

# INTRO AL DESARROLLO FULL STACK

*Charly Cimino*

# Intro al Desarrollo Full Stack

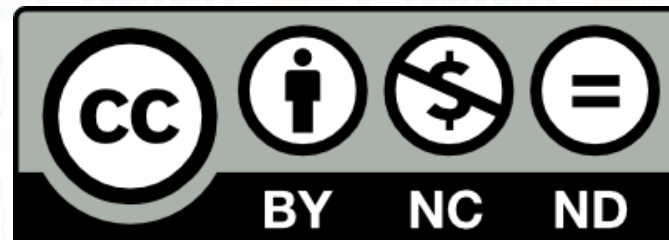
Charly Cimino

Este documento se encuentra bajo Licencia Creative Commons 4.0 Internacional (CC BY-NC-ND 4.0). Usted es libre para:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

Bajo los siguientes términos:

- **Atribución** — Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.
- **No Comercial** — Usted no puede hacer uso del material con fines comerciales.
- **Sin Derivar** — Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted no podrá distribuir el material modificado.



# Redes de computadoras

Conjunto de dispositivos y software conectados entre sí, que envían y reciben datos a través de diferentes medios de transmisión, con la finalidad de compartir información, recursos y ofrecer servicios.

## Clasificación por alcance



Red de área local



Red de área metropolitana



Red de área amplia

<https://pc-solucion.es/wp-content/uploads/2018/03/esquemas-de-redes-lan-man-wan.jpg>

## Medios de transmisión



### Alámbricos

- Cables de par trenzado
- Cables coaxial
- Cables de fibra óptica

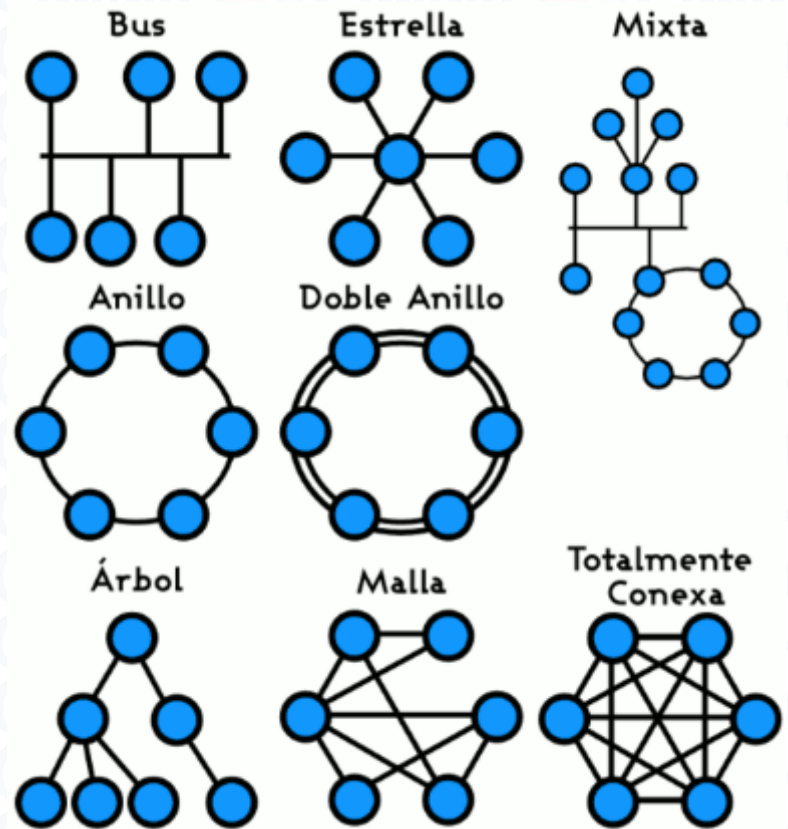


### Inalámbricos

- Radiofrecuencia
- Microondas
- Infrarrojos

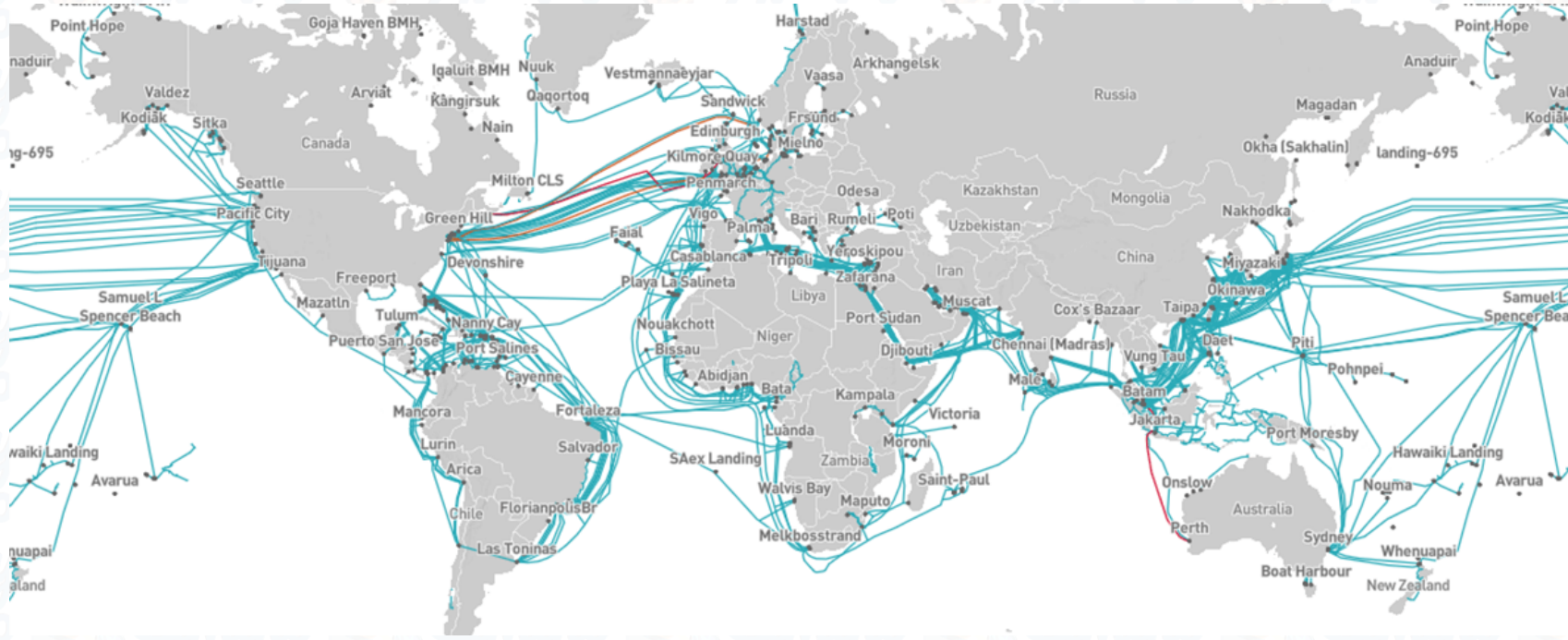
[https://es.wikipedia.org/wiki/Red\\_de\\_computadoras](https://es.wikipedia.org/wiki/Red_de_computadoras)

## Topología de redes



# Internet

Internet es la forma de referirse a la gran red de redes de computadoras interconectadas a nivel mundial.



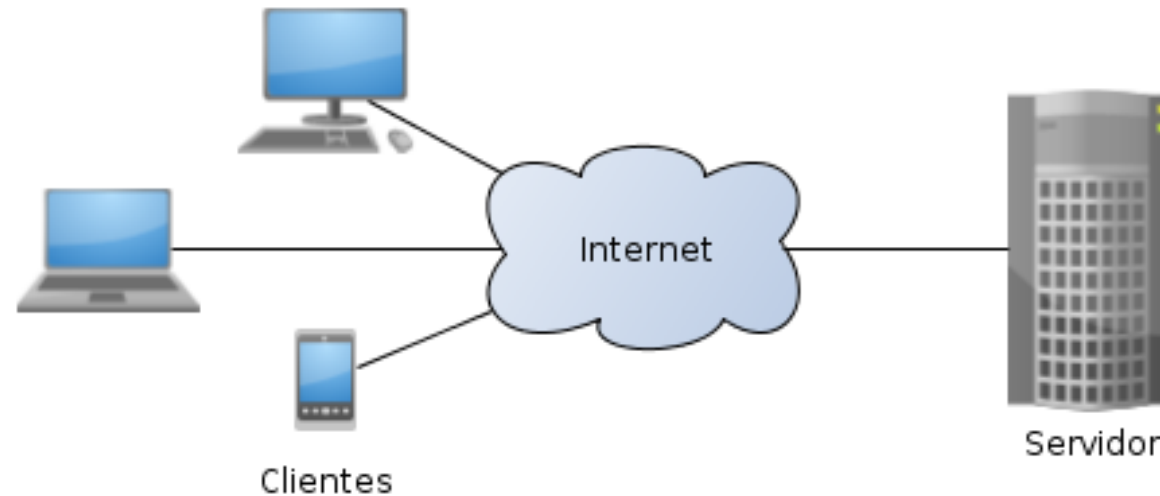
La infraestructura de cable submarinos y otros medios de transmisión (como el satelital) permiten la comunicación desde cualquier lugar del mundo

## ¿Qué es Intranet?

Es una red informática que utiliza la tecnología de Internet para compartir información, pero de forma interna dentro de una organización.

# Arquitectura Cliente-Servidor

A través de diferentes dispositivos (*tablets, smartphones, Smart TVs, laptops, computadoras de escritorio, etc*) llamados **clientes**, establecemos **peticiones (requests)** a un servidor.



El **servidor** recibe la petición, la gestiona y devuelve una **respuesta (response)**.

**WWW**  
(World Wide Web)

**Correo electrónico**

**VoIP**  
(Voz sobre IP)

**RSS**  
(Really Simple Syndication)

**FTP**  
(File Transfer Protocol)

Algunos servicios que corren sobre Internet

# ¿Qué es la web?

**World Wide Web (WWW)**, comúnmente abreviado como **web**, es un servicio que permite el intercambio de recursos a través de enlaces (**links**).

## ¿Cuáles son esos recursos?

Suelen ser páginas web, en formato HTML, aunque puede ser cualquier otro contenido (imágenes, audios, videos, etc).

## ¿Cómo se ven esos recursos?

A través de navegadores web (**browsers**).

## ¿Cómo se comparten esos recursos?

A través de un protocolo llamado HTTP.

## ¿Quién retorna esos recursos?

Una computadora con software instalado que permita comportarse como un servidor web.



Tim Berners-Lee

Científico de la computación inglés

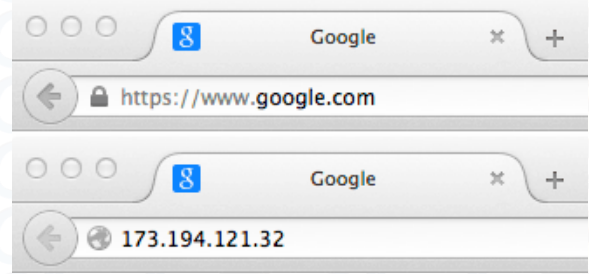
(1955-Actualidad)

Padre de la *World Wide Web*.

Estableció la primera comunicación cliente-servidor con HTTP en diciembre de 1990.

Dirige desde 2007 el *World Wide Web Consortium (W3C)*, el cual desarrolla y mantiene los estándares de la web.

# ¿Cómo funciona la web?



Se escribe una dirección web en un *browser* (como <http://www.google.com>) o se hace click en un enlace.

El *browser* recibe esos paquetes y los organiza para formar el código fuente de la página web, lo interpreta y muestra el resultado.

El *browser* envía una petición a un servidor DNS y obtiene la dirección IP real del servidor donde reside el sitio web.

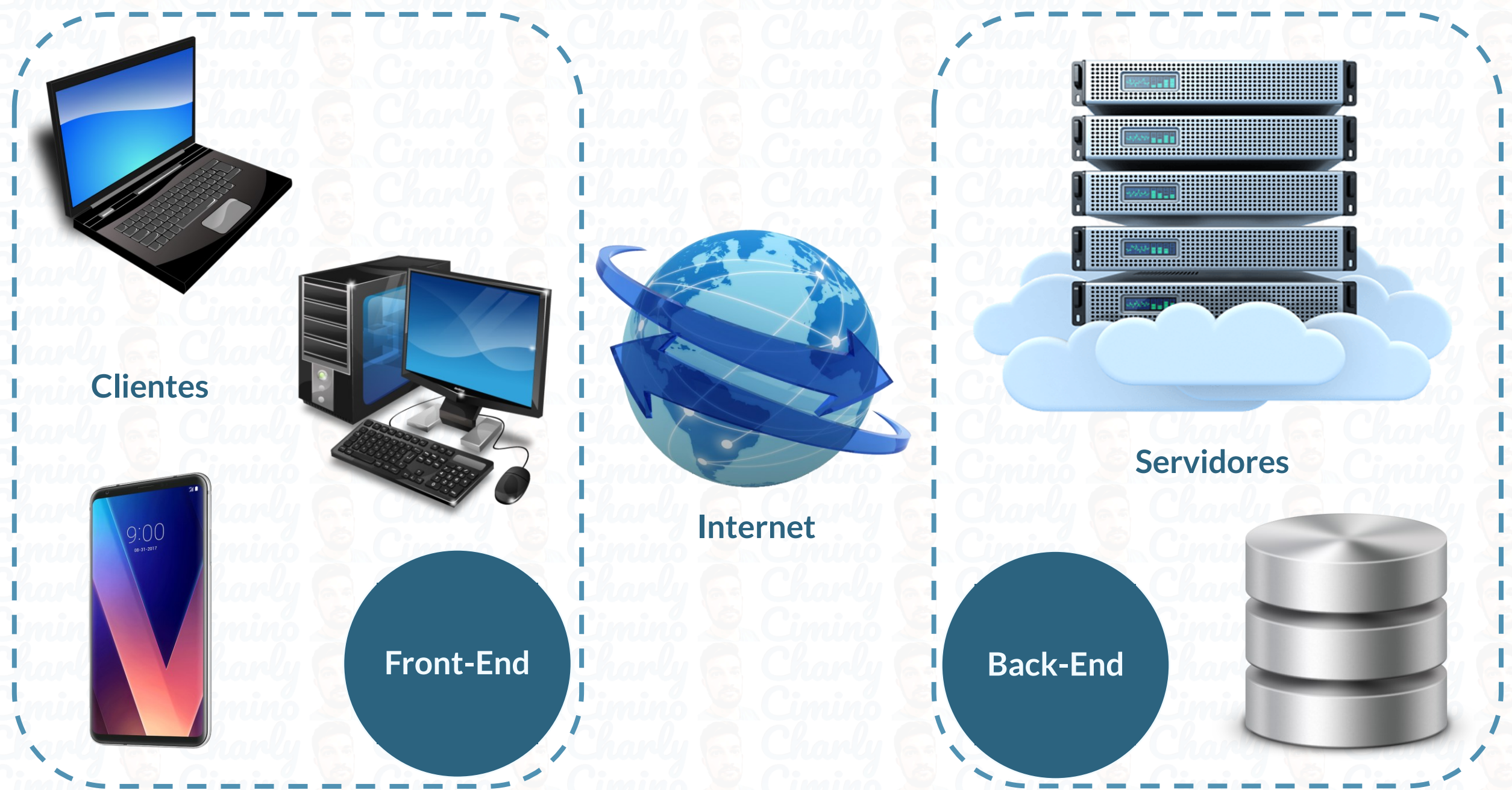
En lugar de recordar una dirección IP (por ejemplo, **173.194.121.32**), es más fácil recordar un nombre de dominio (por ejemplo, **google.com**).  
Ese es el trabajo que hacen los servidores **DNS (Sistema de Nombres de Dominio)**: reemplazar el nombre de dominio por la dirección IP donde se aloja el recurso web.

Los datos se envían a través de la web como miles de trozos pequeños (paquetes), permitiendo que muchos usuarios puedan descargar la misma página web al mismo tiempo. Si los sitios web fueran enviados como grandes trozos, sólo un usuario podría descargarlos a la vez, lo que volvería a la web muy ineficiente.

El servidor evalúa la petición y, si la aprueba, envía una respuesta exitosa en forma de paquetes de datos que conforman la página web.

El *browser* envía una petición al servidor web, pidiendo una copia de la página web que se desea visitar.

Esta solicitud, y todos los datos enviados entre el cliente y el servidor, se realizan a través de una conexión a Internet usando los protocolos **TCP/IP (Protocolo de control de transmisión/Protocolo de Internet)**.





# “Desarrollador Full-Stack”



Programador con conocimiento de tecnologías del lado cliente (*front-end*) y lado servidor (*back-end*).

# Front-End

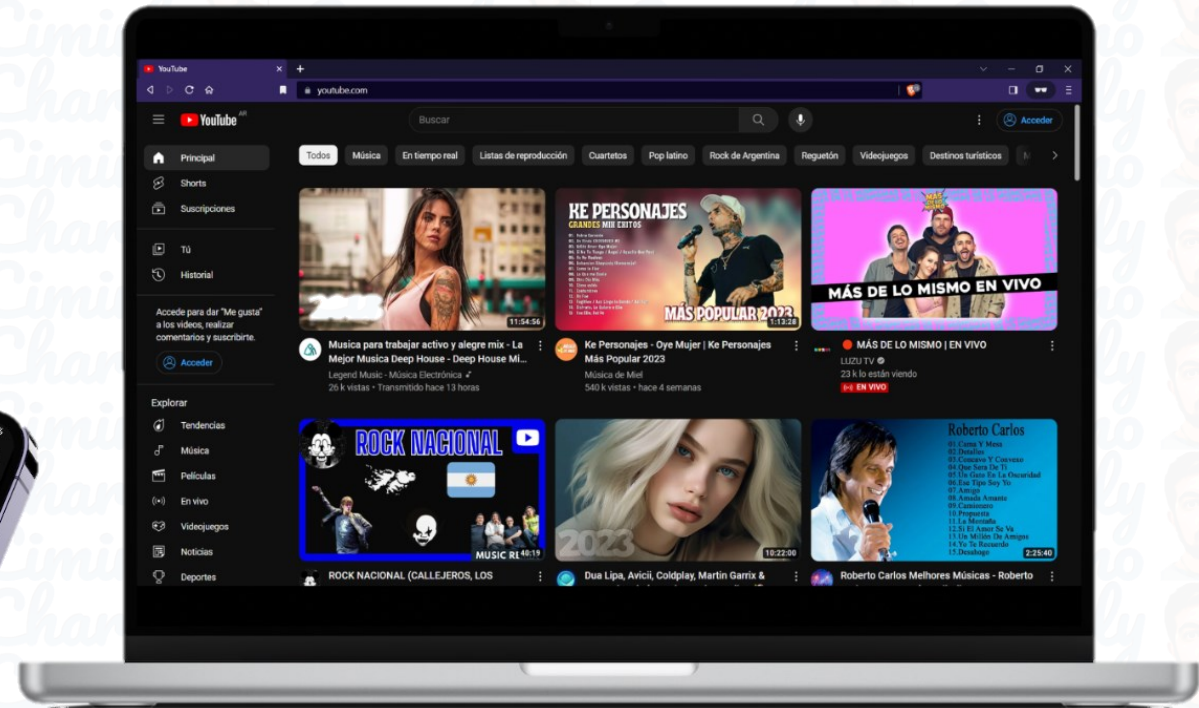
Desktop

Web

Mobile

Rama del desarrollo que se encarga de la parte visual y las interacciones del usuario con nuestra aplicación.

# Misma app, diferentes clientes



# Front-End

Por convención, se llama “**Desarrollador Front-End**” a aquel que domina las tecnologías web del lado cliente.

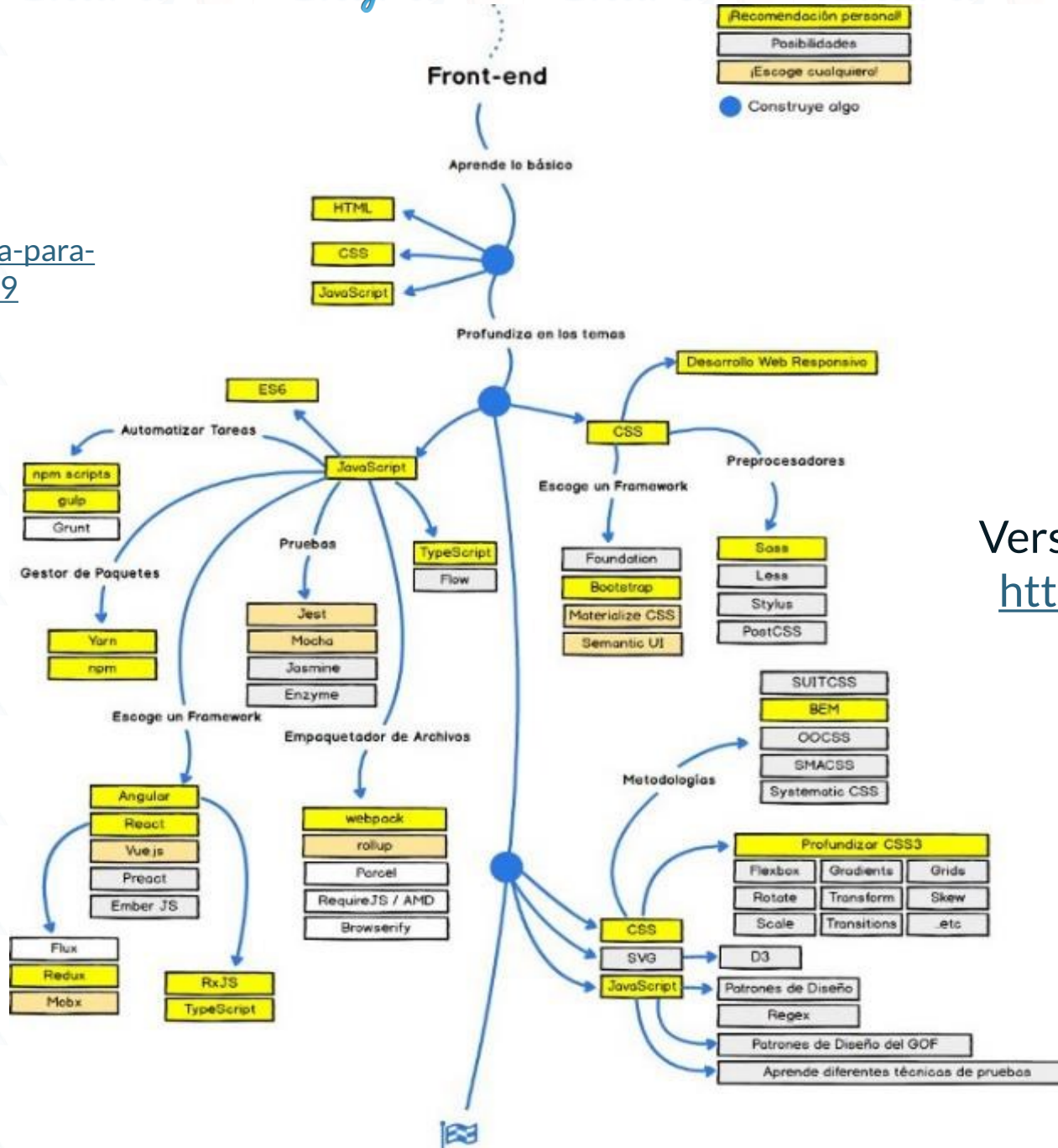


Lenguajes fundamentales

Fuente:  
<https://medium.com/@crisdulcey/ruta-para-ser-desarrollador-web-e1a891374519>

Required for any path

- Git - Version Control
- Basic Terminal Usage
- Data Structures & Algorithms
- SOLID, KISS, YAGNI
- GitHub
- Licenses
- Semantic Versioning
- SSH
- HTTP/HTTPS and APIs
- Design Patterns
- Character Encodings



Versión más completa y actual:  
<https://roadmap.sh/frontend>

# Back-End

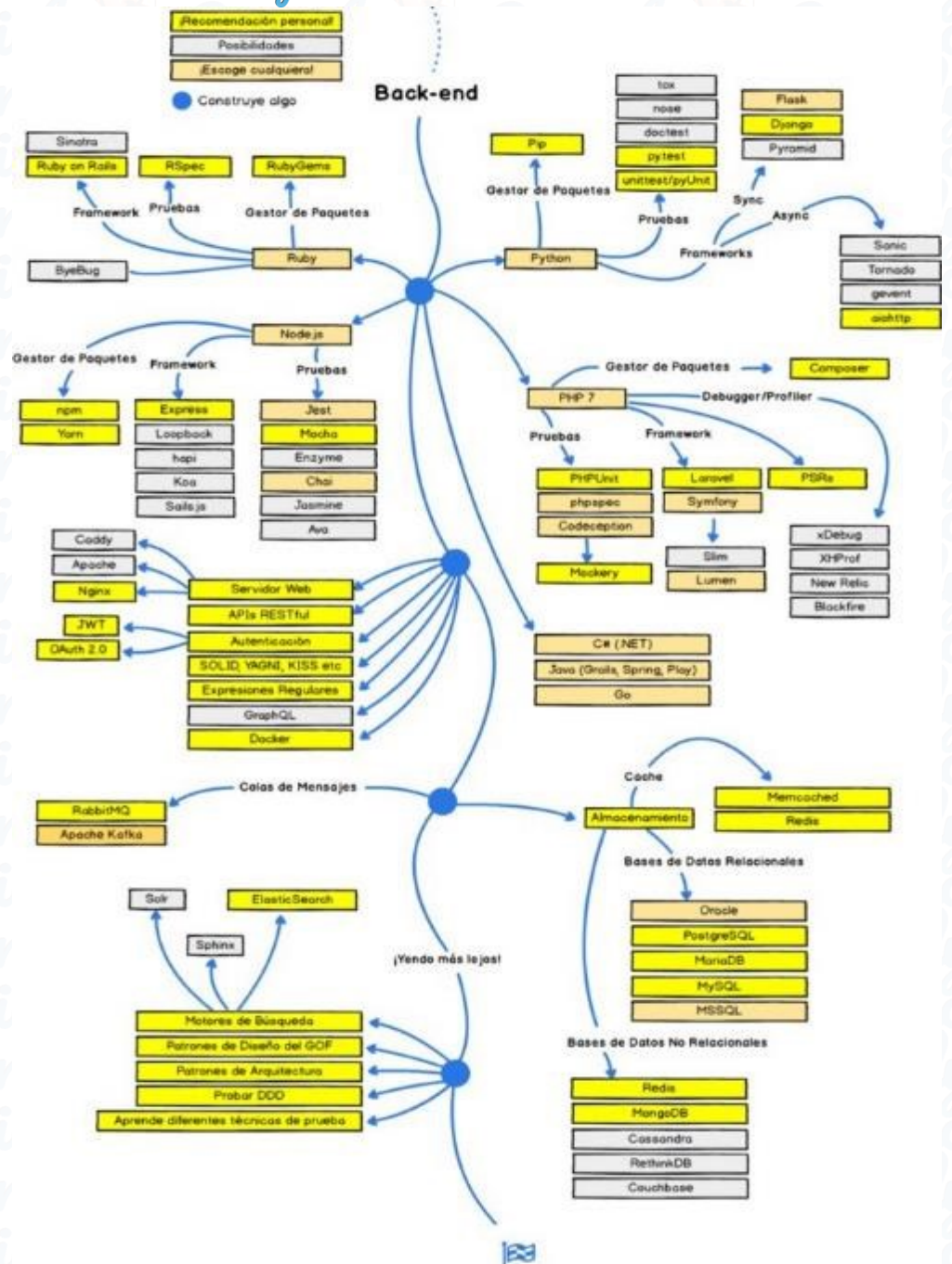
Rama del desarrollo que se encarga de la parte funcional de la aplicación que corre del lado del servidor.



Fuente: <https://medium.com/@cristdulcey/ruta-para-ser-desarrollador-web-e1a891374519>

Required for any path

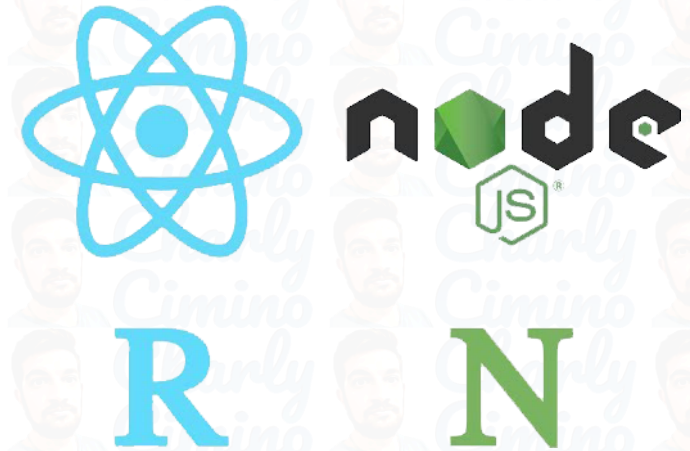
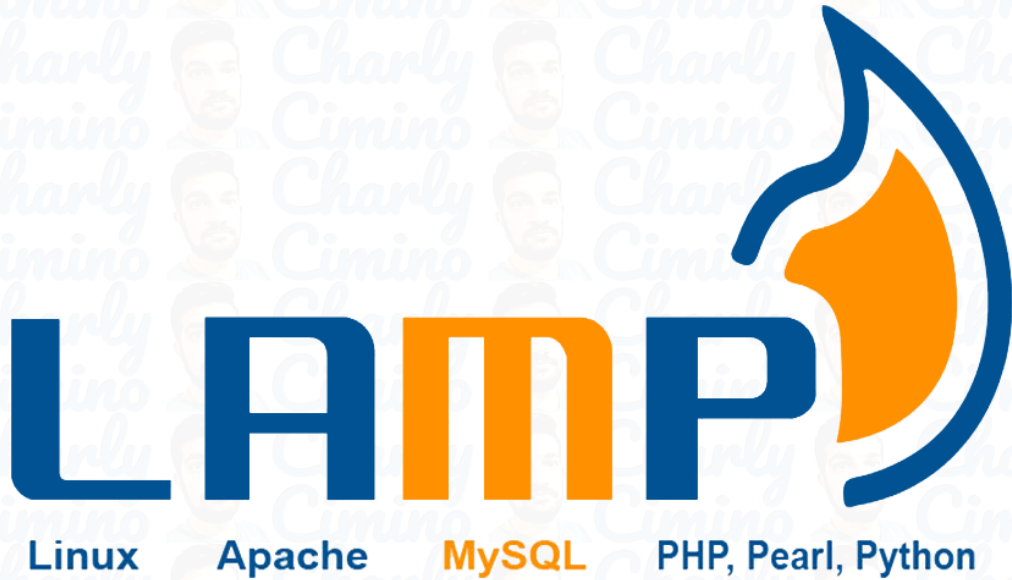
- Git - Version Control
- Basic Terminal Usage
- Data Structures & Algorithms
- SOLID, KISS, YAGNI
- GitHub
- Licenses
- Semantic Versioning
- SSH
- HTTP/HTTPS and APIs
- Design Patterns
- Character Encodings



Versión más completa y actual: <https://roadmap.sh/backend>

# Stack tecnológico

Conjunto de tecnologías para desarrollar y ejecutar una aplicación.





M

MongoDB

Base de datos NoSQL que almacena documentos en formato JSON.

E

Express

Web framework basado en Node para el desarrollo de apps web y APIs REST.

?

Angular  
(Google)

Vue.js  
(Evan You)

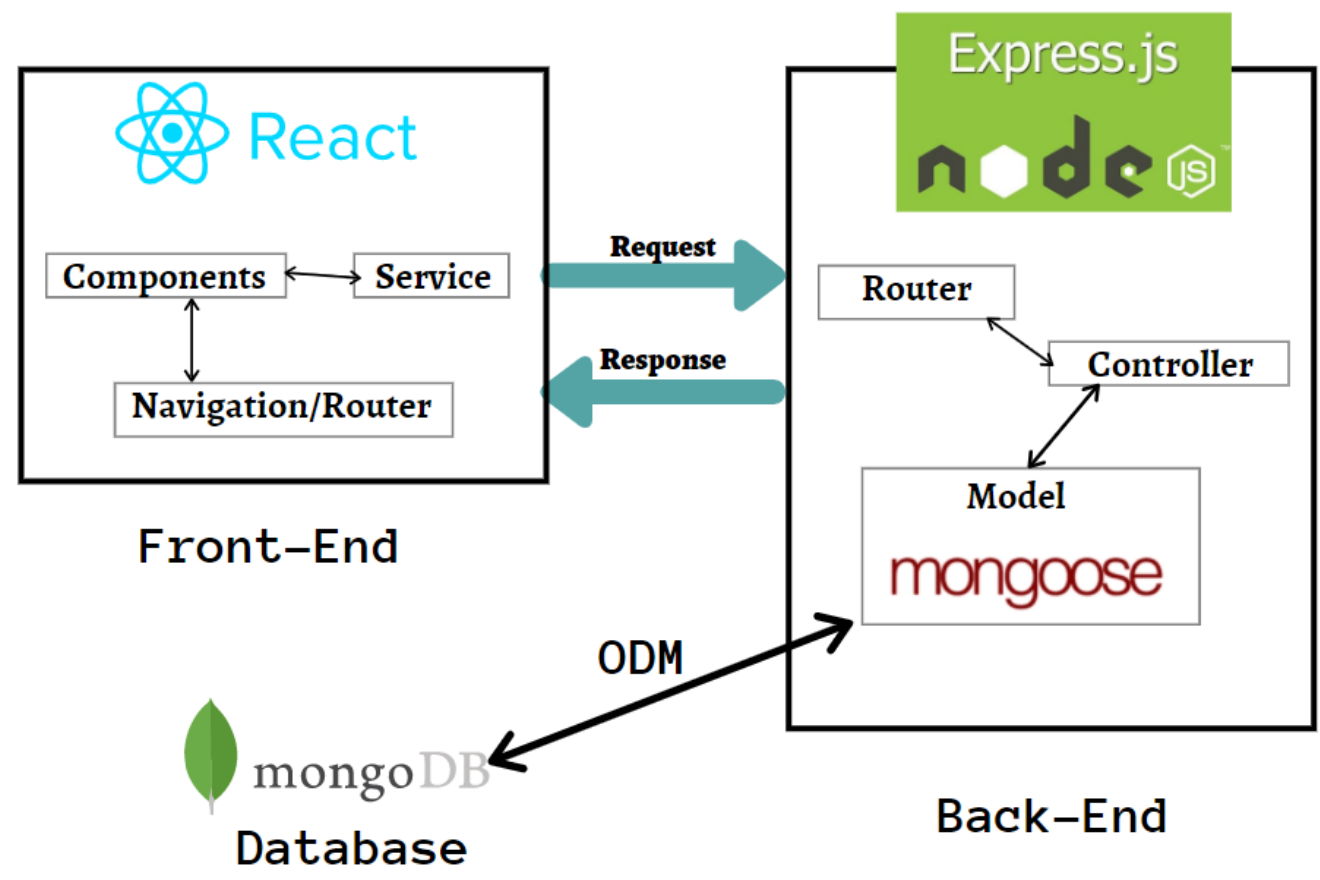
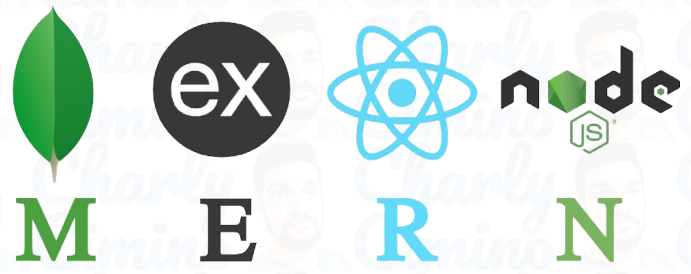
React  
(Facebook)

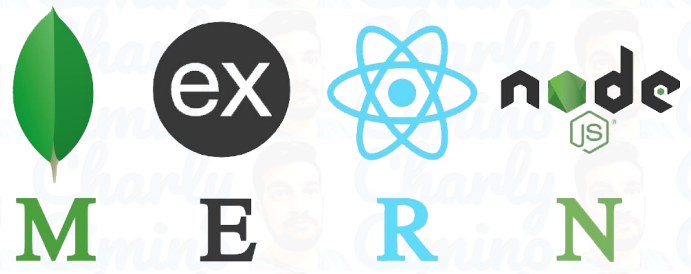
Librerías/Frameworks de código abierto para crear interfaces de usuario y SPAs.

N

NodeJS

Entorno de ejecución JS orientado a eventos asíncronos basado en el motor V8 de Chrome.





- Óptimo para apps real-time.
- Más rápido que LAMP.
- Un solo lenguaje para todo: **JavaScript**.

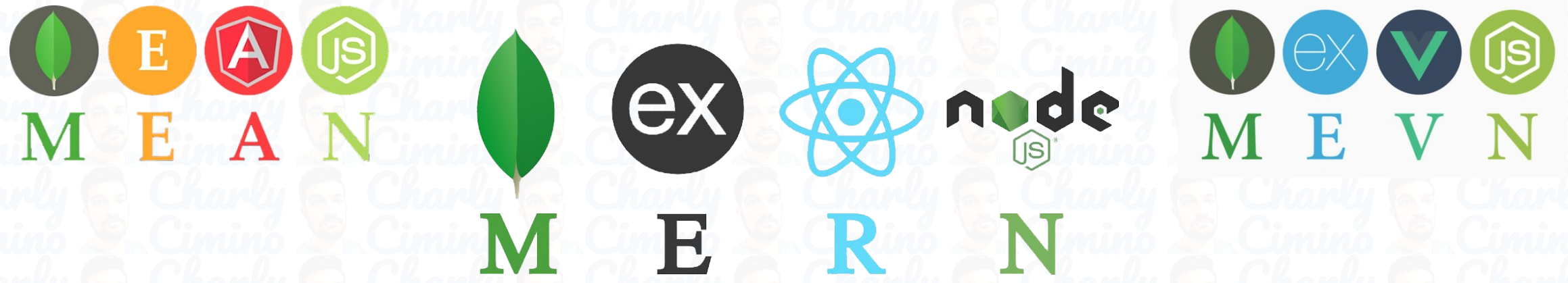
- Tecnología en auge pero aún por madurar.
- Mayor curva de aprendizaje.
- Las BD No-SQL no ofrecen las mismas garantías que los DBMS tradicionales.



- Tecnología madura.
- Bien documentada.
- Gran comunidad.

- Lenguajes diferentes para cada aspecto (servidor, cliente, base de datos).
- Más lento que MERN.
- No muy óptimo para apps real-time.

# Para tener éxito en ...



... debes aprender:



# FIN DE LA PRESENTACIÓN

Encontrá más como estas en mi [sitio web](#).