network slicing.

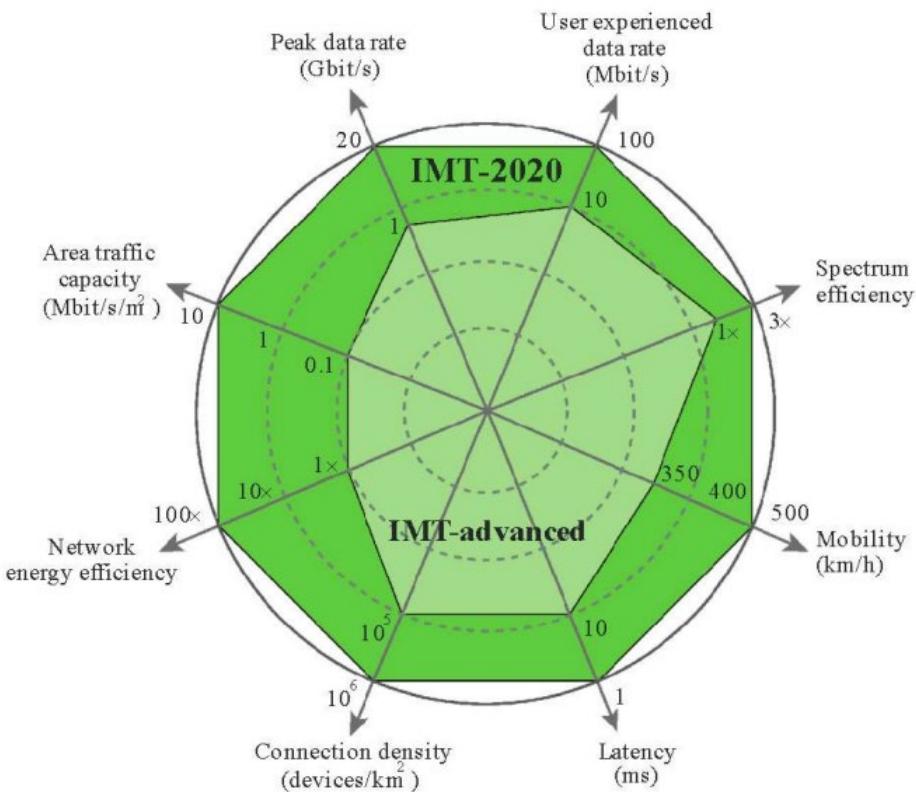


# 5G

5G se basa en 3 características importantes.



## 5G - Objetivos de rendimiento Unión Internacional de Telecomunicaciones (UIT)



The International Telecommunication Union is a specialized agency of the United Nations responsible for many matters related to information and communication technologies.



# Hay más....

