



Desarrollo Web Fullstack

- ¿Qué es el desarrollo web full stack?
- Enumera 2 o más cualidades de un desarrollador web full stack.

Para trabajar en equipos!!

El desarrollo web full stack permite crear **aplicaciones web dinámicas**. Esto se logra en base a los **estándares web** definidos por la w3c y utilizando tecnologías web que pueden variar dependiendo del stack de desarrollo.

Organismo internacional fundado por Tim Berners-Lee que vela por el buen funcionamiento de la web y define los estándares de la web.

<https://www.w3c.es/>



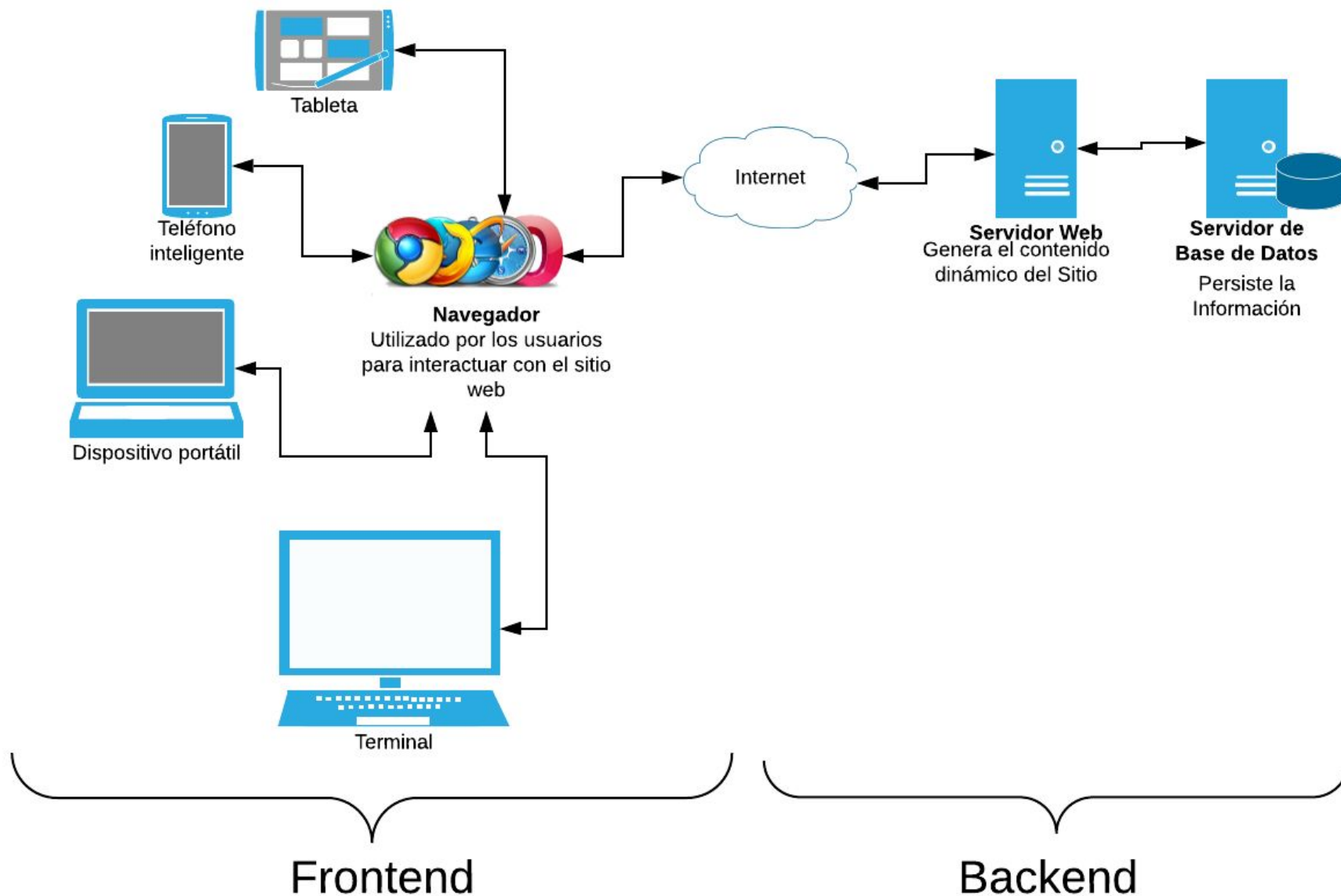
Según la definición previa el desarrollo web full stack tiene por objeto la creación de aplicaciones web dinámicas ¿Qué entendemos por aplicación web **dinámica**? ¿En qué se diferencia de una aplicación web **estática**?

Para debatir en clase...

Las *aplicaciones web* son un tipo especial de aplicaciones cliente/servidor donde tanto el cliente, como el servidor y el protocolo mediante el que se comunican (HTTP) están estandarizados.

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre ordenadores. HTTP se sitúa en el nivel 7 (aplicación) del modelo OSI.

Arquitectura web - 3 niveles



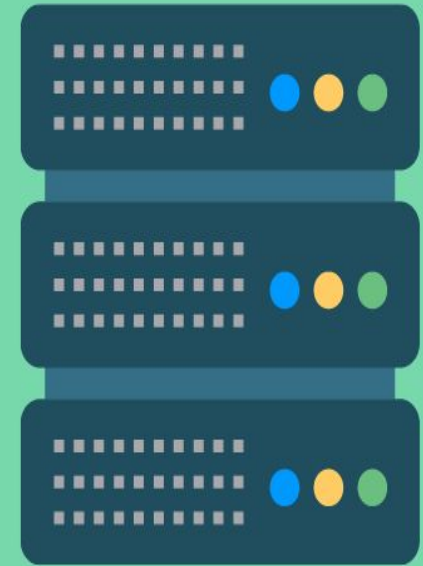
En las *aplicaciones web*, por lo general, suelen distinguirse 3 niveles:

- el **nivel superior** interacciona con el usuario (normalmente un **navegador**).
- el **nivel inferior** proporciona los datos (**base de datos**).
- el **nivel intermedio** procesa los datos (**servidor web**).

Frontend



FRONT END



BACK END

Frontend

Es la parte del sistema que interactúa con el usuario e incluye la **lógica de presentación**. Es decir, la interfaz gráfica y su funcionalidad, teniendo en cuenta los aspectos de usabilidad, de accesibilidad, de web responsive, entre otros.

Las tecnologías básicas del frontend son:

- HTML5 (Lenguaje de marcado de doc. web)
- CSS3 (Hojas de estilo en cascada)
- JavaScript (lenguaje de programación del lado del cliente)



HTML, siglas en inglés de HyperText Markup Language.

Es un lenguaje de marcado de los contenidos de un sitio web, es decir, para designar la función de cada elemento dentro de la página: titulares, párrafos, listas, tablas, etc. Es el esqueleto de la web y la base de todo el frontend.

- `<html>`: define el inicio del documento HTML
- `<script>`: incrusta un script en una web
- `<head>`: define la cabecera del documento HTML
- `<title>`: define el título de la página
- `<style>`: para colocar el estilo interno de la página
- `<body>`: define el contenido principal o cuerpo del documento
- `<table><tr><td>Contenido de una celda</td></tr></table>`
- otros...

CSS, en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado, escrito en un lenguaje de marcado.

Es un lenguaje de hojas de estilo creado para controlar la presentación de la página definiendo colores, tamaños, tipos de letras, posiciones, espaciados, etc.

En pocas palabras, CSS define la estética, es decir que se vea bonito y todo lo concerniente al web responsive, por ejemplo que la ventana se adapte a los diferentes dispositivos.

JavaScript, es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, débilmente tipado y dinámico. Permite mejoras en la interfaz de usuario y páginas web dinámicas.

Fundamentalmente se encarga del comportamiento de una página web y de la interactividad con el usuario.

Backend

Es la parte del sistema “no visible por el usuario”, “del lado del servidor” e incluye la **lógica de negocio y de base de datos**. Es decir, la manipulación de los datos, teniendo en cuenta aspectos tales como el rendimiento, velocidad, seguridad, etc.

Las tecnologías definidas para el backend son mucho más variadas.

- Apache, IIS, etc. (Servidor Web).
- MariaDB, MySQL, MongoDB, Microsoft SQL, etc. (Bases de Datos).
- PHP, C#, Python, etc. (Lenguaje de programación del lado del servidor)

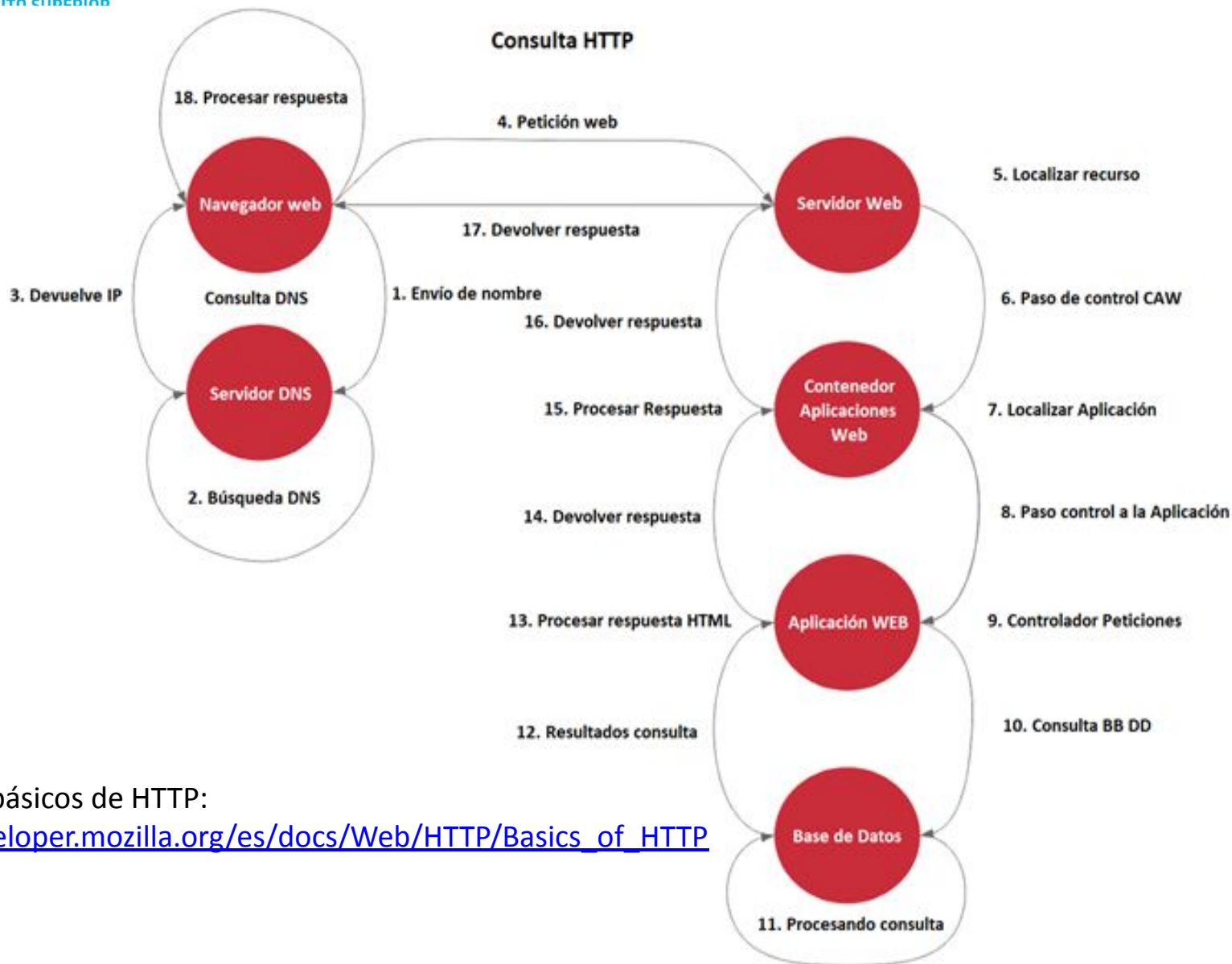
Elementos de la arquitectura web

Cliente Web, es el navegador web. Ej. Chrome, Safari, Firefox, etc.

Servidor DNS, Se ocupa de la administración del espacio de nombres de dominio.

Servidor Web o servidor HTTP, es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

Contenedor de aplicaciones Web, módulo que permite la ejecución de aplicaciones web. Por ejemplo, módulo PHP o Python del Servidor Web. Componente ASP o ASPX de IIS. Servidor o Contenedor de Aplicaciones Web Java: Tomcat, Weblogic, Websphere, JBoss, Geronimo, etc.



Conceptos básicos de HTTP:

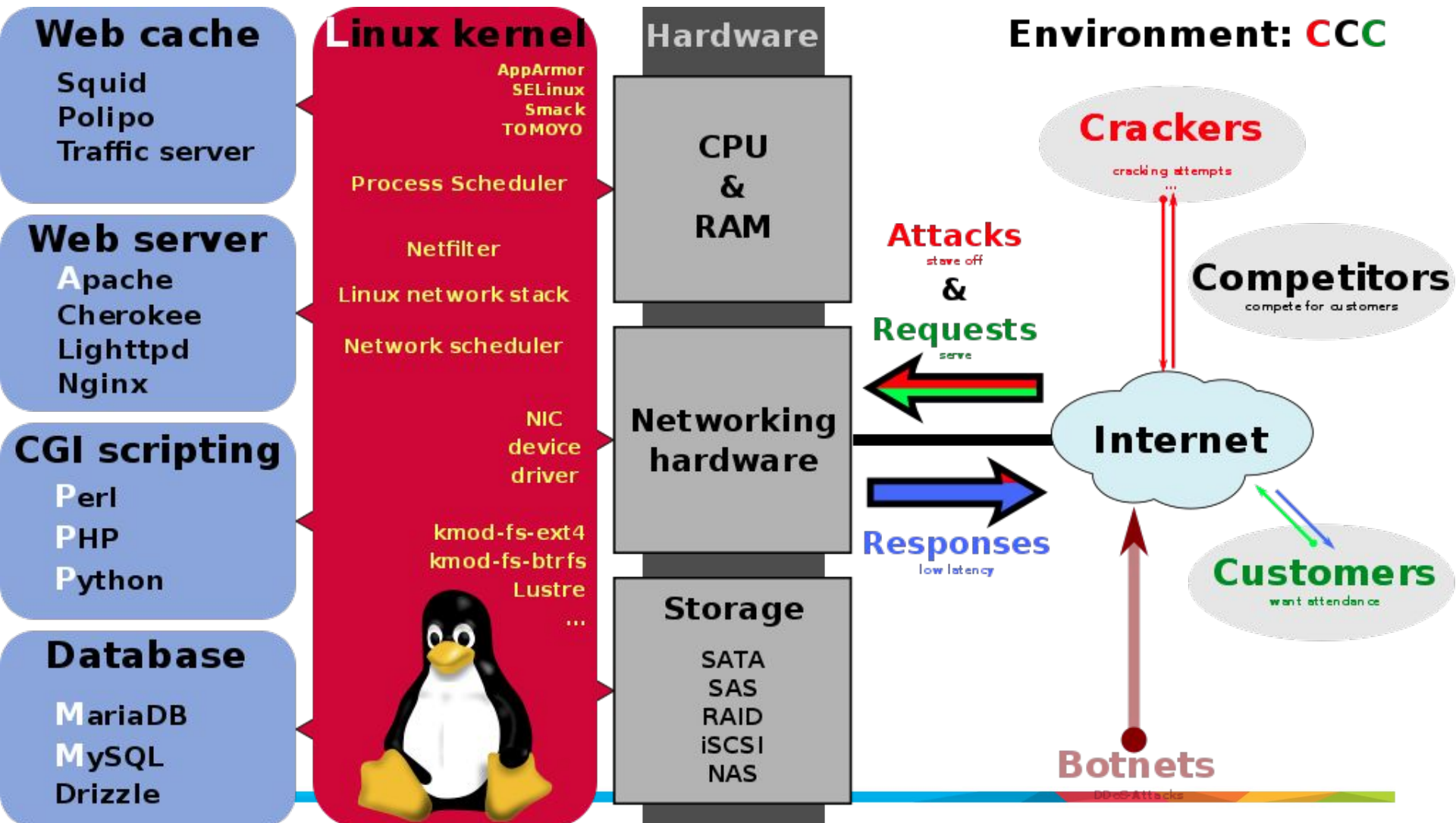
https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP

Stack populares

- **LAMP** stack: Linux - Apache - MySQL - PHP
- **LEMP** stack: Linux - Nginx - MySQL - PHP
- **MEAN** stack: MongoDB - Express - AngularJS - Node.js
- **Django** stack: Python - Django - MySQL
- **Ruby on Rails**: Ruby - SQLite - Rails
- **.NET**: W. Server - IIS - SQL Server - C#

Para más información consulta: https://www.w3schools.com/whatis/whatis_fullstack.asp

LAMP Stack



MEAN Stack

BASE DE DATOS

SERVIDOR con
una API REST

FRONTEND

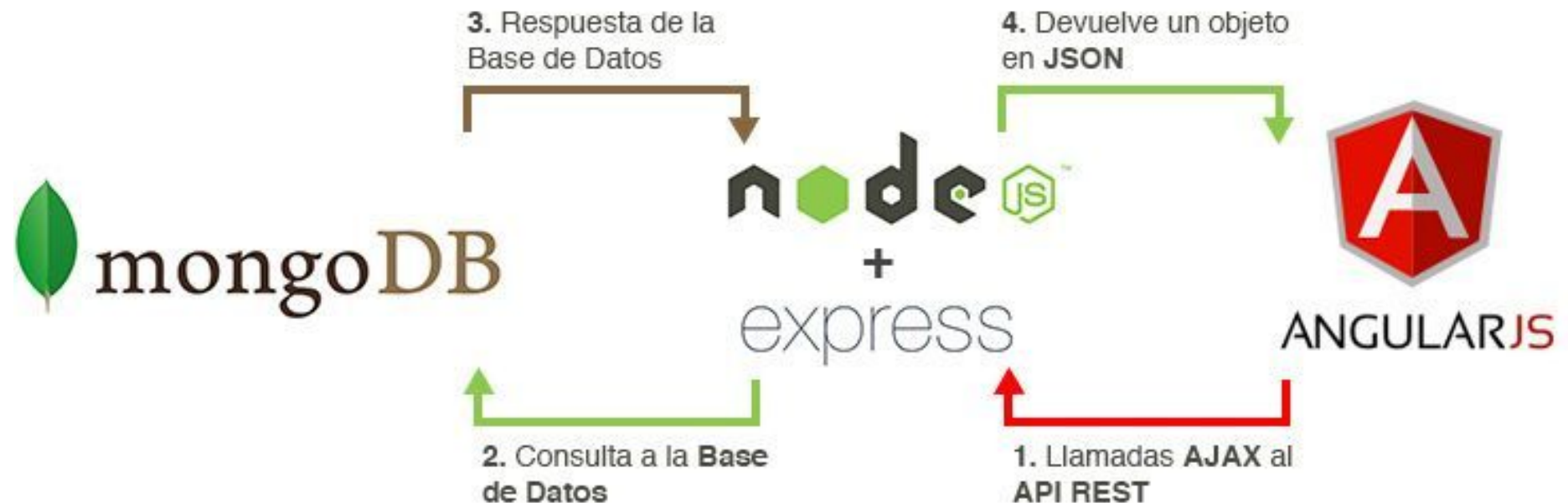


Imagen de Arquitectura MEAN, extraída de :

<https://developer.ibm.com/es/patterns/create-backend-for-frontend-application-architecture/>

Stacks, Tecnologías y Frameworks

Front End Technologies:

HTML/HTML5 CSS JavaScript
PostgreSQL JQuery

Front End Frameworks:

Angular.Js Backbone.Js
Ember.Js Polymer.Js

Back End Technologies:

Ruby On Rails Node.Js PHP
.NET

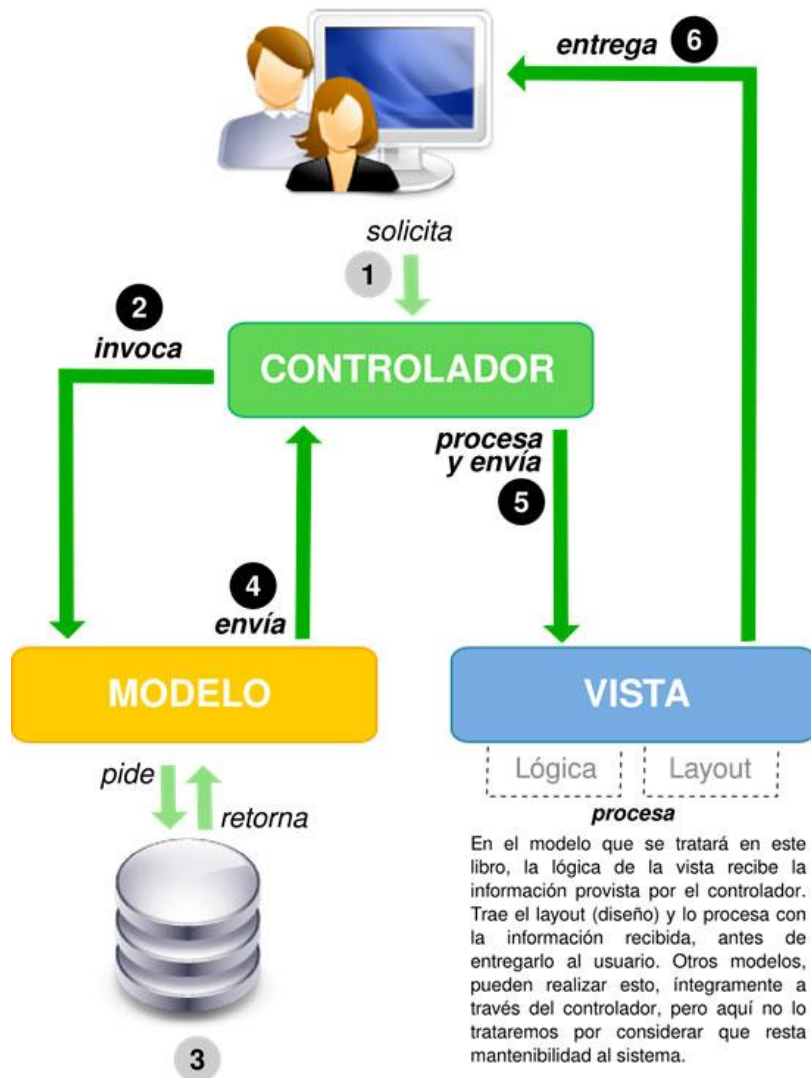
Back End Frameworks:

Rails Express ASP.NET
CodeIgniter And More

Database Development Technologies:

MySQL MongoDB Oracle
Microsoft SQL Server And More

Arquitectura MVC



Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: el modelo, la vista y el controlador.



<https://kahoot.it/>

Para pensar...

¿Existen diferencias entre los términos: pagina web, sitio web y aplicación web?

Para debatir en clase...

