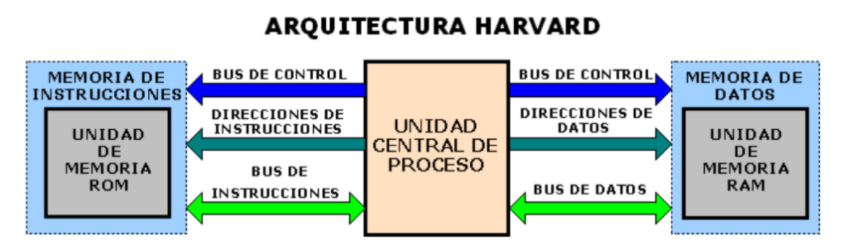
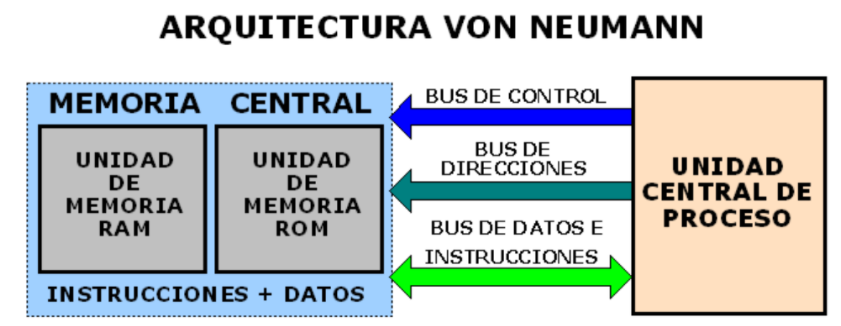
DICCIONARIO

* **Transistor**: El transistor es un dispositivo electrónico semiconductor utilizado para entregar una señal de salida en respuesta a una señal de entrada.
* **Ley-de-moore:** expresa que aproximadamente cada 2 años se duplica el número de [transistores](https://es.wikipedia.org/wiki/Transistor) en un [microprocesador](https://es.wikipedia.org/wiki/Microprocesador).
* **Puertas-lógicas:** Una puerta lógica, o compuerta lógica, es un dispositivo electrónico con una función booleana u otras funciones como sumar o restar, incluyen o excluyen según sus propiedades lógicas. XOR / AND / OR/ NOR/
* **Periférico:** En informática, periférico es la denominación genérica para designar al aparato o dispositivo auxiliar e independiente conectado a la placa base de una computadora.
* **Arquitectura-von-neumann:** La arquitectura de la ALU / CPU etc.
* **Arquitectura-harvard:** La diferencia principal entre las dos arquitecturas se encuentra en el mapa de memoria: mientras que en la **arquitectura Von Neumann** hay un único espacio de memoria para datos y para instrucciones, en la **arquitectura Harvard** hay dos espacios de memoria separados: un espacio de memoria para los datos y un espacio de instrucciones.





* **Sistema-operativo:** es un conjunto de programas que permite manejar la memoria, disco, medios de almacenamiento de información y los diferentes periféricos o recursos de nuestra computadora, como son el teclado, el mouse, la impresora, la placa de red, entre otros.
* **Hardware**: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.
* **Software**: Se conoce como software​, logicial o soporte lógico al sistema formal de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.
* **Raspberry pi:** es un ordenador del tamaño de una tarjeta de crédito. ... Consiste en una placa base que soporta distintos componentes de un ordenador como un procesador ARM de hasta 1500 MHz, un chip gráfico y una memoria RAM de hasta 8 GB. Además, tiene otras muchas otras posibilidades.
* **Arduino**: El arduino es una placa que tiene todos los elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador. Es decir, es una placa impresa con los componentes necesarios para que funcione el microcontrolador y su comunicación con un ordenador a través de la comunicación serial. **Arduino es una plataforma de creación de electrónica de código abierto**basada en hardware y software libre, lo que permite que cualquiera pueda utilizarlos y adaptarlos.

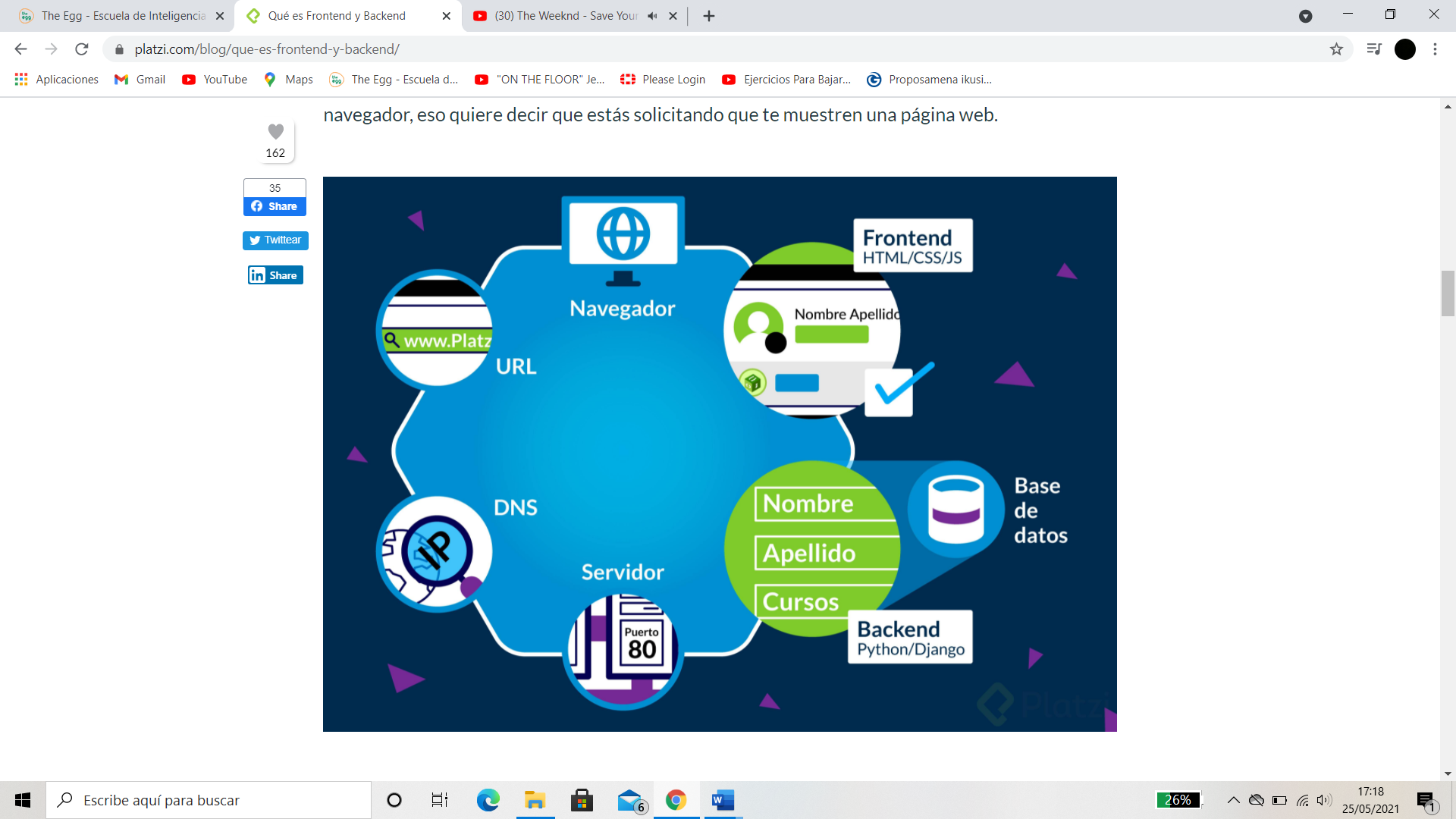
**Arduino y Raspberry Pi son dos conceptos totalmente diferentes**, por lo que es un poco difícil hacer una comparación entre ambos. Son dos productos con diferentes finalidades, aunque por su versatilidad la imaginación de la comunidad maker haya hecho que ambos sean utilizados para crear [todo tipo de proyectos de electrónica](https://www.xataka.com/makers/46-proyectos-makers-para-hacer-verano-arduino-raspberry-pi).

Aunque ambos son proyectos cuyo software es de código es abierto y cualquiera puede revisarlo, **una de las principales diferencias está en la filosofía del hardware**. El hardware de Arduino es también abierto para que cualquiera pueda crear sus propias versiones de la placa, mientras que la Raspberry Foundation mantiene el control sobre las placas Raspberry Pi, y sólo ellos las crean y fabrican.

* **Compilador:** es un tipo de traductor que transforma un programa entero de un lenguaje de programación a otro.​ Usualmente el lenguaje objetivo es código máquina, aunque también puede ser traducido a un código intermedio o a texto.

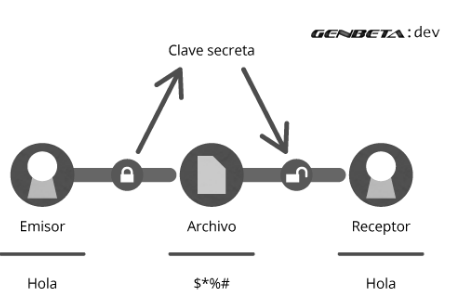
Proceso de traducción que convierte un programa fuente escrito en un lenguaje de alto nivel a un programa objeto en código máquina y listo por tanto para ejecutarse en la computadora.

* **Intérprete**: que analiza el programa fuente y lo ejecuta directamente, sin generar ningún código equivalente.
* **Lenguaje-de-programación:** es un [lenguaje formal](https://es.wikipedia.org/wiki/Lenguaje_formal) (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de [instrucciones](https://es.wikipedia.org/wiki/Instrucci%C3%B3n_(inform%C3%A1tica)) o secuencias de órdenes en forma de [algoritmos](https://es.wikipedia.org/wiki/Algoritmo) con el fin de controlar el comportamiento físico o lógico de un [sistema informático](https://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico), de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina [programa informático](https://es.wikipedia.org/wiki/Software). Por ejemplo, Python, R, Java, Lisp.
* **Editor-de-código:** Un editor de código fuente es un editor de texto diseñado específicamente para editar el código fuente de programas informáticos. Puede ser una aplicación individual o estar incluido en un entorno de desarrollo integrado.
* **Lenguaje-de-alto-nivel:** Se procesa mucho más lento. nos referimos al tipo de lenguaje de programación que no expresa los algoritmos teniendo en cuenta la capacidad que tienen las máquinas para ejecutar órdenes, sino al que se utiliza teniendo en cuenta las capacidades cognitivas de los seres humanos.
* **Lenguaje de máquina**:  es el sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador de una computadora o el microcontrolador de un autómata.
* **Frontend** es la parte de un sitio web que interactúa con los usuarios, por eso decimos que está del lado del cliente.
* **Backend** es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web, por eso decimos que el backend corre del lado del servidor.

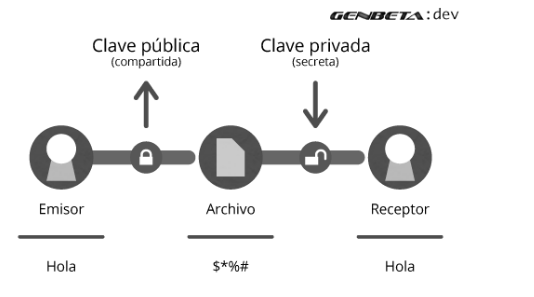


Si estás aprendiendo a programar, seguramente vas a encontrarte con muchos de los términos de este artículo. Además, en algún punto vas a tener que decidir si prefieres el [Desarrollo Frontend o el Desarrollo Backend](https://platzi.com/web), aunque es cierto que hay [Desarrolladores FullStack](https://www.youtube.com/watch?v=ePZBgZfoQLE) que involucran las dos áreas.

* **Repositorio de control de versiones:** Necesitamos una herramienta que gestione las partes del programa de cada individuo y todas sus versiones. Pero tranquilos, todo está inventado. Estas herramientas se conocen como repositorios de control de versiones y son una especie de biblioteca dónde todo se guarda de una manera organizada. Existen diferentes marcas: GitHub, bitbucket, subversion, etcétera.
* **Algoritmo:** Es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, solucionar un problema y llevar a cabo otras tareas o actividades.​
* **Diagrama de flujo:** Es la representación gráfica de un algoritmo o proceso. Permite ordenar ideas por los que a partir del mismo se escribe un programa en cualquier lenguaje de programación. Se utilizan tanto para planificar el desarrollo de un software como para diseños en otras disciplinas que nada tienen que ver con la informática.
* **Código RSA:** Es un algoritmo de cifrado asimétrico, o de clave pública. Sirve para cifrar y descifrar información.
* **Algoritmo de cifrado simétrico:** Este cifrado funciona con una sola clave. El emisor y el receptor deberá conocer la clave y el mensaje se cifrará y descifrará con la misma clave.



* **Algoritmo de cifrado asimétrico:** Este algoritmo funciona con dos claves, una clave pública (se puede difundir sin problema) y la otra privada (no se podrá revelar nunca). Las dos claves estarán vinculadas. Si queremos que tres compañeros de trabajo nos manden un archivo cifrado debemos de mandarle nuestra clave pública (que está vinculada a la privada) y nos podrán mandar de forma confidencial ese archivo que solo nosotros podremos descifrar con la clave privada.



La criptografía simétrica es más insegura ya que el hecho de pasar la clave es una gran vulnerabilidad, pero se puede cifrar y descifrar en menor tiempo del que tarda la criptografía asimétrica, que es el principal inconveniente y es la razón por la que existe la criptografía híbrida.

* **Criptografía híbrida:**
  + Generar una clave pública y otra privada (en el receptor).
  + Cifrar un archivo de forma síncrona.
  + El receptor nos envía su clave pública.
  + Ciframos la clave que hemos usado para encriptar el archivo con la clave pública del receptor.
  + Enviamos el archivo cifrado (síncronamente) y la clave del archivo cifrada (asíncronamente y solo puede ver el receptor).
* **Funciones en programación:**  Es una sección de un programa donde se “empaquetan” trozos de código para calcular o devolver un valor.
* **Algoritmo Hash:** Algoritmo de huella. Es una función matemática que nos genera una huella. Es la huella de un archivo, esta huella es siempre única. Algoritmo de cifrado de un solo camino. Se usa para verificar la autenticidad del archivo. Hay varias versiones desde 0 hasta el 5.
* **Estadística descriptiva:** Disciplina que se ocupa de estudiar los datos del pasado. Organiza, reúne y comunica información numérica.

Las medias de tendencia central son la media (contenida en promedio), mediana y moda. Son llamadas así dado que representan un punto central en torno al cual se encuentran las observaciones.

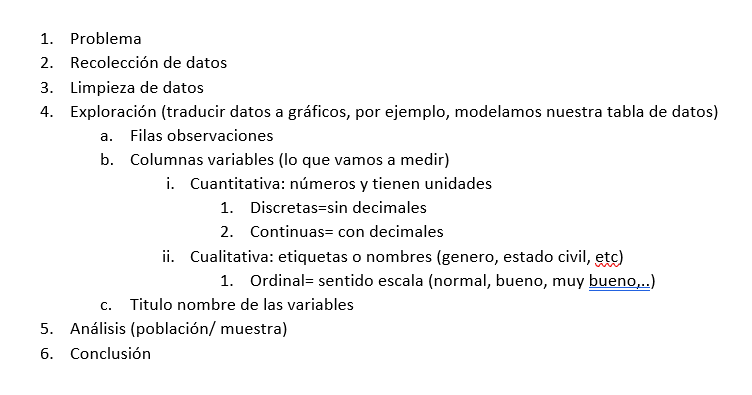
* **Moda**: se da cuando se repiten valores.
* **Media:** La media aritmética, es un promedio.
* **Mediana:** Medida de tendencia central.

Cuando hay presencia de datos extremos se recomienda utilizar la mediana como medida de tendencia central pues esta será más representativa que la media.

Las medidas de dispersión cuantifican la variabilidad de los datos. Las más usadas son la **varianza**, la **desviación estándar** y el **rango**.

Las medidas de posición reciben este nombre pues ayudan a comprender, valga la redundancia, cual es la posición de una observación con respecto al conjunto total de observaciones. Para ello se divide el conjunto total de observaciones en sub grupos con el mismo número de datos. Las medidas de posición más usuales son los **percentiles**, **cuartiles**, **quintiles** y **deciles**.

* **Campana de Gauss:** Nos permite ver gráficamente donde se concentran el mayor porcentaje de la población.
* **Análisis exploratorio:** Se basa en gráficos y estadísticos que permiten explorar la distribución identificando características tales como: valores atípicos o outliers, saltos o discontinuidades, concentraciones de valores, forma de la distribución, etc.
* **Dato-estructurado:** Datos que pueden ser ordenados y procesados fácilmente.  Son archivos de tipo texto que se suelen mostrar en filas y columnas con títulos. Son datos que pueden ser ordenados y procesados fácilmente por todas las herramientas de minería de datos.
* **Dato-semiestructurado:** Datos quetienen algunas características consistentes y definidas, y no se limitan a una estructura rígida como la necesaria para las bases de datos relacionales.
* **Dato-noestructurado:** datos presentes en forma absoluta sin procesar.  Generalmente son datos binarios que no tienen estructura interna identificable. Es un conglomerado masivo y desorganizado de varios objetos que no tienen valor hasta que se identifican y almacenan de manera organizada. Ejemplo; correo electrónico, pdf, hojas de cálculo, imágenes, videos. Etc.
* **Pasos para analizar datos:**

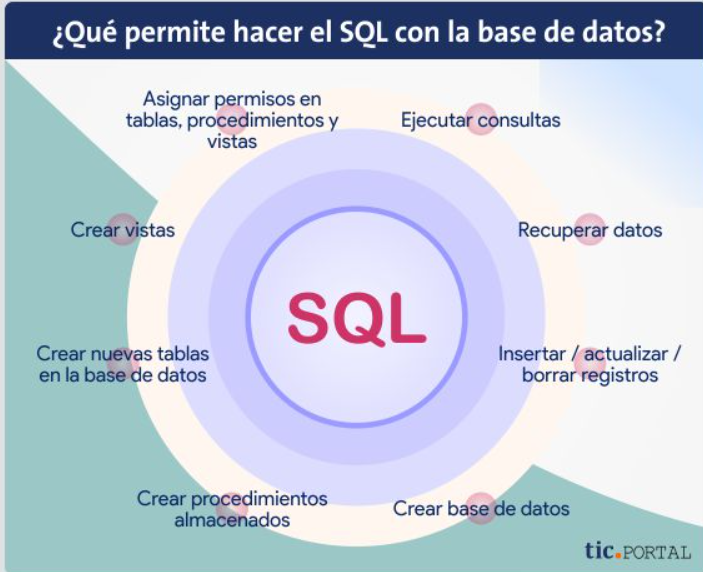


* **Base de datos:** Conjunto de información almacenada y consultada sistemáticamente.
* **Sistema gestor de la Base de datos:** Es el software de la Base de datos.
* **Bases de datos relacionales:** Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. Esta relación se realiza mediante un key ID, una columna con datos que coinciden de en dos tablas como mínimo. Este identificador será único en cada tabla. El modelo relacional es el mejor para mantener la consistencia de los datos en todas las aplicaciones y copias de la base de datos (denominadas *instancias*). El lenguaje de programación que se utiliza para hacer consultas en las BD relacionales es SQL.
* **Bases de datos distribuidas:**  Consiste en el almacenamiento de porciones de la base de datos en diferentes ubicaciones físicas y, por tanto, el procesamiento está distribuido o replicado entre los distintos puntos de una red de trabajo. Consecuentemente, la base de datos distribuida tiene una mayor disponibilidad de los datos debido a sus múltiples ubicaciones. De esta forma, si una de las bases de datos fallase, se podría seguir funcionando correctamente, aunque puede que un poco ralentizado. Sin embargo, esta misma característica de estar distribuido puede provocar que haya duplicidad de los datos y un menor nivel de seguridad.
  + **Homogéneas:** Mismo esquema y mismo sistema de gestión de la base de datos.
  + **Heterogéneas:** Se suelen utilizar entre distinta empresas u organizaciones.
* **Bases de datos No SQL:** Plantea modelos de datos específicos de esquemas flexibles que se adaptan a los requisitos de las aplicaciones más modernas.  Es una forma más flexible de almacenar, ordenar y captar mayor cantidad de datos comparado con base de datos relacionales.
  + **Bases de datos documentales:** Son bases de datos orientados a documentos. Estos BD se utilizan para almacenar, recuperar y administrar datos semiestructurados. Las bases de datos documentales almacenan cada uno de los registros y los datos asociados en un solo documento. Estas bases de datos ofrecen mucha flexibilidad, escritura rápida, y consultas rápidas gracias a su gran capacidad de indexación. Entre las BBDD documentales más reconocidas encontramos la ya conocida por nosotros a MongoDb, de 10gen, y CouchDB, de Apache.
  + **Bases de datos orientadas a grafos:** Herramientas para trabajar con datos complejos. Nos permiten representar los datos en estructura de grafo. Permiten ejecutar consultas y almacenar datos de cualquier característica sin preocuparnos por el volumen de datos. Twitter es uno de los muchos que utilizan esta forma de almacenamiento de datos. Entre las marcas más reconocidas está nuestra ya conocida Neo4j, pero esta no es la única. También están las conocidas HyperGraphDB, InfoGrid, AllegroGraph, InfiniteGraph, Sones y DEX/Sparksee.
  + **Bases de datos Clave/ Valor:** son modelos no relacionales que utilizan un método simple de almacenamiento de datos. Este tipo de BBDD toma los datos como un conjunto de pares “clave-valor” en los que las claves cumplen funciones de identificadores únicos. Algunos ejemplos, [BigTable de Google](https://cloud.google.com/bigtable/?hl=es), Dynamo de Amazon, Project Voldemort de LinkedIn, OracleNoSQL y Riak.
  + **Base de datos multivalor:** Son sistemas interesantes que incorporan diferentes características multidimensionales y NoSQL para la clasificación y manejo de los datos.
  + **Bases de datos orientados a objetos:** Bases de datos conformadas por objetos. Admiten lenguajes de programación como JAVA, Visual Basic o C++.  Entre las más Bases de Datos orientadas a objetos más conocidas tenemos a ObjectDB, Zope Object Database, ZooDB, [GemStone](https://en.wikipedia.org/wiki/Gemstone_(database)),  Objectivity y Realm.io.
  + **Bases de datos tabulares:** No es más que la estructuración de una BBDD en forma de tabla. Incorpora elementos en columnas y líneas. Cada una de las celdas genera intersecciones entre las columnas y las líneas. A estas intersecciones se le asignan una numeración única para establecer un orden eficiente de los datos. Están pensadas para grandes volúmenes de datos. Entre las principales DDBB de este estilo podemos conseguir a HBase de Apache que es utilizada para soportar el servicio de mensajería de Facebook, también a BigTable de Google y la versión abierta llamada LevelDB y a Hypertable.
  + **Bases de datos de Arrays:** sirven para trabar colecciones de datos conocidas como raster data. Sitúan los datos en una cuadricula regular con más de dos dimensiones. Estas bases de datos se utilizan para representar simulaciones, sensores y datos estadísticos.  Las más destacadas son  [SciDB, de Paradigm4](https://www.paradigm4.com/)
* **Bases de datos autogestionadas:** Bases de datos en la nube.

La nube nos permite resolver en segundos lo que tardaríamos días en nuestros ordenadores. Se utilizan miles de computadoras en paralelo para resolver el problema y solo se paga por aquello que se utiliza.

En las bases de datos no relacionales la información puede ser redundante, es decir se repite. La finalidad de este tipo de BD es la velocidad de los datos y no la integridad. Todas las empresas usan estas más de un tipo de Base de datos. No existe tecnología sin datos, por eso son vitales los bases de datos. La información es dinero.

* **SQL:** Es un lenguaje de consulta estructurada utilizada para administrar y recuperar información desde sistemas de gestión de las bases de datos relacionales**.** Es el lenguaje estándar de base de datos relacionales. Se utiliza para definir, gestionar y manipular los datos de la base de datos.
  + DDL:DATA DEFINITION LAGUAGE (create/drop/alter)
  + DML: DATA MANIPULATION LANGUAGE. (Select / insert / delete /update)
  + DSL: DATA CONTROL LANGUAGE. (Permisos de acceso a la BD)



* **MySQL:** Es un software que gestiona bases de datos cuyo lenguaje de consulta es SQL. Es decir, es un sistema de gestión de la base de datos relacionales, disponible bajo un esquema de doble licenciamiento: licencia pública GNU GPL así como con licencia comercial por Oracle Corporation.
* **PostgreSQL:** Es otro sistema de gestión de la base de datos relacionales, orientada a objetos de código libre, con un énfasis en la extensibilidad y el cumplimiento de los estándares.

En presencia de proyectos basados en aplicaciones web que requieran alta velocidad y no necesiten de consultas complejas, MySQL será nuestro mejor aliado. En cambio, si necesitamos alta fiabilidad y rendimiento manejando grandes bases de datos y consultas complejas, será PostgreSQL quién nos proporcionará las mejores herramientas.

* **MongoDB:** Sistema una base datos NO relacional (NoSQL). Está orientada a documentos. Es de código abierto y se encuentra bajo la licencia GNU AGPL v3.0.
* **ETL:** Es el proceso que se utiliza para dar valor a los datos. consiste en “Extraer” los datos crudos desde su origen (Source), “Transformarlos” según nuestras necesidades de analítica o la estructura que deseamos y “Cargarlos” a una base de datos orientada a procesos analíticos (Target). Es decir, se trata de traer, reformatear, limpiar, cargar, analizar y reutilizar los datos. Se utiliza mucho en BigData.



* **Data-preprocessing:** Es el proceso que se utiliza para eliminar o manipular datos antes de su utilización. Para poder garantizar el rendimiento a la hora de analizar los datos. Este concepto se utiliza en minería de datos. Se preparan los datos para su posterior utilización.
* **API:** Es una interface para que se comuniquen diferentes aplicaciones o programas software y compartan datos bajo ciertos estándares. Hace posible que una aplicación pueda conectarse con otras.

Terminología en el mundo de las APIs:

* + **Interface:** Capa de abstracción para que dos sistemas se comuniquen. Interactuar con un sistema sin necesidad de saber lo que pasa por debajo.
  + **Arquitectura software:** La forma en que esta diseñado un sistema.
  + Servicio web: permite la comunicación de equipos que están en la misma red. Se utiliza la comunicación http.
  + **Rest:** Es una arquitectura. Una representación de transferencia de estado. Se pueden definir los permisos, ya que implica que puedan guardarse los datos en caché. La arquitectura más usada actualmente.
  + **Protocolo de comunicación:** Es un conjunto de normas que permite la comunicación entre ordenadores, estableciendo la forma de identificación de estos en la red, la forma de transmisión de los datos y la forma en que la información debe procesarse. Se trata de las reglas o el estándar que define la [sintaxis](https://es.wikipedia.org/wiki/Sintaxis), [semántica](https://es.wikipedia.org/wiki/Sem%C3%A1ntica) y [sincronización](https://es.wikipedia.org/wiki/Sincronizaci%C3%B3n) de la [comunicación](https://es.wikipedia.org/wiki/Comunicaci%C3%B3n), así como también los posibles [métodos de recuperación de errores](https://es.wikipedia.org/wiki/C%C3%B3digos_detectores_y_correctores_de_error). Los protocolos pueden ser implementados por [hardware](https://es.wikipedia.org/wiki/Hardware), por [software](https://es.wikipedia.org/wiki/Software), o por una combinación de ambos.
  + **XML/ Javascript object notation (JSON):** Son dos diferentes formatos que se utilizan para enviar los datos.
    - **JSON:** Es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.
    - **XML:** es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

JSON es un formato más sencilloque el xml, que también es más fácil de procesar. Por lo tanto, es la solución más idónea para aquellos dispositivos que no requieren de grandes procesos.

* + **Token:** Objeto con todos los datos de la autentificación.

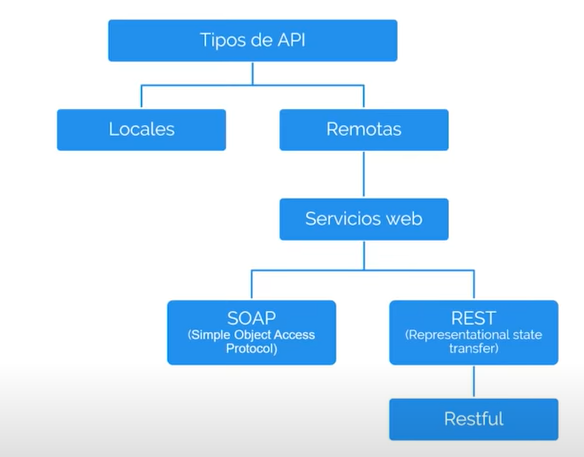
Ejemplos API:

* Open Street map (https://www.openstreetmap.org/#map=6/40.007/-2.488)
* Open weather map (<https://openweathermap.org/>)
* API-Football (https://www.api-football.com/)

Desarrollar una API Rest implica:

Rest (Representational State Transfer), es un modelo de arquitectura web basado en el protocolo HTTP para mejorar las comunicaciones cliente-servidor, mientras que Restful Web Service o Restful API son programas basados en REST.

* + Cada consulta tiene una URI (identificador único)
  + Solicitaciones realizadas, el servidor devuelve un código con diferentes informaciones
  + También se pueden utilizar métodos HTTP
  + Se pueden devolver datos en diferentes formatos



* **La arquitectura cliente-servidor:** es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.
* **Dirección IP:** Es una dirección única que identifica a un dispositivo en Internet o en una red local. Las direcciones IP son el identificador que permite el envío de información entre dispositivos en una red. El rango completo de direcciones IP va desde 0.0.0.0 hasta 255.255.255.255.

Cada individuo o empresa con un plan de servicio de Internet tendrá dos tipos de direcciones IP: sus direcciones IP privadas y su dirección IP pública. Los términos “pública” y “privada” se relacionan con la ubicación de la red, es decir, una dirección IP privada se utiliza dentro de una red, mientras que una pública se utiliza fuera de ella.

* **Máscara de red:** Es una combinación de [bits](https://es.wikipedia.org/wiki/Bits) que sirve para ámbito de una [red de ordenadores](https://es.wikipedia.org/wiki/Red_de_ordenadores). Su función es indicar a los dispositivos qué parte de la [dirección IP](https://es.wikipedia.org/wiki/Direcci%C3%B3n_IP) es el número de la red, incluyendo la subred, y qué parte es la correspondiente al [host](https://es.wikipedia.org/wiki/Host).
* **Host:** Se usa en informática para referirse a las [computadoras](https://es.wikipedia.org/wiki/Computadora) u otros [dispositivos](https://es.wikipedia.org/wiki/Perif%C3%A9rico_(inform%C3%A1tica)) conectados a una [red](https://es.wikipedia.org/wiki/Red_de_computadoras) que proveen y utilizan servicios de ella.
* **Servidor de correo:** Es una aplicación de red de computadoras ubicada en un servidor de internet, para prestar servicio de correo electrónica. De forma predeterminada, el protocolo estándar para la transferencia de correos es SMTP.
* **FTP:** Es un protocolo de transferencia de archivos. Cuando subimos ficheros a un servidor probablemente utilicemos un servidor FTP.
* **Un protocolo** se puede entender como un idioma. Son una secuencia de bits debidamente alineadas para que las computadoras se entiendan. Siempre hará falta un cliente (navegador) y un servidor y el idioma (protocolo) para que se entiendan.
* **Servidores DNS:** Es un sistema de nombres de dominio. Estos servidores se encargan de traducir una dirección web (url) en una dirección IP. De esta manera mi ordenador sabrá exactamente a que servidor conectarse y acceder así a los contenidos de la página. No cifran ni esconden tu dirección IP.
* **Ataque DDOS:** Se ataca al servidor desde muchos ordenadores para que deje de funcionar. Lo que puede provocar es una caída de web.
* **Conexión VPN:** Es una red privada virtual. Permite tener varios dispositivos conectados a un mismo router, cada uno con una dirección local de IP no visible desde internet. Es decir, una red local, un conjunto de dispositivos conectados de tal manera que puedan compartir archivos e impresoras sin necesidad de pasar por internet. Permite el teletrabajo, etc.



* **Red LAN:** Red de comunicación entre ordenadores situados en el mismo edificio, de forma que permita el intercambio de datos y la compartición de recursos. Es una red de área local.
* **Red WAN:** Es una red de área metropolitana. Es una red de computadoras que se extiende en una gran franja de territorio.
* **Router:** Es un dispositivo que permite interconectar computadoras que funcionan en el marco de una red. Su función es la de establecer la ruta que destinará a cada paquete de datos dentro de una red informática.

