



## **Clase 1**

# **WEB. Arquitectura, tecnologías y herramientas**



## **Arquitectura cliente-servidor**

- **Cliente:**
  - Consumidor de recursos externos
- **Servidor:**
  - Comparte recursos externos
- **Mayoría de servicios de internet:**
  - Email, Web, DNS, etc.
  - Son cliente-servidor

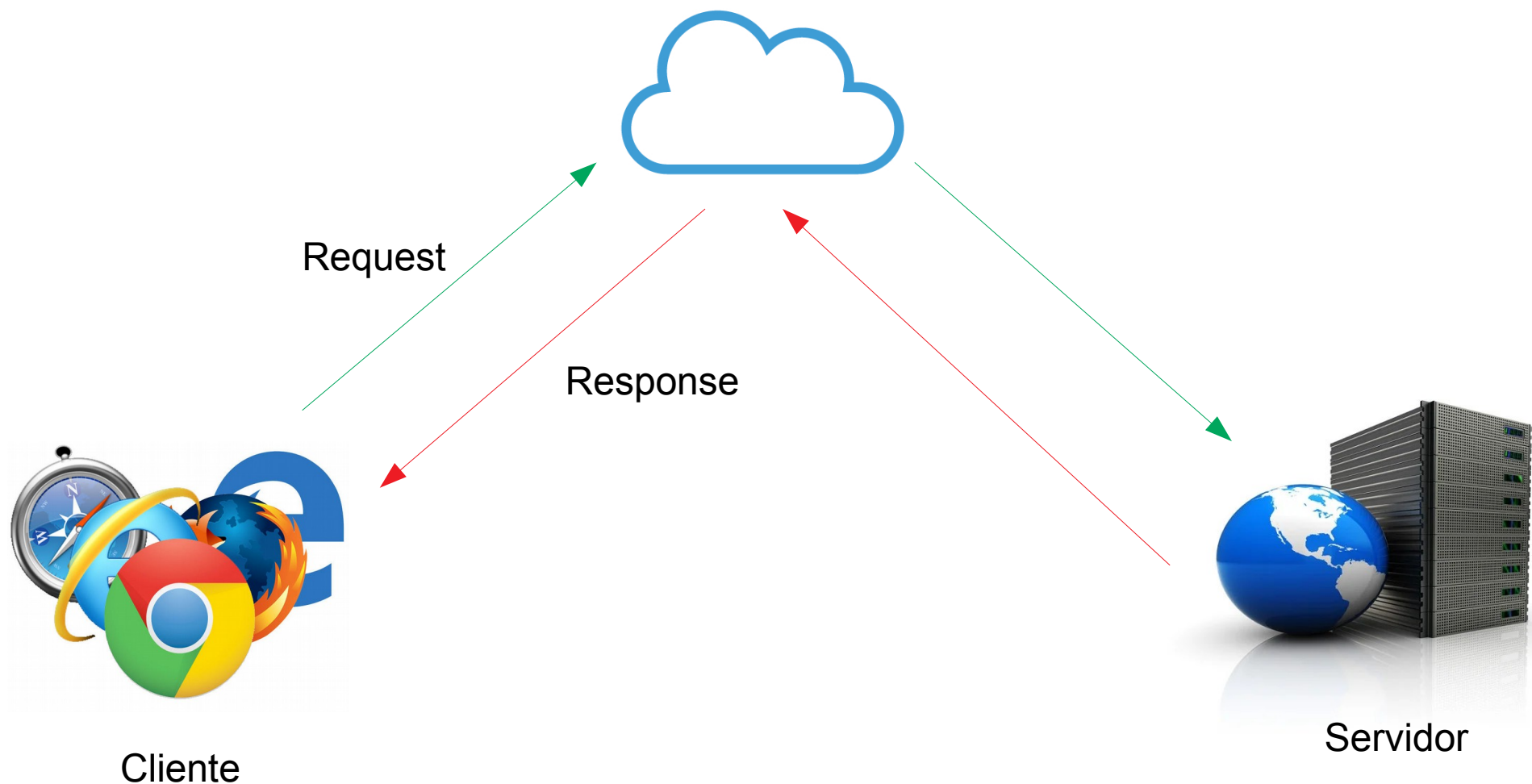


## **Arquitectura cliente-servidor**

- **Protocolo TCP/IP**
  - **Cliente/Servidor**
  - **Alguien escucha una comunicación**
  - **Alguien inicia una comunicación**
- **Protocolo HTTP**
  - **Sobre TCP/IP**

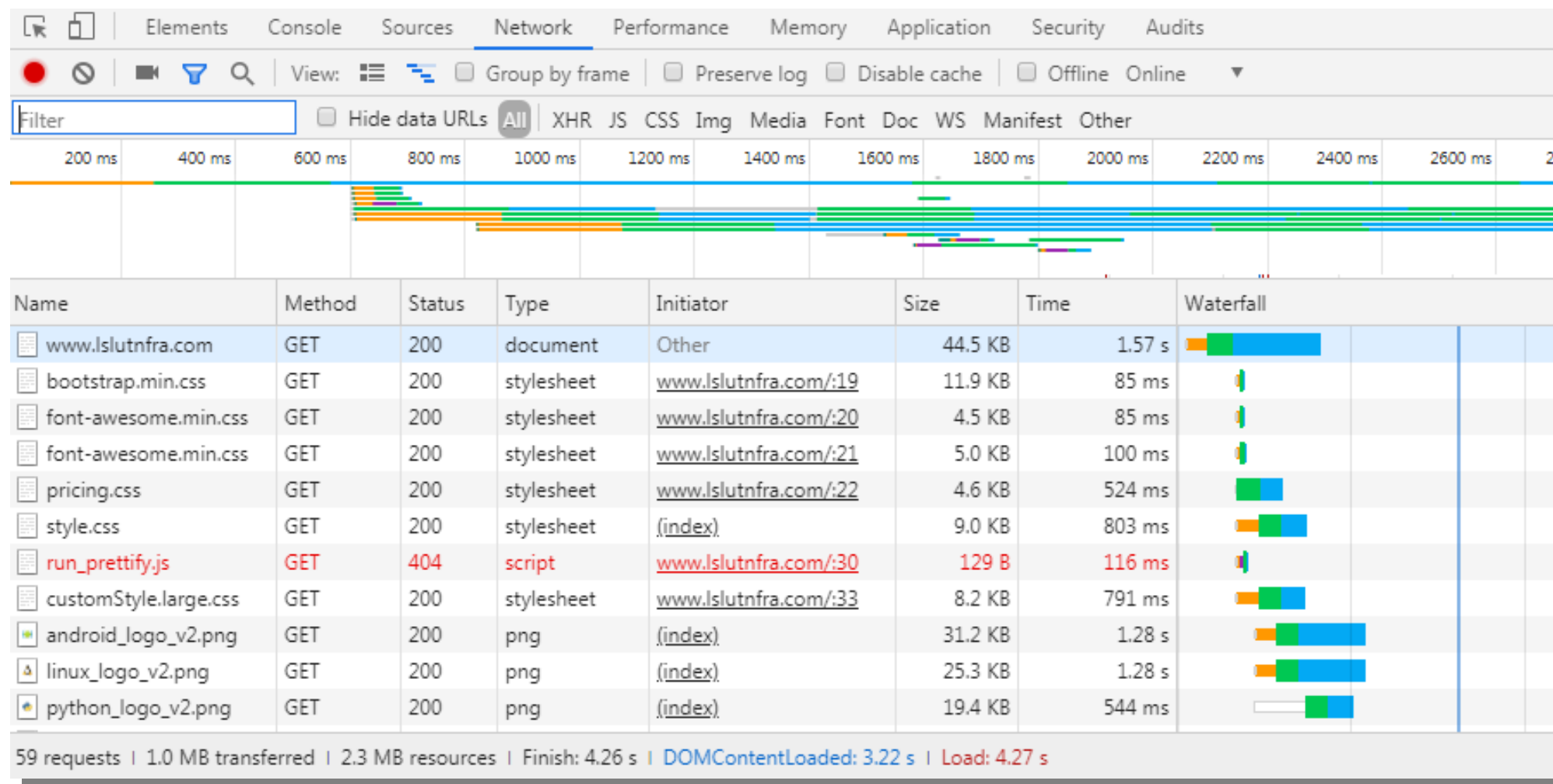


## Arquitectura cliente-servidor





## Arquitectura cliente-servidor





## Arquitectura cliente-servidor

- Protocolo HTTP: Hypertext Transfer Protocol
  - Protocolo de request-response
  - Sus recursos se identifican con URLs
  - Posee un *Header*
  - Se transmite texto plano



## **Arquitectura cliente-servidor**

- **Request:**
  - Dirección (URL).
  - Versión del protocolo.
  - Campos (Por Ej. "Accept-Language: en").
  - Cuerpo de mensaje (opcional).
  - Método (GET, POST, DELETE, etc.)
    - Definen la acción a realizar
    - Las acciones se definen en el server.



## **Arquitectura cliente-servidor**

- **Response:**
  - Dirección (URL).
  - Versión del protocolo.
  - Campo estado (200,404, etc.)
  - Campos (por Ej. "Content-Type: text/html")
  - Cuerpo de mensaje (opcional).



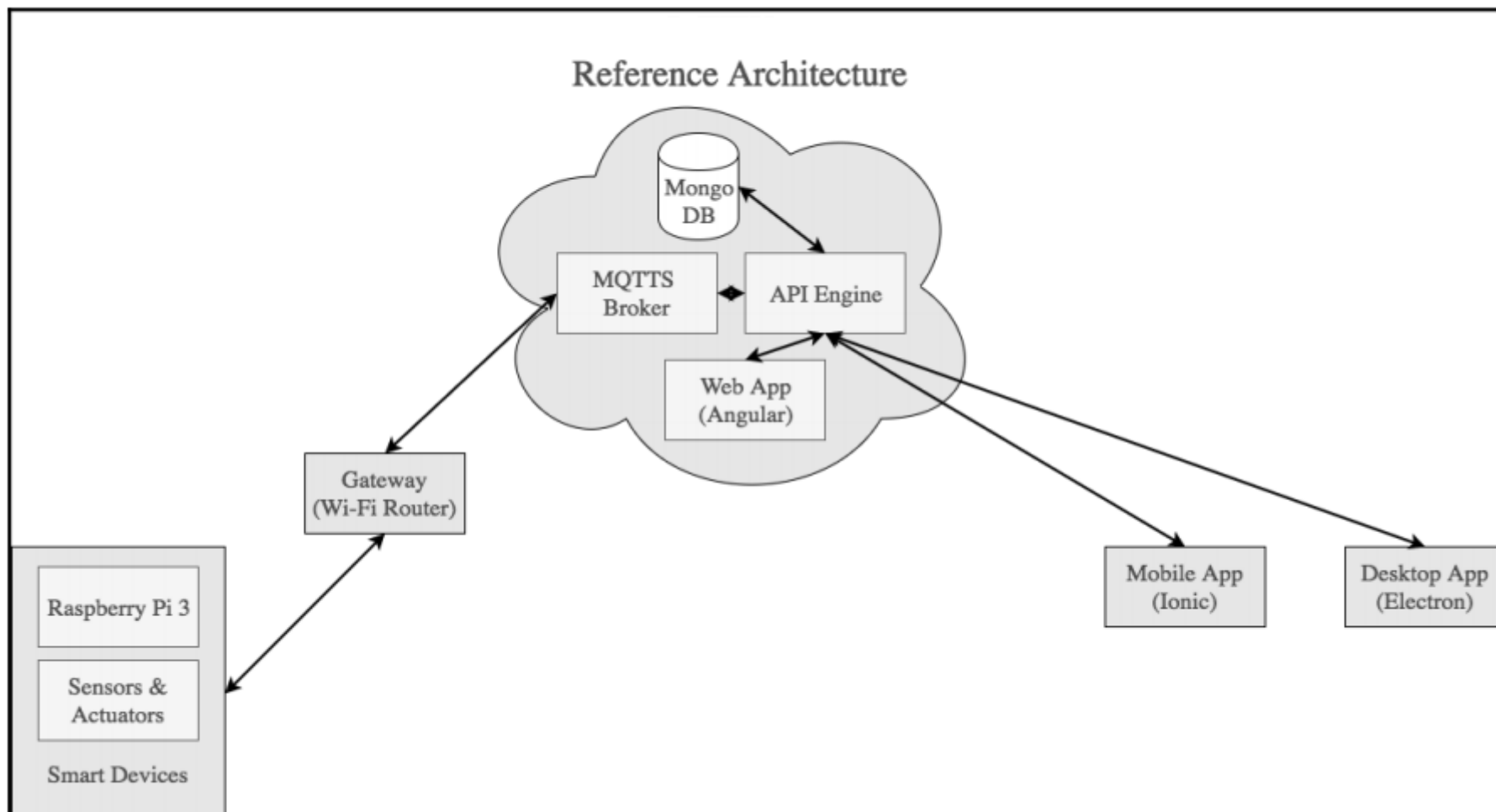


## Arquitectura cliente-servidor

HTTP method ↕	RFC ↕	Request has Body ↕	Response has Body ↕
GET	<a href="#">RFC 7231</a>	Optional	Yes
HEAD	<a href="#">RFC 7231</a>	Optional	No
POST	<a href="#">RFC 7231</a>	Yes	Yes
PUT	<a href="#">RFC 7231</a>	Yes	Yes
DELETE	<a href="#">RFC 7231</a>	Optional	Yes
CONNECT	<a href="#">RFC 7231</a>	Optional	Yes
OPTIONS	<a href="#">RFC 7231</a>	Optional	Yes
TRACE	<a href="#">RFC 7231</a>	No	Yes
PATCH	<a href="#">RFC 5789</a>	Yes	Yes



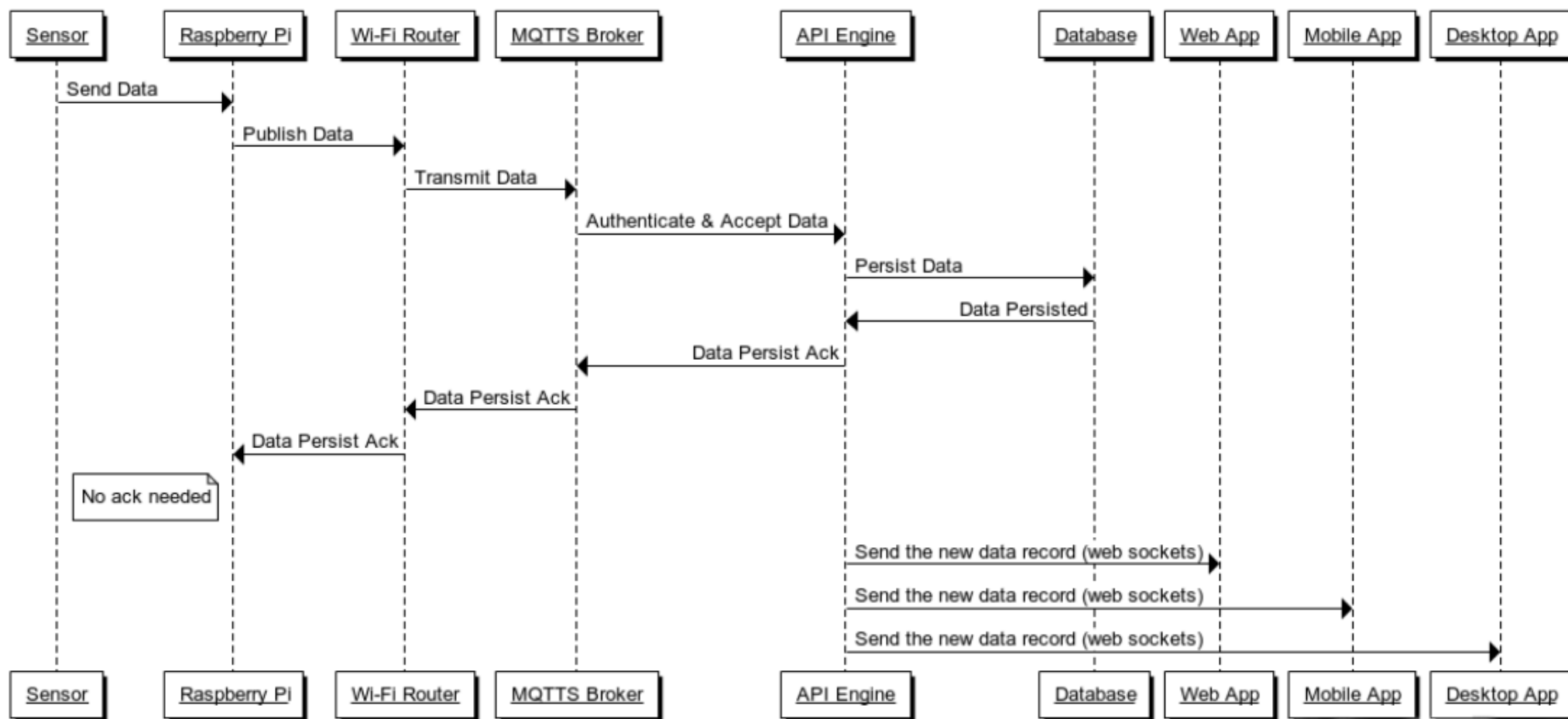
## Arquitectura cliente-servidor: Ejemplo





## Arquitectura cliente-servidor: Ejemplo

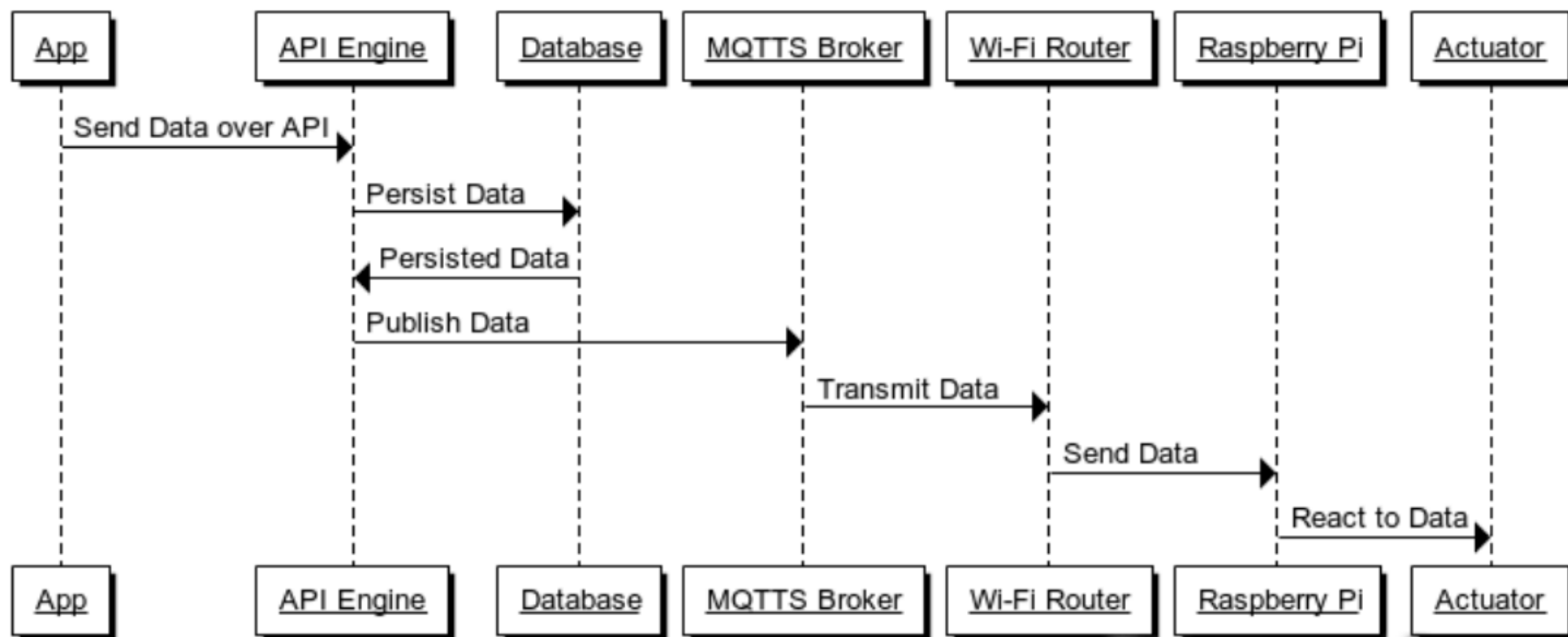
Data Flow from Sensor to Apps





## Arquitectura cliente-servidor: Ejemplo

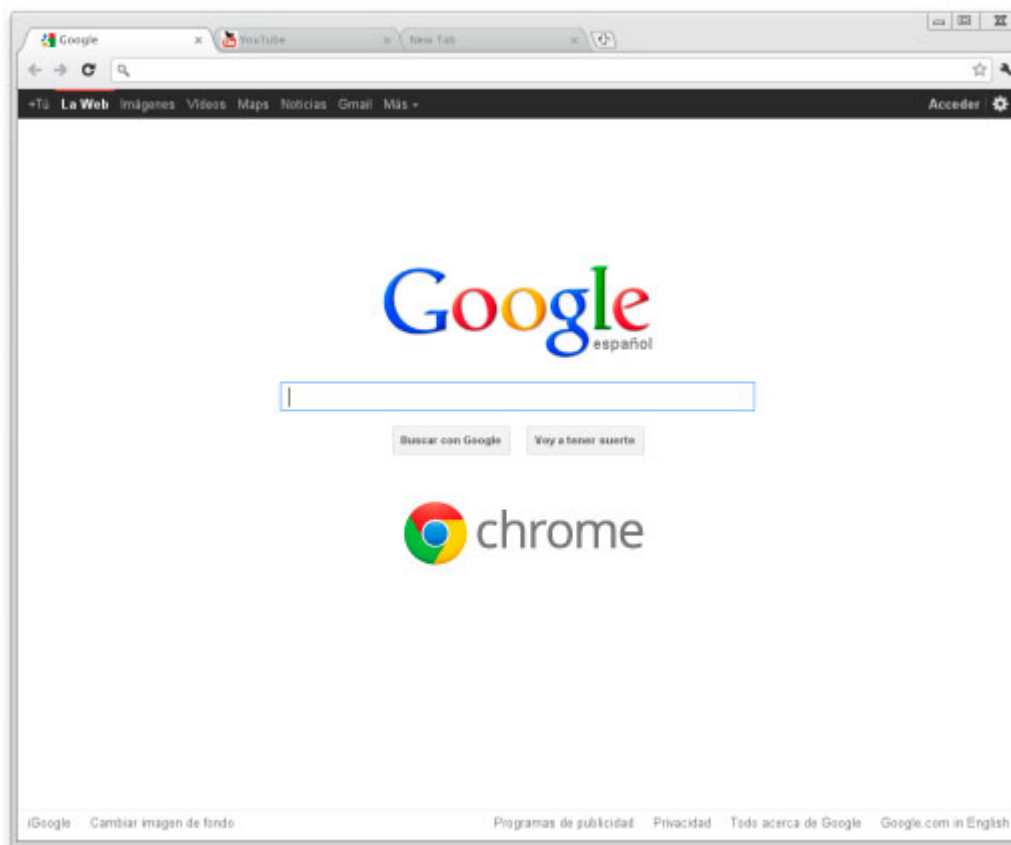
Data Flow from Apps to Actuators





## Tecnologías: Lado cliente

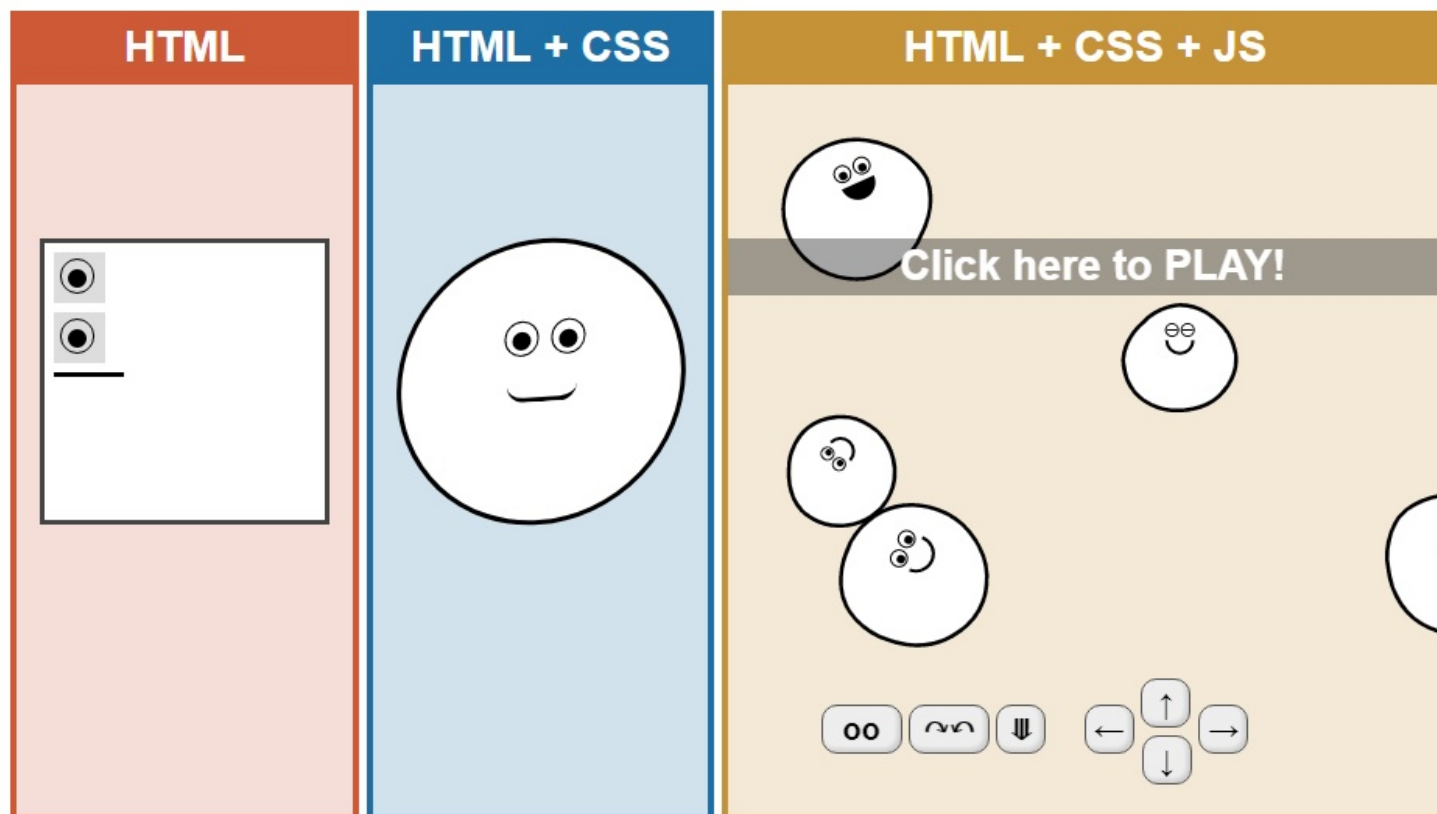
- HTML
- CSS
- Javascript





## Tecnologías: Lado cliente

- HTML
- CSS
- Javascript





## **Tecnologías: Lado cliente**

- **CSS**
  - **Style.css (manual)**
- **Bootstrap**
- **Materialize**
- **Etc.**
  - **Sistema de grillas**
  - **Estilos para tablas y componentes (botones, etc.)**



## **Tecnologías: Lado cliente**

- **CSS compilers**

- **SASS**

- **LESS**





## **Tecnologías: Lado cliente**

- **Javascript**
  - **Boilerplate.js (manual)**
- **jQuery**
- **React**
- **Angular**
- **Vue.js**
- **Etc.**



## Tipos de aplicaciones Web

- **El server devuelve:**
  - **Todo el contenido en formato HTML**
    - Todo lo que visualiza el usuario ya viene del server como HTML y CSS.
    - Las acciones del usuario en la interface generan un nuevo request y **la recarga de la página.**
- **Contenido mínimo de HTML, CSS y JS**
  - JS ejecuta requests para obtener información a mostrar.
  - JS muestra la info obtenida construyendo HTML nuevo.
  - Las acciones del usuario en la interface las detecta el código JS y éste genera nuevos requests **sin recargar la página.**



## **Tecnologías: Lado server**

- **Servidores web**

- NodeJS (Javascript)
- Apache web server (PHP)
- Tomcat/Jeti (Java)
- Apache web server (C/C++/Python)
- NGINX (contenido estático)
- Otros



## Tecnologías: Lado server

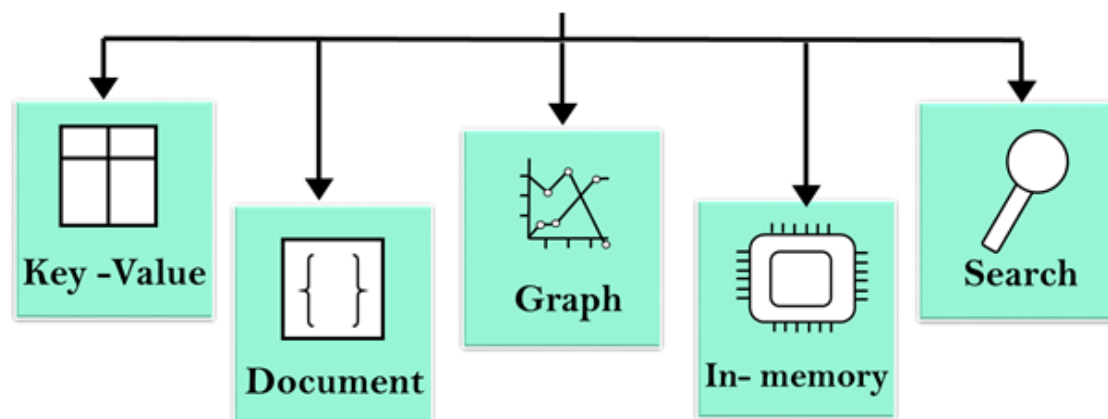
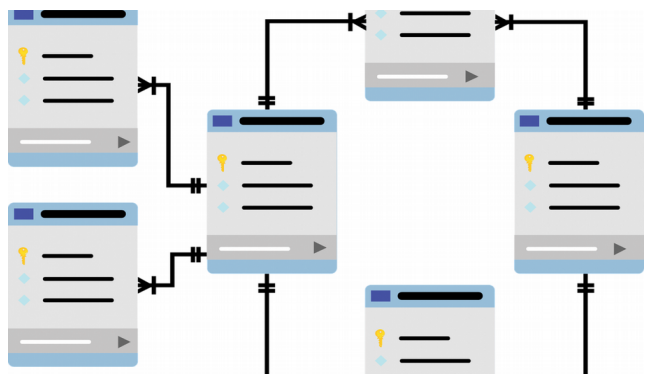
- **Bases de datos**

- Relacionales

- MySQL, MariaDB, Oracle, SQLServer

- No Relacionales

- MongoDB, Cassandra, Redis, Neo4j





## **Tipos de aplicaciones en servidor**

- Servidor de contenido estático (archivos multimedia, archivos js, etc.)
- Servidor de contenido web (HTML,CSS,JS)
- Servicio web (JSON,XML,SOAP)
- Servidor web socket
- MQTT broker
- Otros



## Tipos de aplicaciones en servidor

- Servidor de contenido estático (archivos multimedia, archivos js, etc.)
- Servidor de contenido web (HTML,CSS,JS)
- Servicio web (JSON,XML,SOAP)
- Servidor web socket
- MQTT broker
- Otros



## Herramientas seleccionadas

- **Cliente:**
  - HTML5
  - CSS3: Materialize
  - JS: Typescript.
- **Server:**
  - Apache web server
  - PHP
  - Servicios web: FatFree framework
  - DB: MySQL



## Herramientas desarrollo

- IDE: Visual Studio Code
- OS: Ubuntu 16/18





## Bibliografía

- Fielding, Roy T.; Gettys, James; Mogul, Jeffrey C.; Nielsen, Henrik Frystyk; Masinter, Larry; Leach, Paul J.; Berners-Lee, Tim (June 1999). Hypertext Transfer Protocol – HTTP/1.1
- Practical Internet of Things with JavaScript. Arvind Ravulavaru. 2017. Packt>
- Raspberry Pi for Arduino Users – Building IoT and Network Applications and devices. James R. Strickland. 2018. Apress.