Web 101

TB022 - Esteban - Riesgo - Kristal

Qué es la Internet?

Es una red de computadoras conectadas entre sí.

Hoy en día casi todas las computadoras del mundo están conectadas a esta red (cuando hablamos de computadoras hablamos de casi cualquier dispositivo, una PC, un celular, una heladera).

Cuando navegamos en la web, en una red social, o lo que fuera, nuestro dispositivo está buscando en esta red otra computadora específica que tiene la información que nosotros queremos.

Cómo se conectan entre sí?

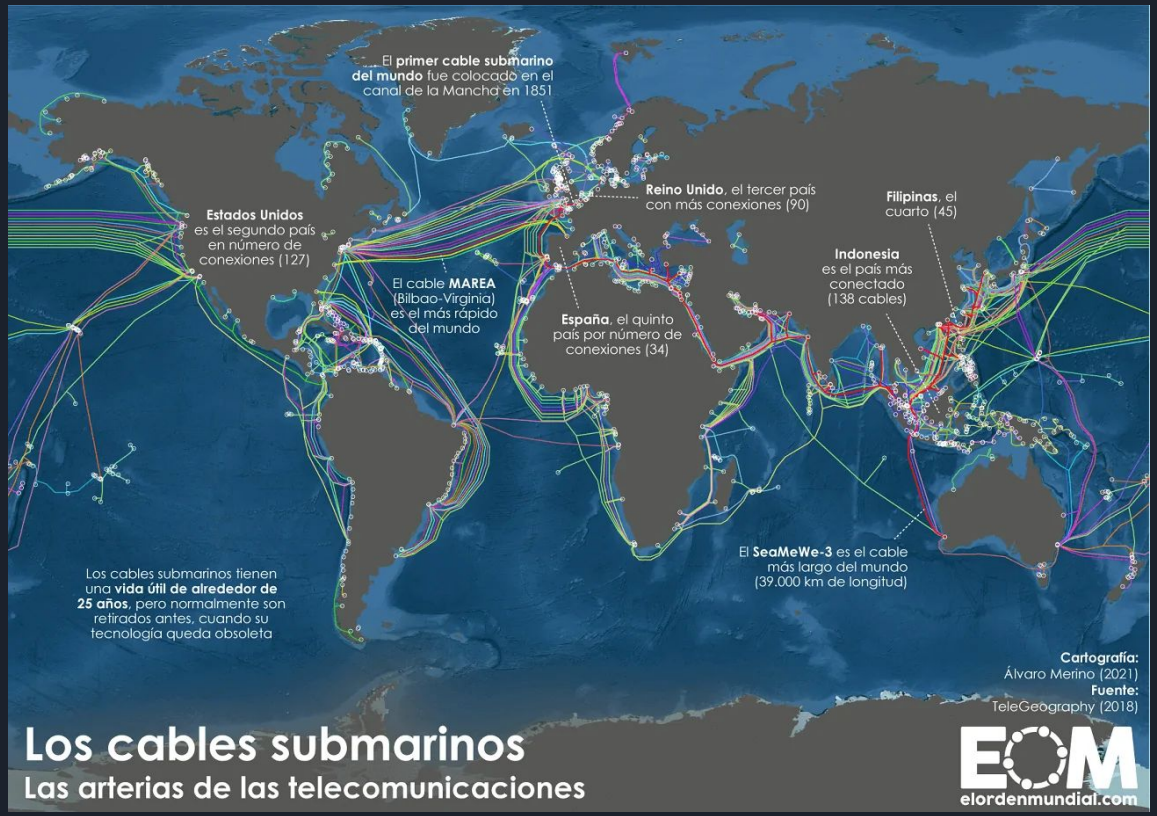
Cómo saben encontrar la computadora que quieren?

Qué es la Internet?

Versión interactiva.

Los cables submarinos conectan los continentes entre sí.

Todos los cables de Argentina salen por Bahía Blanca



Mientras más lejanos los dispositivos que uno quiere acceder, más tiempo tardará en ir y venir la información.

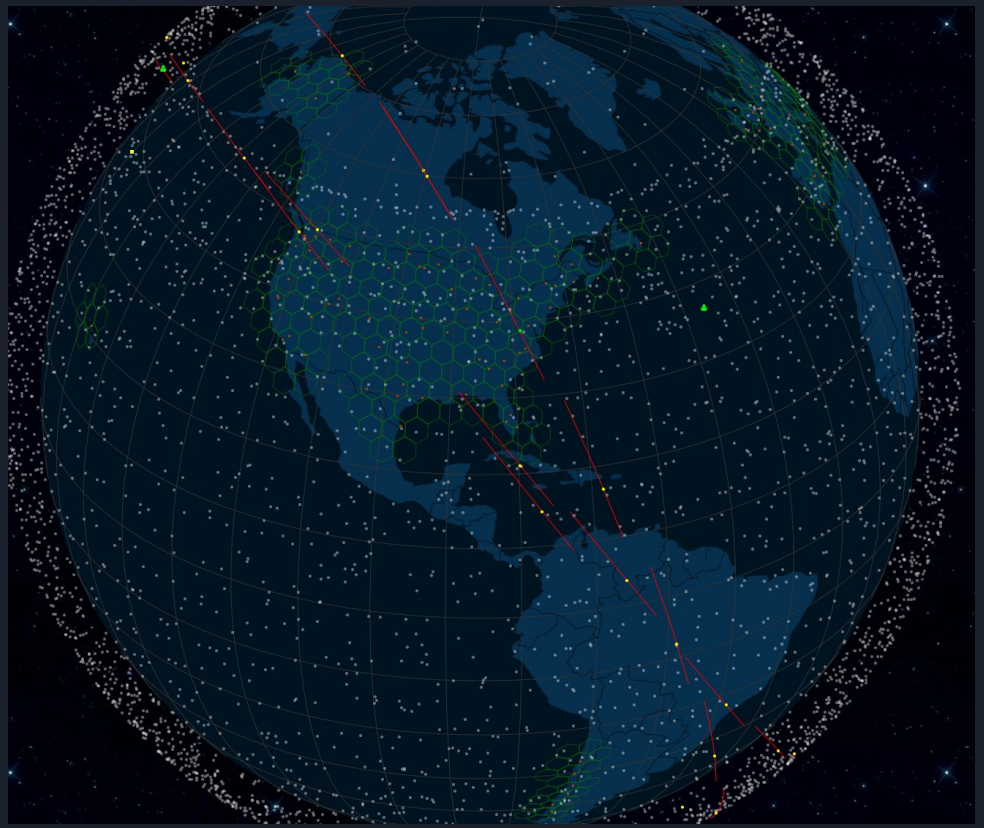
Esto se llama *delay*, *latencia* o *ping*, el tiempo que pasa entre que un emisor envía un mensaje y el receptor lo recibe.

Qué es la Internet?

La Internet satelital hoy en día es cada vez más común.

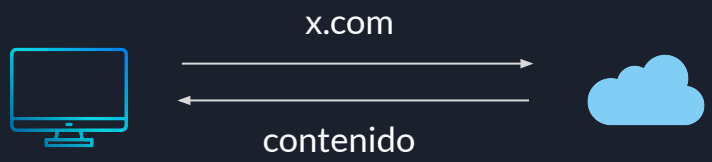
Starlink es uno de lo proveedores más conocidos y prolíficos.

Este mapa muestra todos los satélites que hay rotando alrededor de la Tierra (la mayoría son de Starlink).



Cómo funciona?

Qué pasa desde que abro un navegador web y escribo una dirección en el buscador hasta que veo algo en la pantalla?



Cómo funciona?

Qué pasa desde que abro un navegador web y escribo una dirección en el buscador hasta que veo algo en la pantalla?

HTTPS

Es un protocolo de comunicación de redes. Establece una forma estándar de comunicarse entre computadoras.

Además, provee medidas de seguridad para que la información vaya segura.

Por qué necesitamos el estándar?

La información en los cables o satélites viaja como 0s y 1s. Ambas partes necesitan un método inequívoco acordado de antemano para que la información que es enviada pueda ser entendida correctamente por quien la recibe.

TPC/IP

Es un protocolo de comunicación de redes. Establece una forma estándar de comunicarse entre computadoras.

Hay muchas capas entre lo que el usuario ve y los 0s y 1s que viajan en los cables. Cada capa tiene sus estándares y protocolos.

IP

Es una dirección inequívoca dentro de una red para identificar un dispositivo puntual.

IPv4 (Internet Protocol version 4): xxx.xxx.xxx.xxx

Tiene 4 segmentos, cada segmento es un número entre 0 y 255 (un byte, 8 bits)
Ocupa 4 bytes (32 bits)
Permite identificar 2^32 dispositivos distintos dentro de una misma red
Como no alcanza por sí mismo, hay muchas técnicas utilizadas hoy en día para ampliar la capacidad.

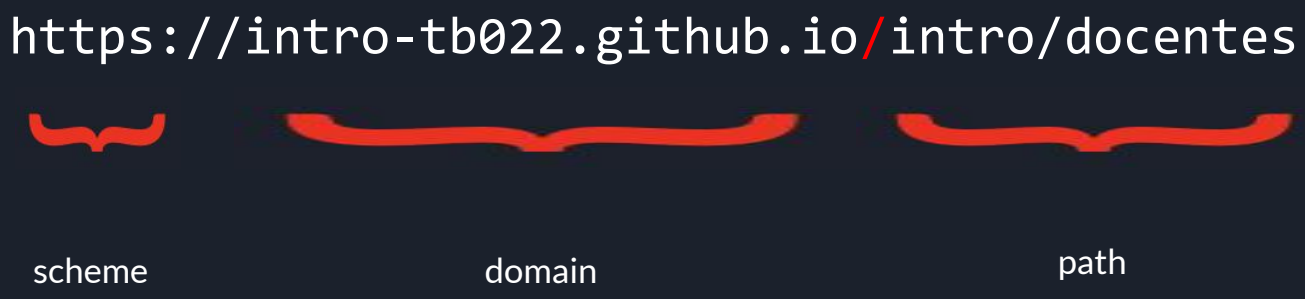
IPv6: XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX

Tiene 8 segmentos, cada segmento se representa en código hexadecimal (base 16) y va de 0 a FFFF (65535 en decimal, o 2^16)
Ocupa 32 bytes (128 bits)
Permite identificar a 2^128 dispositivos distintos dentro de una misma red

2^128 = 3.4028237e+38 o 340 undecillones. Por ejemplo, la Tierra pesa ~2^92 gramos.

URL: Uniform Resource Locator

Una URL es lo que normalmente conocemos y usamos en el día a día en un navegador web. Una URL permite identificar inequívocamente a un recurso en toda la web.



Cómo funciona?

Entonces, si las computadores se identifican como IPs, que son las URLs?

Cómo encontramos lo que queremos a partir de eso?

DNS (Domain Name System)

Es un servidor (computadora) especial que tiene guardado para cada dominio que conoce la IP que le corresponde

Los DNS son conocidos, cada ISP (Internet Service Provider, Fibertel, Movistar, Claro, etc) tiene esta información. Hay varios en todo el mundo. Según la locación global de la computadora que se está usando (determinado a partir de la IP de la misma), el ISP nos comunicará con el DNS más cercano y nos devolverá la IP del servidor* al que le corresponde el dominio de la URL que estemos queriendo conectarnos.

* El servidor al cual nos dirige el DNS también depende de la locación. Alguien en Japón entra al mismo sitio que nosotros pero es redirigido a una IP distinta.

Para saber más sobre DNS leer esto.

Cómo funciona?

Qué pasa desde que abro un navegador web y escribo una dirección en el buscador hasta que veo algo en la pantalla?

El path le indica al servidor qué parte del sitio queremos ver puntualmente

Y qué es el contenido del Sitio?

El mismo está definido por código HTML (que NO es un lenguaje de programación, es un lenguaje de maquetado).

Es texto que el navegador (Google Chrome, Firefox, etc.) sabe interpretar para mostrar las cosas bonitas por pantalla.

El HTML define la estructura de la página, el estilo (colores, transiciones, animaciones) y qué contenido multimedia se va a mostrar (imágenes, videos).

El contenido multimedia NO se devuelve con el HTML, solo las URLs al mismo (cada pieza, imagen, video, tiene su propia URL).

Y entonces, donde están alojados los videos e imágenes?

Contenido Multimedia

No se aloja en el mismo lugar que el resto del contenido porque es muy pesado en comparación. Se tardaría mucho más tiempo en retornar todo el contenido.

File Servers

Servidores especiales que se encargan de guardar archivos que pueden tener cualquier tamaño o tipo. Son ideales para guardar contenido multimedia.

Usan *bases de datos* especiales que guardan el contenido como un file system.

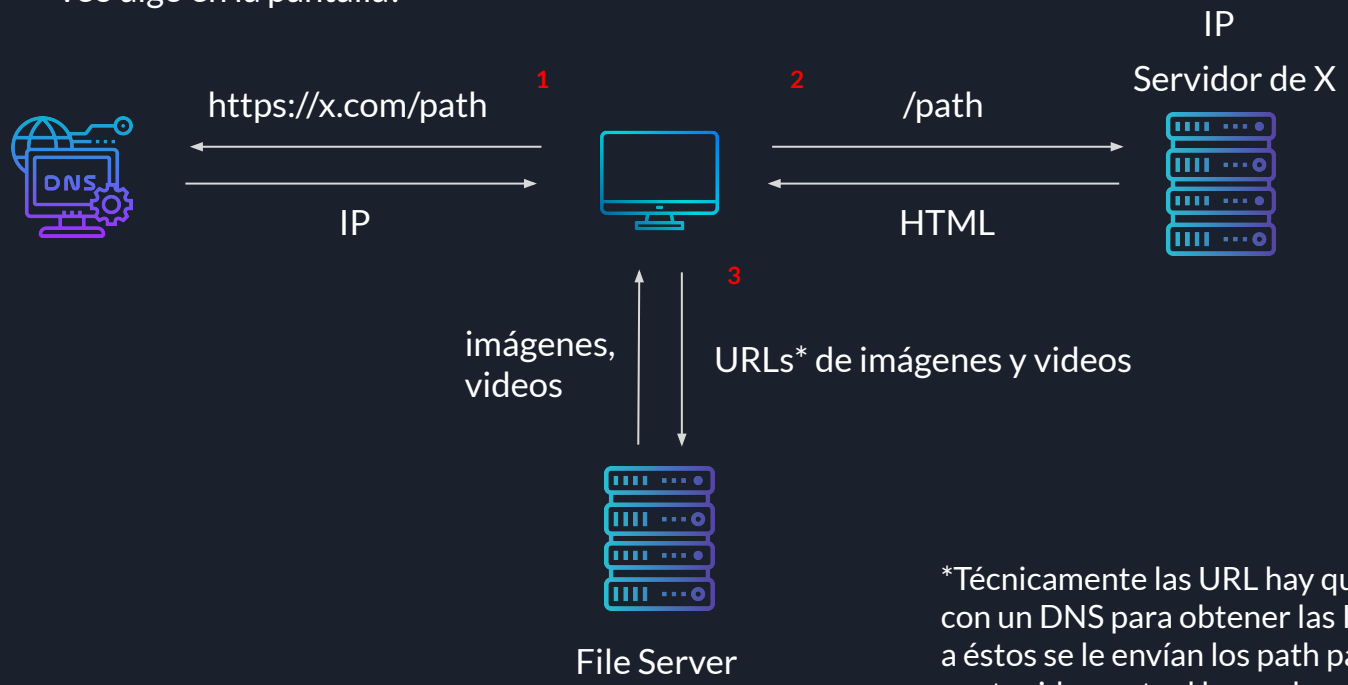
Hay varios proveedores muy conocidos (Amazon S3, Google Cloud Filestore) que son los más usados en casi todos los sitios.

Los sitios más grandes (Facebook/Instagram, Amazon y otros) usan sus propios File Servers

Una vez que el navegador obtiene el HTML, mientras renderiza el contenido de texto, va a buscar el contenido multimedia a partir de las URLs devueltas en el HTML y las muestra a medida que le van llegando.

Cómo funciona?

Qué pasa desde que abro un navegador web y escribo una dirección en el buscador hasta que veo algo en la pantalla?



*Técnicamente las URL hay que traducirlas primero con un DNS para obtener las IPs de los file servers y a éstos se le envían los path para que devuelvan el contenido puntual buscado

Y la información propia de la aplicación?

Donde guarda un servidor la información de sus usuarios?

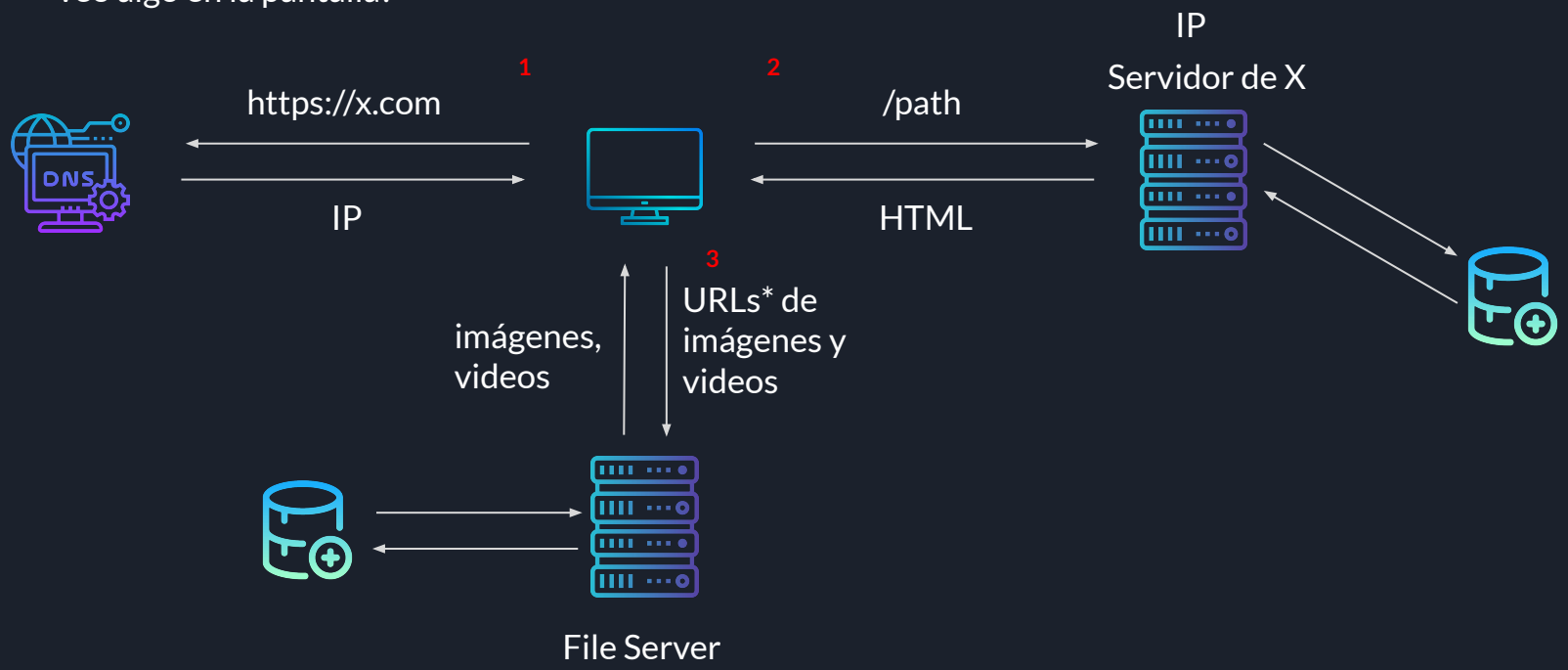
La mayoría de los servidores se conectan a algún tipo de base de datos que va a persistir la información para que los usuarios la pueda acceder en distintos momentos.

Hay muchos tipos distintos de bases de datos.

A veces un mismo servidor tiene muchas bases de datos distintos, según que tipo de información esté guardando.

Cómo funciona?

Qué pasa desde que abro un navegador web y escribo una dirección en el buscador hasta que veo algo en la pantalla?



*Técnicamente las URL hay que traducirlas primero con un DNS para obtener las IPs de los file servers

Esto es así siempre?

Más o menos. La estructura general sí, pero lo que va a cambiar es la parte del servidor.

Cuando una aplicación web empieza a ser muy grande, se vuelve complicado manejarla. Tanto por la organización de la misma, como el manejo de recursos o la misma colaboración entre gran cantidad de personas al mismo tiempo hace que en muchos casos sea necesario separar el servidor en varios servidores.

Puntualmente nos interesa a nosotros saber que la parte del frontend y la parte del backend suelen estar separadas. Son dos servidores distintos.

Backend vs Frontend

Un servicio ***frontend*** (FE) suele contener únicamente el HTML que se va a devolver y renderizar del lado del cliente (la computadora que está accediendo a un sitio puntual) y además código *Javascript* (JS) que se va a estar ejecutando en el cliente.

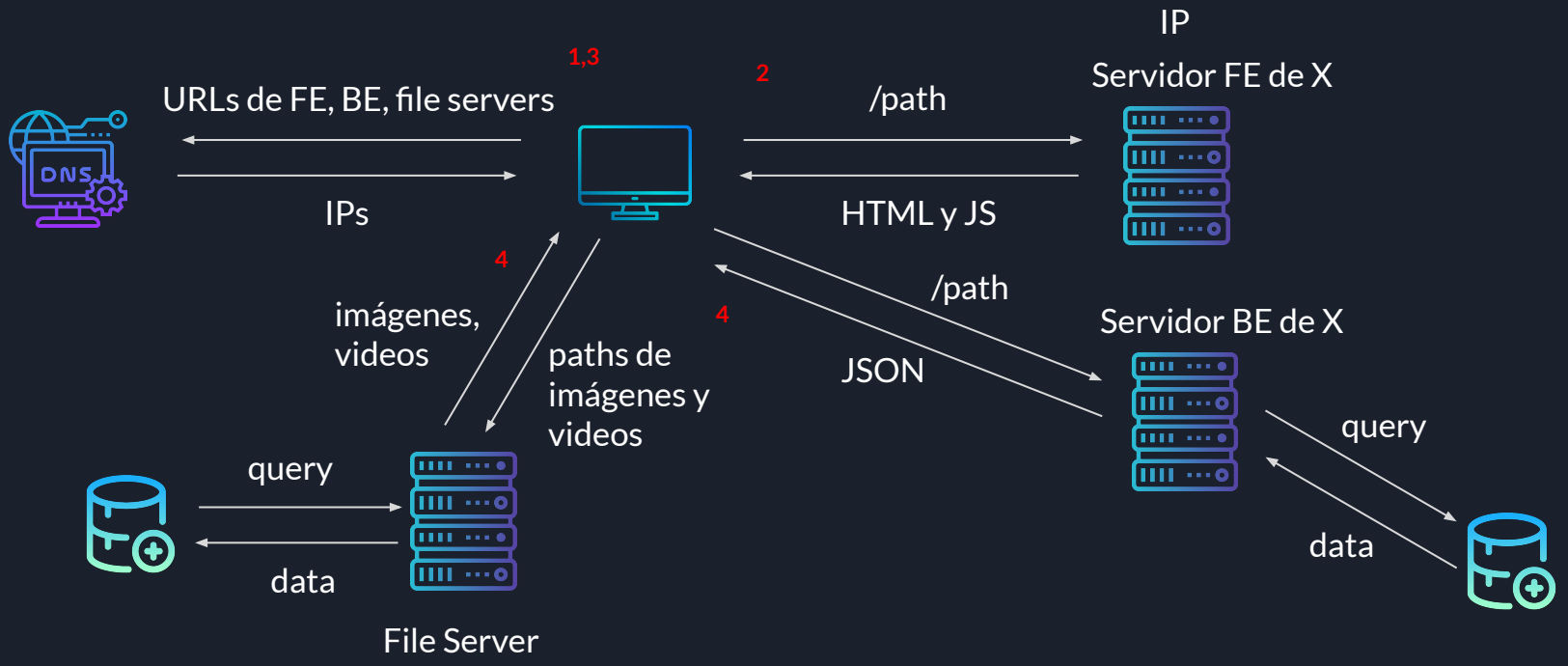
Este código JS va a poder permitir hacer que los sitios sean dinámicos. Que se puedan modificar (gracias al código que se ejecuta) y actualizar en base a las acciones del usuario sin necesidad de hacer requests al servidor ni de cargar nuevamente la página o un nuevo enlace.

Además va a generar requests a uno o varios servicios *backend*

Un servicio ***backend*** (BE) va a alojar toda la lógica de negocio de la aplicación y la capa de persistencia (las bases de datos con la información de los usuarios) suele devolver contenido en formato JSON.
Va a definir una interface (API) a la cual uno o varios clientes FE se van a comunicar.
Además, hay backends que sólo van a servir a otros servicios backend, los cuales a su vez pueden servir sólo a otros servicios backend... etc.

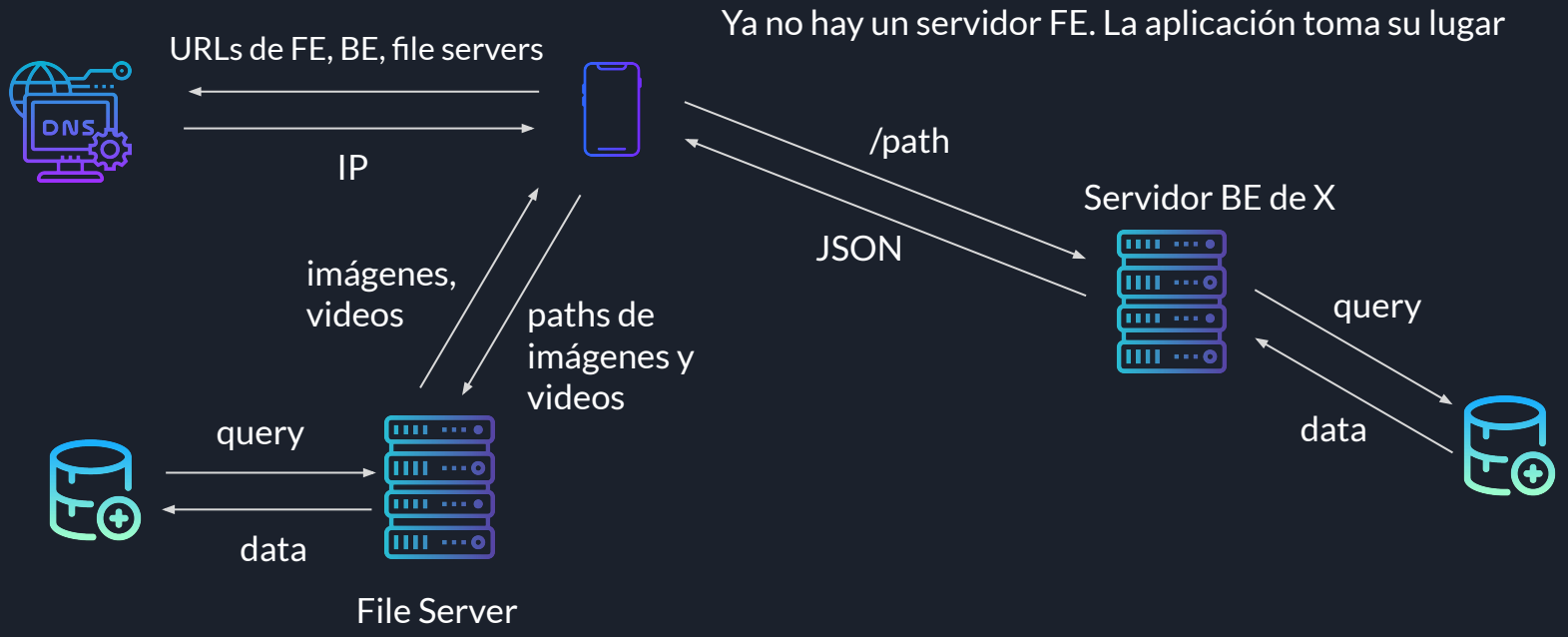
Cómo funciona?

Qué pasa desde que abro un navegador web y escribo una dirección en el buscador hasta que veo algo en la pantalla?



Y Mobile?

Cómo funcionan las apps de los celulares?



Herramientas

Además del navegador web que usemos, hay varias herramientas que nos permiten hacer requests a servidores y ver sus respuestas pero que tienen facilidades y opciones extra para facilitar el desarrollo.

cURL

Herramienta de CLI (Command Line Interface): ideal para hacer pruebas con servidores que responden con JSON ya que al ser una herramienta por consola, no visualiza el HTML como un navegador. Ya viene por defecto en la gran mayoría de las distribuciones de Linux y en macOS.

Postman

Herramienta GUI (Graphical UI): ideal para cualquier tipo de servidor y respuesta. Posee muchas facilidades de visualización, automatización, colecciones, sharing y mucho más.

Bruno

Mismas capacidades que Postman, pero se guarda todo localmente en archivos de texto plano que pueden ser versionables.

Otras 20 herramientas similares

Manejo de Proyectos

Herramientas

Hacer software es difícil. Ninguna pieza de software es igual a otra. Cada situación es distinta.

Al arrancar el desarrollo de un sistema no se sabe realmente cuál va a ser el "estado final". Las prioridades cambian en el camino, hay ideas que se descartan por otras y muchos cambios más.

Sumado a eso, es un trabajo colaborativo. Pocas veces un proyectos es desarrollado por una única persona.

Se necesitan herramientas dedicadas que nos ayuden a manejar y gestionar el desarrollo de proyectos.

Estas herramientas permitirán mantener un trackeo que de qué tareas se hicieron, se están haciendo y faltan por hacer. Además de identificar quién es el responsable de cada una. Setear deadlines, requerimientos mínimos, y muchas cosas más.

Todas las herramientas comparten estas días, pero todas lo hacen de maneras muy distintas.

Herramientas

Algunas opciones:

JIRA

Herramienta super completa que es usado mayoritariamente en empresas medianas/grandes. Es pago pero tiene muchas integraciones con otras herramientas que hacen que el trabajo sea más fácil (siempre y cuando se usen las otras herramientas de la misma compañía).

Trello

Herramienta gratuita y super simple de usar. Sus capacidades son algo limitadas, pero es ideal para usar en proyectos no muy grandes o relativamente simples. Es la que vamos a usar nosotros.

Github project

Herramienta integrada en el repositorio de GH. Relativamente simple de usar y bastante completa.

Bonus Tracks

Bonus Track: ad blockers

El funcionamiento de un adblocker es muy simple:

Todos los ads que se muestran en los sitios web, son provistos por un pequeño grupo de compañías. Para mostrar estos ads, el sitio simplemente debe agregar un poco de código JavaScript (provisto por estas mismas compañías) que se encargan de todo.

Estos servicios de ads mostrarán publicidad en base a muchos parámetros: la ubicación, historial de navegación y un par de cosas más.

Toda esta lógica ocurre en servidores de estas compañías. Nuestros navegadores hacen requests a estos servidores y en las respuestas están los ads que nosotros vemos.

Es por lo tanto muy fácil determinar cuáles son estas requests en base al dominio y URL usados (como dijimos, no son muchos distintos).

Los adblockers funcionan básicamente de dos maneras: bloquean requests salientes a estos dominios puntuales, haciendo que nunca se carguen los ads; o bloquean los módulos de Javascript para que el código en cuestión nunca se ejecute.

Bonus Track: cookies

Hay cierta información que a los web browsers les interesa guardar sobre el usuario que está interactuando con los distintos sitios web.

Esta información que se guardar se denomina como cookie. Cada cookie guarda una pequeña porción de información asociando una clave (o nombre de la cookie) a un valor que puede ser cualquier cosa.

Generalmente las cookies se utilizan para guardar información de la sesión del usuario (si necesita loguearse), para personalizar ciertas cuestiones del sitio web (estilos, colores, etc.) y para trackear las acciones del usuario que toma en el sitio.

Location from IP

Muchos sitios detectan automáticamente desde donde uno está haciendo la request en base a la IP pública que tenemos asignada (la IP de quien hace la request es parte de la información que se envía a los servidores en cada request).

Hay muchos sitios en Internet con esta información, por ejemplo ip2location que permite descargar CSVs con la info o ip-API que brinda una API publica para consultar la información de una IP particular.

VPN

Fin