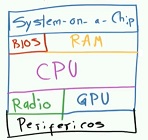
# **Qué es un system on a chip**

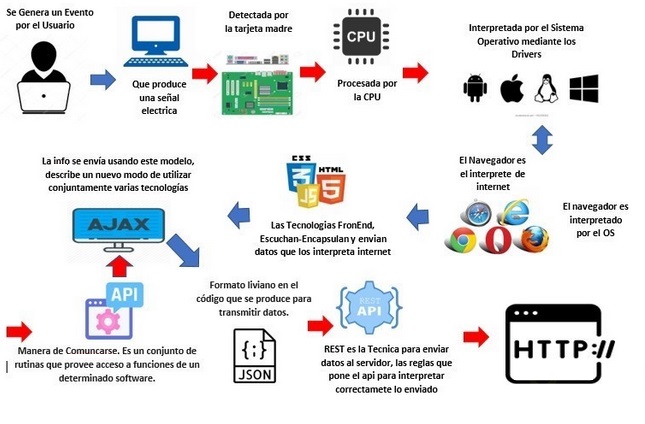
La arquitectura rápida y compleja en un sistema tradicional podemos hoy tenerlo en un dispositivo pequeño gracias a un **system on a chip**.

Un system on a chip es una CPU que también tiene una memoria RAM y un disco duro en un chip.

También incluye chips especializados que permiten realizar algunos procesos necesarios por un dispositivo móvil.

# **Cómo funcionan las computadoras y los teléfonos móviles**

Cuando enviamos un e-mail a una dirección de correo lo que sucede por detrás es lo siguiente:



La señal de la digitación en teclado es transferida por la motherboard al CPU y de la CPU al sistema operativo, que se comunica con estos componentes a través de los drivers.

Cuando das enter en Google Chrome se produce un Evento, que es mostrado por HTML, y accionado por JavaScript, que pasa por las API (interfaces de programación de aplicaciones/**A**pplication **P**rogramming **I**nterface), que son AJAX (**A**synchronous **J**avaScript **A**nd **X**ML) > JSON (**J**avaScript **O**bject **N**otation) > REST> **R**epresentational **S**tate **T**ransfer que es una técnica para enviar datos al servidor, para finalmente pasar por el protocolo HTTP protocolo de transferencia de hipertexto.

Es este el inicio de una dirección de un sitio web, luego sigue el dominio, que para nosotros se vería como Gmail.com y para los computadores como una serie de números separados por puntos. Por ej: 172.217.162.14

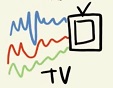
Para saber el dominio en números de un sitio, podemos ir a la terminal (en Windows el PowerShell) y escribir: ping google.com, y el sitio automáticamente nos mostrará la ip más cercana a nuestro país, es decir, desde Colombia es diferente que desde Japón el grupo de números.

Al dar la orden desde el navegador, esta será transmitida a un servidor en Google de sistema operativo linux, que puede ser Apache o Nginx, y que puede procesar la información en PHP, JAVA, PYTHON Y GO, entre otros. Creará una copia de la información en una base de datos MySQL, Oracle, PostgreSQL.

Una vez damos a enviar un correo, la dirección a la que lo enviaremos, al igual que las páginas web, tendrá un dominio como @platzi.com, que indicará a qué servidor será enviado el e-mail. Enviar SMTP/ Recibir POP3.

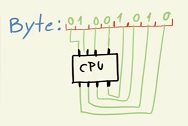
# **Qué son Bits y Bytes**

La electricidad funciona con ondas, si modulamos la onda con una frecuencia y ritmo y la conectamos a un altavoz, la onda se tomará la forma de sonido, siendo esa la forma en la que podemos almacenar el sonido, el cual dependerá del tono de la onda.

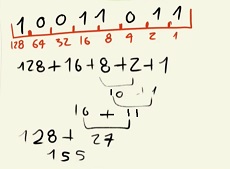
La televisión funciona de forma similar, sólo que con ondas diferentes para diferentes colores.

Las ondas electromagnéticas tienen subidas y bajadas, las subidas representadas por un 1 y las bajadas por un 0.

* Bit: 1 o un 0

El sonido que reproducían los primeros modems antiguamente, eran esos bits siendo transmitidos.

En 1956, IBM crea los bytes, que eran agrupaciones de 8 bites, que en ese tiempo eran el límite de información que podía ser recibido por los procesadores, que tenían 8 “**paticas**”

Los bytes reciben su lugar y valor leyendo de derecha a izquierda, multiplicando sus valores por 2 hasta llegar a 128, los ceros reciben un valor de 0 multiplicado por el número anteriormente descrito y los de 1 se multiplican a sí mismos por el valor de abajo, y luego se suman.

Ejercicio del minuto 10:15 = 155

@ es el signo representado en bytes como 64, siendo entonces Alt - 64. Su byte asignado es entonces 01000000.

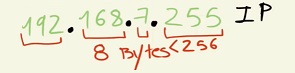
Así como el @, cada letra también posee un byte, y está todo registrado bajo la nomenclatura ASCII.

## Assembler

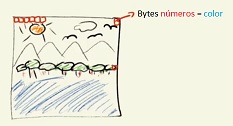
Son Bytes *especiales* de instrucciones del procesador. Están hechas para ser leídas a alta velocidad por la CPU.

TODO EN LA HUMANIDAD ACTUAL SON BYTES.

## Las IP son bytes

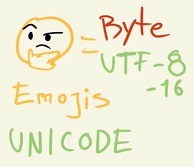
Por ejemplo 192.168.255, si vemos esta dirección, las IP son bytes que son menores a 256, que es el número máximo que podemos formar con los números desde 1 a 128.

## Los bytes pueden ser colores

Los bytes pueden representar los números presentes en cada uno de los pixeles, el número de bytes en una imagen dependerá del número de pixeles de la misma. Algunas pueden utilizar 2 bytes para representar el color, y serán mucho más pesadas que las que utilizan 1 solo byte.

## Para idiomas diferentes

Cada idioma cuenta con caracteres diferentes, como lo puede ser el chino mandarín, idiomas como el ruso, el japonés y demás, y antiguamente, cada uno contaba con su propia tabla, hasta que se creó el UTF-8, que permite que todos estos caracteres puedan ser compatibles entre sí

Cuando apareció el UTF-16, se abrió un nuevo horizonte con más espacios, y cuando ya todos los idiomas funcionaban bajo este estándar, los desarrolladores empezaron a agregar emojis.

UNICODE es la organización encargada de manejar todo lo concerniente al UTF-16, lo que incluye también determinar qué emojis serán agregados al sistema, lo que depende de Samsung, Google, Windows.

# **Procesadores y arquitecturas de CPU**

Vivimos en un mundo donde todo es un computador como celulares, Tablets, televisores, etc.

Existen una arquitectura para computadores de escritorio y laptops, estos internamente tienen:

* **CPU**: Su velocidad de mide en GHz y tienen Cores que son el número de instrucciones en paralelo que pueden hacer.

En este momento se hacen las CPUs en silicio y estas se calientan de acuerdo a la cantidad de procesos.

* **BIOS**: Es un pequeño sistema que arranca el computador y comprueba la conexión del hardware.
* **Disco duro**: Es donde se almacena el sistema operativo y los archivos que guardamos.
* **Memoria RAM**: Los datos del sistema operativo se envía a la memoria que sirve como intermediario con la CPU, porque es una memoria de acceso rápido, esta memoria solo funciona cuando hay electricidad.
* **GPU**: Puede procesar en paralelo los pixeles de la pantalla y enviarlo en tiempo real para representarlos.

# **Qué es la memoria RAM y cómo funcionan los discos duros**

Sabemos que los archivos se almacenan en el disco duro y la hora de abrirlo, se cargan en memoria RAM, pero… ¿Cómo exactamente funciona eso?

* **Los discos duros antiguos funcionan como los vinilos antiguos**: tienen un cabezal que va leyendo los datos que pasan por debajo de ella a medida que el disco gira a una velocidad constante.

Los discos duros son lentos porque deben posicionarse donde está el archivo almacenado y esto puede implicar que el brazo mecánico que mueve el cabezal se tome mucho tiempo en encontrar todos los pedazos de datos del archivo.

La memoria RAM es más rápida ya que puede acceder a los datos almacenados de manera instantánea.

Los discos duros de estado sólido no tienen el cabezal ni los discos que giran, sino que son más parecidos a las memorias RAM: funcionan electrónicamente. A pesar de eso guardan los datos en memorias flash, que son un poco más lentas que las RAM.

* **Los discos duros no son volátiles**: Guardan la información de manera persistente, aunque se les quite el suministro de energía.
  + **Además, almacenan los archivos de manera secuencial**: Para almacenar un archivo, éste se parte en varios pedacitos y se guarda la posición de cada uno de estos pedacitos y su ubicación en el disco para poder leerlos secuencialmente.

Los sistemas de archivos son convenciones internas de los sistemas operativos para poder acceder a los archivos almacenados.

* En Linux existe: ext3 o ext4
* En Windows existía: FAT16 o FAT32 (File Allocation Table), que fue reemplazado por NTFS (New Technology File System)
* En Mac OSX el sistema de archivos se llamaba HFS (Hierarchical File System) pero ahora se llama AFS (Apple File System) en macOS Sierra

Cuando abrimos un archivo, el CPU se lo pide al disco duro y luego lo lleva a la memoria RAM para leerlo.

En la RAM están todos los programas y archivos que están en ejecución. Si abrimos un archivo con el Bloc de Notas, por ejemplo, ambos deben estar cargados en la RAM. Y el CPU puede acceder a la memoria RAM a través de un índice compartido, es decir, un índice que indica en qué posiciones de memoria se encuentran qué partes de un archivo o proceso.

Los datos viajan a través de un conjunto de cables paralelos llamado bus de datos, que comunica el CPU con el disco duro y la RAM para permitir la transferencia de datos.

# **GPUs, tarjetas de video y sonido**

Sabemos cómo los archivos se cargan en memoria, pero ¿Cómo veo en pantalla que el archivo se ha abierto?

Esto se logra gracias a la Graphic Processing Unit o GPU.

La CPU puede ejecutar cualquier proceso, incluido el dibujado en pantalla de ciertos datos. Pero no es ella quien se encarga, sino la GPU, tarjetas especialmente fabricadas para realizar estas tareas.

La comunicación entre la CPU y la GPU se realiza actualmente a través de un socket llamado PCI-Express.

Estas placas de vídeo tienen sus propias unidades o núcleos de procesamiento y su propia memoria RAM.

Lo que sucede es que la GPU divide la pantalla en una matriz y cada núcleo se encarga de dibujar una parte de esa matriz, para lograr una mejor performance.

Esto es mucho más rápido de lo que podría lograr la CPU sola ya que debería dibujar pixel por pixel ella sola.

# **Periféricos y sistemas de entrada de información**

Los sistemas operativos normalmente tienen un núcleo llamado kernel, que es el principal elemento que los representa y es la primera parte del sistema operativo que se carga en la memoria RAM. El kernel del sistema operativo tiene acceso a todo en nuestra computadora:

* Nuestros archivos.
* Nuestros periféricos.
* A los datos de las aplicaciones.

El kernel, inmediatamente después de ser cargado en RAM, se encarga de cargar los drivers, que son pequeñas piezas de software que permiten interpretar las señales eléctricas del hardware, para que el sistema operativo pueda comunicarse con ellos.

Luego tenemos otro set de drivers que pueden ser los controladores de arranque llamados drivers de aplicación. Cuanto más nos alejamos del kernel, menos privilegios tenemos. Los drivers de aplicación deben pedirles permisos a los drivers anteriores para poder acceder al hardware.

La última capa la representan las aplicaciones. Esta es la capa que menos permisos tiene, ya que las aplicaciones no deberían poder acceder al hardware directamente.

# **Arquitectura de la computación**

En el inicio de la computación no existía un procesador y una memoria aparte. Las computadoras estaban más cerca de ser una máquina de escribir que una de las computadoras que conocemos ahora. Eran máquinas grandes y pesadas, que requerían ser trasladadas en aviones o camiones. El código binario se escribía en tarjetas perforadas, cuyas perforaciones (o falta de ellas) representaban los 1 y 0.

Hoy en día tenemos computadoras en nuestros propios bolsillos y las cargamos a todos lados, tenemos laptops cuyos monitores se pueden desacoplar y funcionan como Tablets, tenemos microchips que sirven como una computadora común y corriente.

Ese salto evolutivo en la computación ocurre gracias a la estandarización de la arquitectura de las computadoras, decidimos que un Byte son 8 bits, que la CPU es la encargada de procesar, que la GPU representa datos visualmente, que 1024 Bytes son un KiloByte, y que 1024 KB son 1 MB, que exista un puerto común como el USB que nos permite conectar otros dispositivos externos.

Estandarizamos la transferencia de datos y los protocolos de comunicación. Hay un formato definido para cada tipo de imágenes, hay una forma de escribir HTML para que el navegador lo interprete y pueda mostrarnos elementos visuales en la pantalla. Definimos una forma para comprimir un archivo.

# **Introducción a las redes, protocolos e Internet**

El concepto de que Internet es una nube es erróneo. La manera simple de definir internet es que son ordenadores conectados entre sí.

* **Switch**: Comparte la red para todo aquel que se conecte a sus puertos, eliminando su conexión una vez finalizada, envía réplicas de la información en todos los puertos cuando no conoce al destinatario.
* **Router**: Su función principal es enviar o encaminar paquetes de datos de una red a otra. Estos pueden o no tener Wifi. Es inteligente, necesita saber el nombre de la red, passwords y la lista de IP (DHCP). Los router normalmente recuerdan los equipos por medio de MAC. A veces funcionan como un switch y a veces necesitan uno.
* **IP list**: Dirección IP (Internet Protocol) es una matrícula identificativa que te define dentro de una red, ya sea esta interna o externa, de cara a Internet. Y una IP list no es más que una lista donde se guarda IPs en un router.
* **DHCP** (Dynamic Host Configuration Protocol): Protocolo de configuración dinámica de host, es un servidor que usa protocolo de red de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van quedando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después. Así los clientes de una red IP pueden conseguir sus parámetros de configuración automáticamente.
* **MAC Adress**: La dirección MAC (Media Access Control) es un número único que identifica las piezas de hardware que se conectan a internet, se le conoce también como dirección física.
* **Modem**: Permite la comunicación entre computadoras a través de la línea telefónica o del cable módem. Puede hacer parte de Router y Switch, es donde llega la conexión a la red por medio del ISP (Internet Service Provider).

# **Puertos y protocolos de red**

Los routers son las puertas enlace a diferentes redes. La dirección IP 127.0.0.1 **siempre** hace referencia al dispositivo local.

El router asigna IPs dentro de la red local y esa IP es única en esa red, hacia afuera todos los equipos se conectan con la IP que te da el proveedor de internet que tienes contratado.

Para asignar IPs un software se encarga de revisar la Mac address de cada dispositivo y asignarle una IP que esté disponible.

En los esquemas de red se crea una red virtual dentro de los sistemas operativos con un concepto interno que se le conoce como los puertos.

## **Un puerto es una puerta específica para un programa específico.**

Cada solicitud que tú haces desde tu pc a través de una red trabaja con una ip y un puerto amigo, los puertos sirven para identificar los miles de servicios que maneja un SO, por ejemplo: Cuando tu entras a twitter desde tu navegador tú estás haciendo una petición a (102.102.20.02, ejemplo de ip de twitter), y el puerto 80, pero si quisieras subir un archivo por protocolo ftp seria 102.102.20.02 por puerto 21 que se ve reflejado como 102.102.20.02:21 y así sucesivamente cambia el puerto dependiendo del servicio.

“Los protocolos son como un lenguaje de comunicación entre máquinas y los puertos son autopistas donde los mensajes del protocolo pueden transitar”.

La cantidad total de puertos disponibles es 65,535 gracias a que internamente hay dos bytes disponibles para definir la cantidad de puertos que existen.

## **Explicación del por qué tenemos 65, 535 puertos disponibles**

Un byte son 8 bits si tú tienes el valor 11111111 es igual a 255 por consiguiente dos bytes puede ser 256 (si incluimos el 0) por 256

256 \* 256 = 65.536

Ahora restamos 1 por el puerto 0, porque no es válido y obtenemos los 65.535

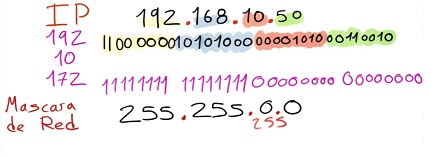
El sistema operativo tiene reservado los puertos del 1 al 1024, para usarlos necesitas elevar los privilegios de usuario.

## **Listado de puertos más utilizados**

* **HTTP**: Puerto 80;
* **HTTPS**: Puerto 443;
* **FTP**: Puerto 21;
* **FTPS/SSH**: Puerto 22;
* **POP3**: Puerto 110
* **POP3 SSL**: Puerto 995
* **IMAP**: Puerto 143
* **IMAP SSL**: Puerto 993
* **SMTP**: Puerto 25 (alternativas: puerto 26 / puerto 2525)
* **SMTP SSL**: Puerto 587
* **MySQL**: Puerto 3306
* **CPanel**: Puerto 2082
* **CPanel SSL**: Puerto 2083
* **WHM** (Webhost Manager): Puerto 2086
* **WHM** (Webhost Manager) SSL: Puerto 2087
* **Webmail**: Puerto 2095
* **Webmail SSL**: Puerto 2096
* **WebDAV/WebDisk**: Puerto 2077
* **WebDAV/WebDisk SSL**: Puerto 2078

# **Qué es una dirección IP y el protocolo de Internet**

IP es la sigla de **I**nternet **P**rotocol y una dirección IP es un número único con el cual una computadora o un dispositivo se identifica cuando está conectada a una red con el protocolo IP.

Cada dirección IP está compuesta por 4 números separados por puntos y son una forma de comprender números más grandes y complejos. Las direcciones IP tienen una estructura que las convierten en **privadas** o **públicas** y que además hacen parte de la máscara de red y el Gateway.

**Máscara de red**: Es un permiso de cómo se pueden asignar redes a través del límite. Por Ejemplo: Si tenemos una

IP 192.168.0.50

Máscara de red 255.255.255.0

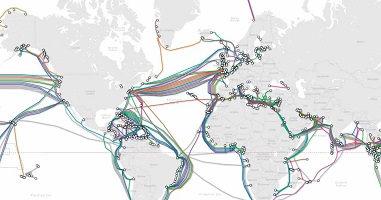
Él numero 255 limita la cantidad de redes de números IP que se pueden conectar. Sí en su lugar queremos tener la IP 192.168.120.50 deberíamos tener la máscara de 255.255.0.0

Las direcciones IP permiten que cada computador o dispositivo pueda conectarse al exterior, es decir a Internet, esto a través de tecnologías como NAT o **N**etwork **A**ddress **T**ranslation.

**NAT**: Es un protocolo que traduce de las direcciones de internet a las direcciones de red local.

**Gateway**: Es la IP del dispositivo principal que hace enrutamiento el cual se conecta al resto de todos los computadores de la red local y afuera, es decir, internet.

# **Cables submarinos, antenas y satélites en Internet**

La mayoría de personas imaginan que el acceso a Internet consiste en conexiones satelitales, lo cual es un error, pues los satélites están destinados sólo para áreas remotas. Internet funciona a partir de cables que atraviesan diferentes lugares del mundo.

Cuando usas tu computadora o dispositivo, este se conecta a un ISP o un prestador de servicios de Internet (ese a quién le pagas tu factura). De ahí, la conexión con diferentes puntos en el mundo a través de cables submarinos, que pueden ser de fibra óptica o cobre.

Estos cables pueden comenzar en una ciudad como Nueva York y terminar en Japón y aunque no parezca, la red de Internet un poco frágil pues los cables pueden romperse por diferentes causas, como las anclas de los barcos.

# **Qué es un dominio, DNS o Domain Name System**

DNS es una base de datos donde se sabe cuál nombre de dominio pertenece a que IP, por ejemplo:

www.google.com = 172.217.172.78

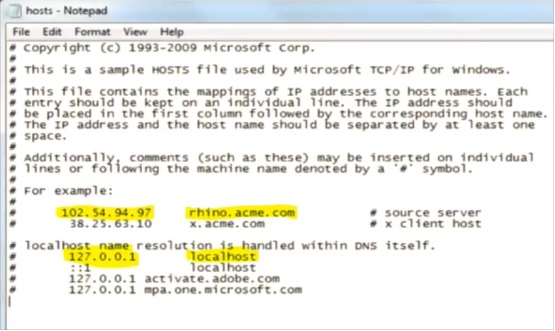
Hay DSN en todos lados, ya que tienen una copia exacta lo que permite acceder más fácil a los servidores. Existen subdominios, como por ejemplo api.google.com

Como los servidores de email son diferentes, estos DNS también guardan en sus bases de datos la información de correo electrónico de los dominios.

## **Mx Records**

Un registro MX es un tipo de registro, un recurso DNS que especifica cómo debe ser encaminado un correo electrónico en internet. Los registros MX apuntan a los servidores a los cuales envían un correo electrónico, y a cuál de ellos debería ser enviado en primer lugar, por prioridad.

Dentro de nuestros SO tenemos una pequeña BD “**tipo**” DNS llamada hosts, es un archivo donde traduce dominios a sus respectivas IPs, se usa este archivo para “bloquear” el acceso a ciertos sitios enviándolos a IPs erróneas, por ejemplo www.platzi.com -> 127.0.0.1 como resultado no cargaría ninguna página. También se puede editar el nombre del dominio el cual apunta a una dirección IP.

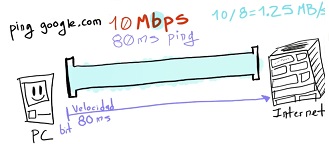


# **Cómo los ISP hacen Quality of Service o QoS**

Cuando queremos conectarnos a puntos muy lejanos, por ejemplo, el otro lado del mundo para nuestro proveedor de Internet es muy costoso por lo tanto prioriza las conexiones usando un QoS (Quality of Service) cuando la conexión está cerca, nuestro proveedor de internet hace una red MAN (Metropolitan Area Network) y el costo de eso es casi nulo.

Hay forma de burlar el QoS y simular una conexión más cercana y esto es a través del CDN (Content Delivery Network) que funcionan solamente para mostrar contenido estático como imágenes, videos, etc.

# **Cómo funciona la velocidad en internet**

La mayoría de los ISPs (Internet Service Providers) nos venden ancho de banda en Mb y debemos tener claro qué significa, ya que existe una importante diferencia entre Megabits y MegaBytes.

Otro aspecto importante en el funcionamiento del internet es la velocidad. A menudo confundimos la **velocidad** con el **ancho de banda** por eso debemos tener en claro que la velocidad del internet se mide obteniendo el tiempo que le toma a la información viajar a través de un punto a otro en milisegundos, a esto se le conoce como **ping** o **latencia**.

**Mb/s**: Es la cantidad de datos que caben en 1seg por el tubo de la conexión.

# **Qué es el Modelo Cliente/Servidor**

Las tecnologías utilizadas en aplicaciones web son:

**Bases de datos**: MySQL es una base de datos relacionales y MondoDB es una base de datos no relacional

**Backend**: Existen muchos lenguajes que puedes usar cómo Python, Ruby, JavaScript.

**Servidores**: Existen tecnologías como NGINX, Apache, Node.

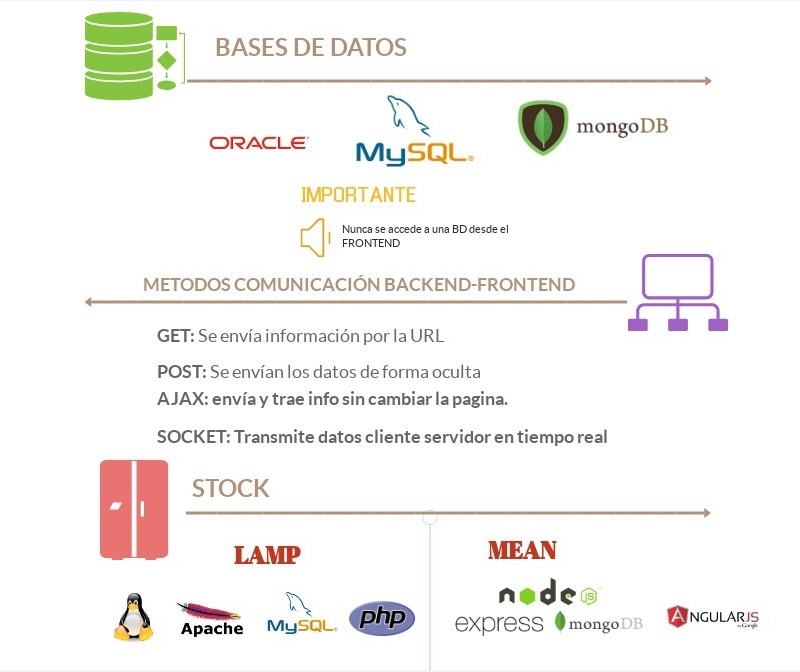
**Frontend**: Son las tecnologías que corren en el navegador, HTML, CSS y JavaScript.

A un grupo de tecnologías se les conoce como **Stack**.

Recuerda:

* Si tuvieras un código en el Frontend que se conectara a una base de datos, esta sería visible para todos.





Ejemplo de uso



# **Cómo funciona realmente un sitio web**

1. Los protocolos se encargan de manejar todas las peticiones que hacen las páginas de internet desde tu navegador hacia los servidores DNS, éstos transforman la dirección de la página web en una dirección IP y tu navegador se conecta a esa IP.
2. Una vez se tiene la dirección IP el navegador envía un HTTP request en donde envía información con las características del cliente y los requerimientos del mismo, es decir, Host requerido, página del sitio que necesita, tipo de navegador, versión del navegador, etc.
3. El servidor envía los resultados por medio del mismo protocolo HTTP en forma de un HTTP Response en donde manda todo el HTML del sitio web, así como otros datos que el navegador necesita.
4. Por último, se cargan los assets de nuestro sitio web y es aquí donde se descargan imágenes, sonidos, etc.