



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Desarrollo de Aplicaciones I

Ramos Matias

Clase 1

- Web
- Arquitectura Cliente/Servidor
- 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

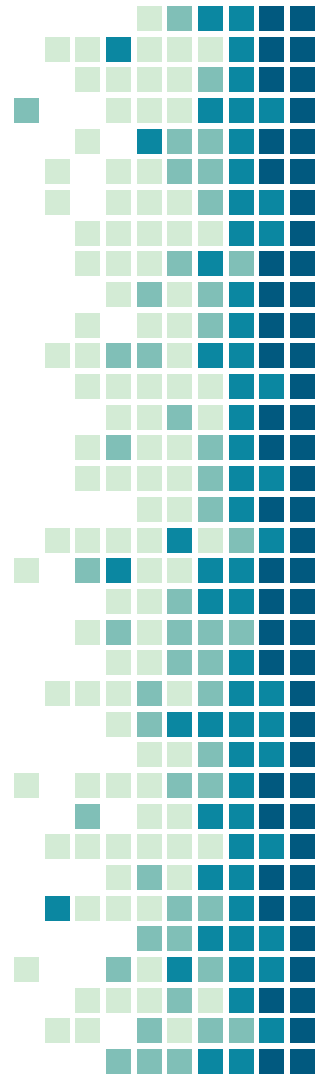
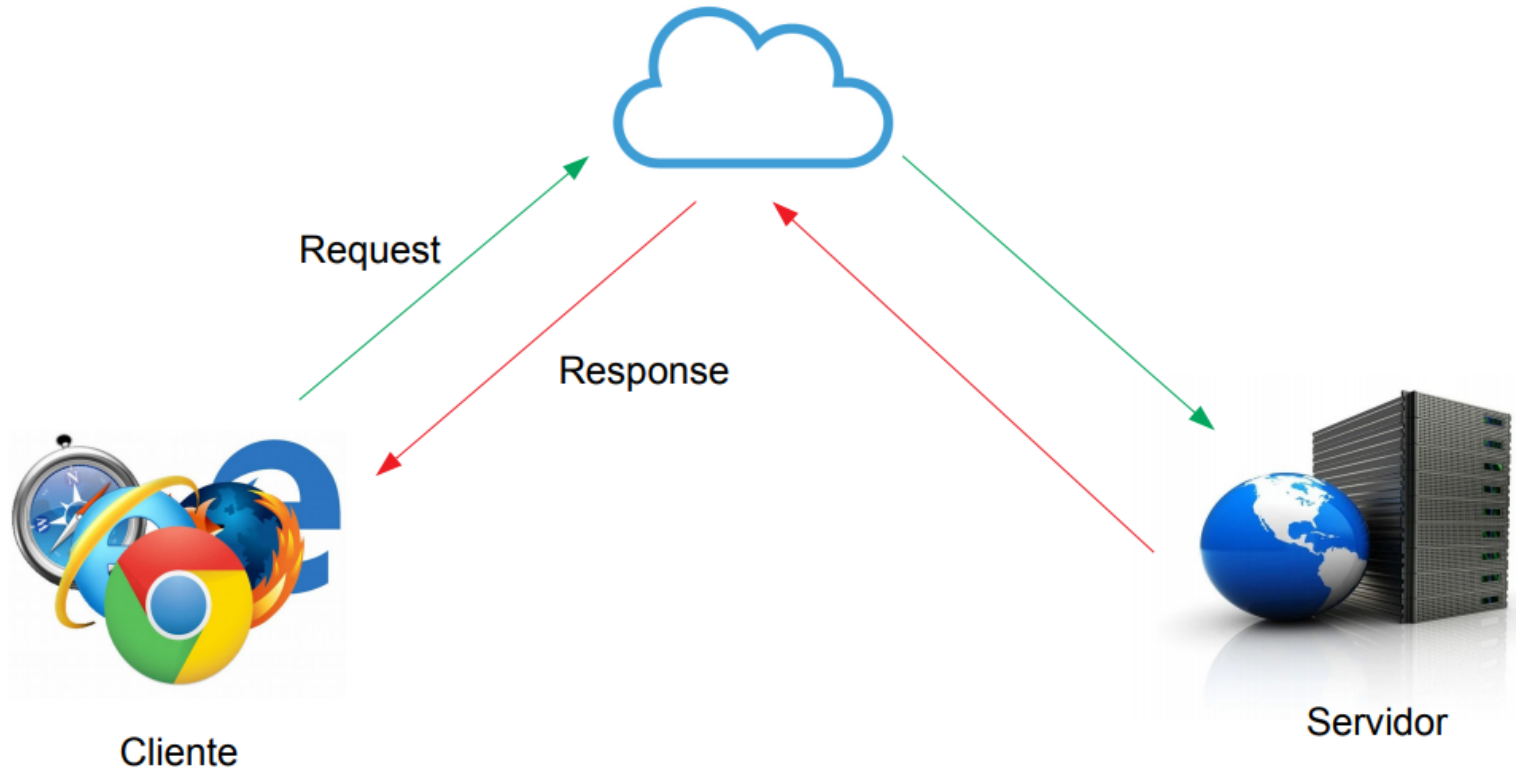


Arquitectura cliente-servidor

- Protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*)
- Protocolo HTTP (*HiperText Transfer Protocol*)
 - Sobre TCP/IP



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)
 - Sus recursos se identifican con URLs
 - 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)
 - Campo estado (200,404, etc.)
 - Cuerpo de mensaje (opcional)



Ejemplo

- El cliente hace una solicitud para ver la página principal.

GET /principal.html HTTP/1.1

Host: www.pagina.com

- El servidor envía una respuesta al cliente.

HTTP/1.1 200 OK

Date: Tue, 24 Feb 2020 14:32:01 GMT

Content-Type: text/html; charset=UTF-8

Content-Length: 75

Connection: close

<html>

 <head>

 <title>Título</title>

 </head>

 <body>

 Nada aquí.

 </body>

</html>



Tipos de métodos

HTTP method ↕	RFC ↕	Request has Body ↕	Response has Body ↕
GET	RFC 7231	Optional	Yes
HEAD	RFC 7231	Optional	No
POST	RFC 7231	Yes	Yes
PUT	RFC 7231	Yes	Yes
DELETE	RFC 7231	Optional	Yes
CONNECT	RFC 7231	Optional	Yes
OPTIONS	RFC 7231	Optional	Yes
TRACE	RFC 7231	No	Yes
PATCH	RFC 5789	Yes	Yes

Api Rest

Server



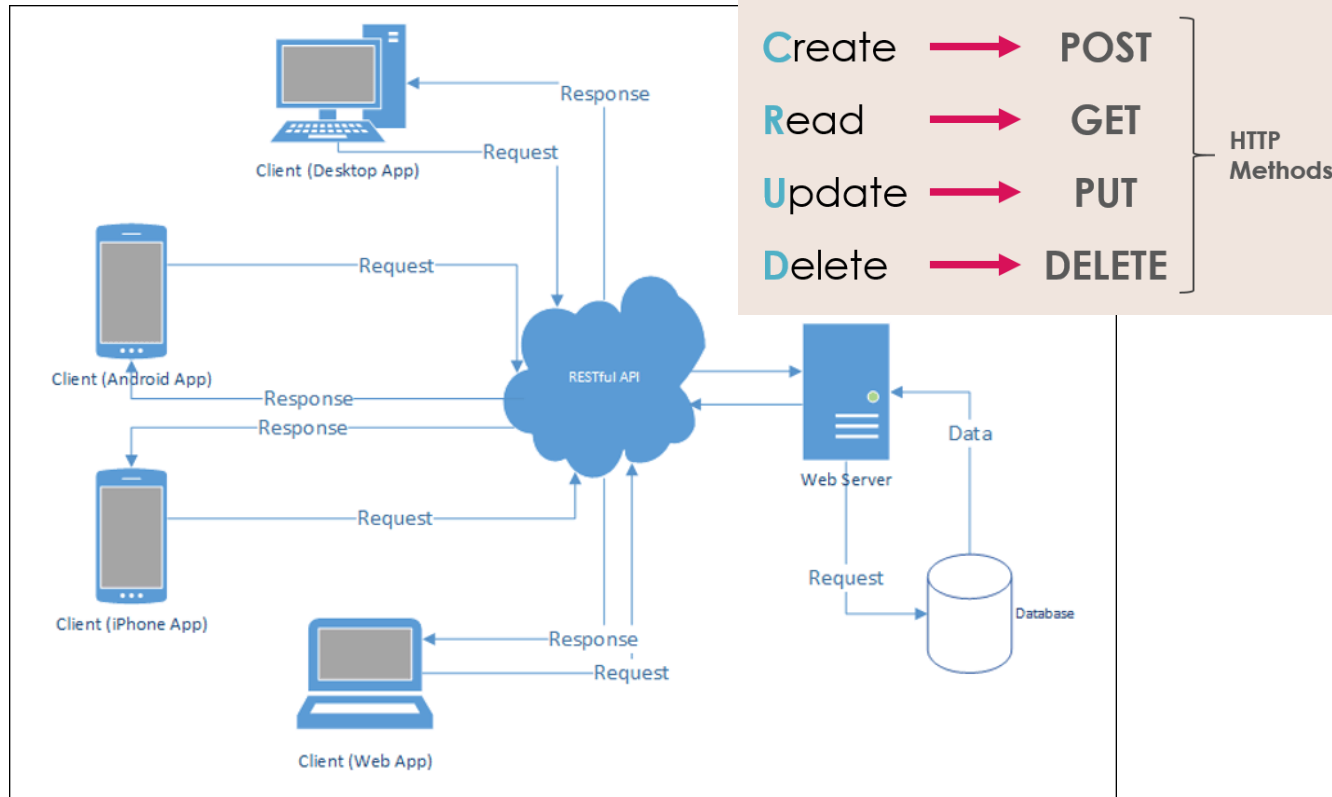
Api Rest

Url: <https://607eee2c02a23c0017e8c685.mockapi.io/users>

- GET: Pido informacion de todos los usuarios
 - Si pongo /20 la info del usuario 20
- Delete: Borro a un usuario
- Put: Modifico un usuario (los datos van en el body)
- Post: Creo un nuevo usuario (body)



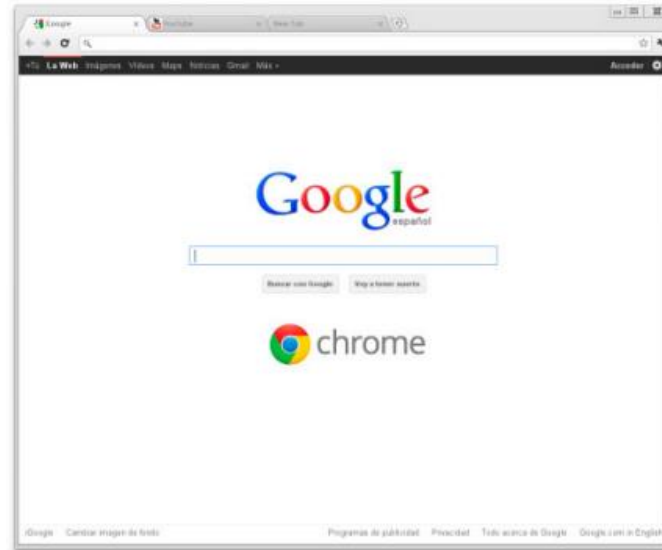
Api Rest



Tecnologías: Lado cliente

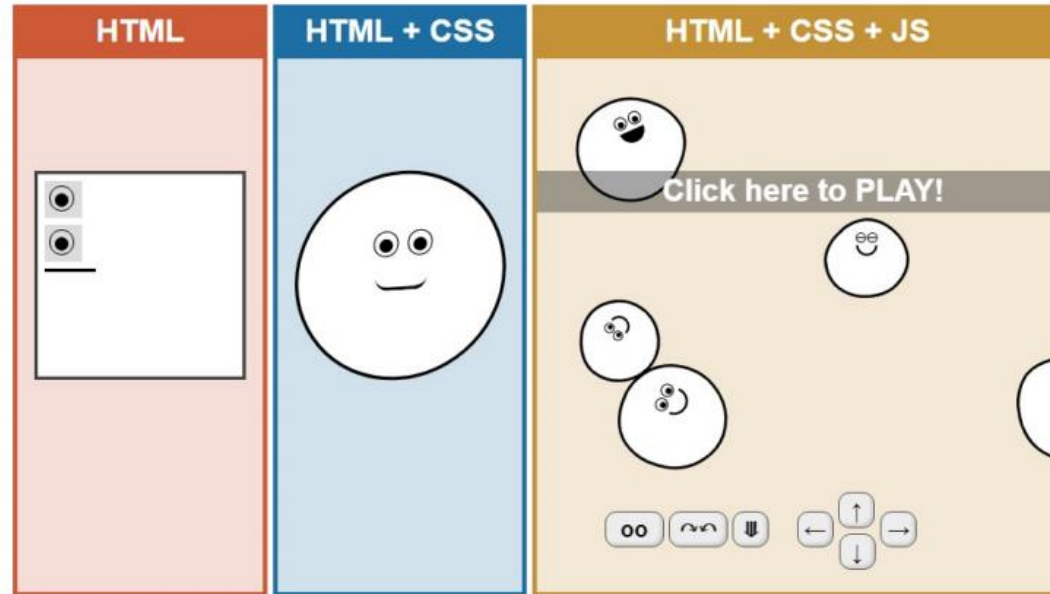
- HTML
- CSS
- Javascript

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <link href="bootstrap.min.css" rel="stylesheet">
    <script src="run_prettify.js"></script>
  </head>
  <body>
    ...
```



Tecnologías: Lado cliente

- HTML
- CSS
- Javascript



Tecnologías: Lado cliente

- CSS
 - Style.css (manual)
 - Bootstrap
 - Materialize



Tecnologías: Lado cliente

- CSS compilers
 - SASS
 - LESS



Tecnologías: Lado cliente

- Javascript
 - Boilerplate.js (manual)
 - JQuery
 - React
 - Angular



Tipos de aplicaciones Web

- **Static web app:** El contenido lo genera el servidor y siempre es el mismo (Por ej. Un server de archivos). No se ejecuta código en el server. Cada contenido tiene una URL/página diferente.
- **Dynamic web app:** El contenido lo genera el servidor ejecutando código que construye el contenido. Cada contenido tiene una URL/ página diferente.
- **JS web app:** El servidor genera cierto contenido y el resto lo genera el código JS que realiza requests extras al server y consume HTML o JSON que agrega al contenido. Cada contenido tiene una URL/página diferente.

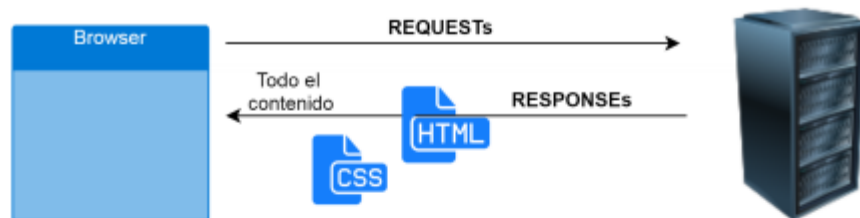


Tipos de aplicaciones Web

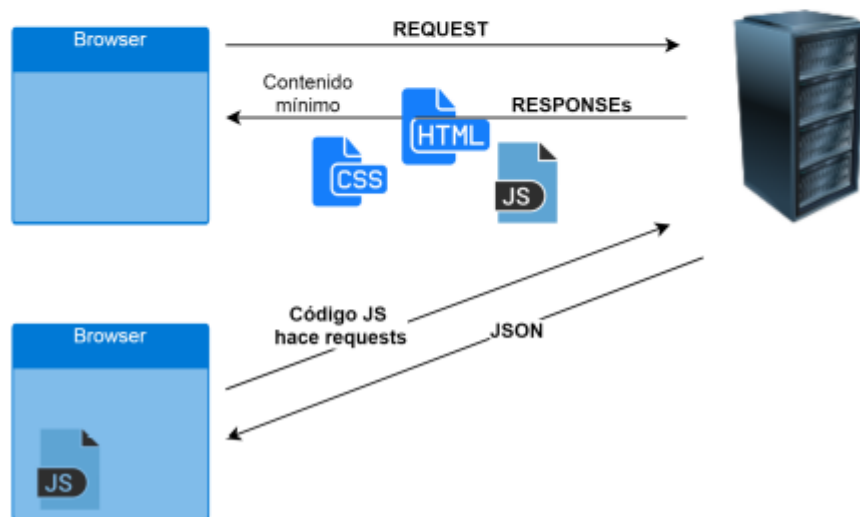
- **Single Page Application:** El código JS que corre en el cliente cambia el contenido de la página sin recargarlo, haciendo requests al server, que consumen JSON/XML del mismo.
- **Progressive Web App:** Es una SPA con mejoras en la interfaz mobile para que sean fluídas y confiables bajo una mala conexión u offline. Funcionan en el navegador y/o desktop y pueden instalarse como una app nativa mobile.



static/Dynamic web app



Single Page Application (SPA)



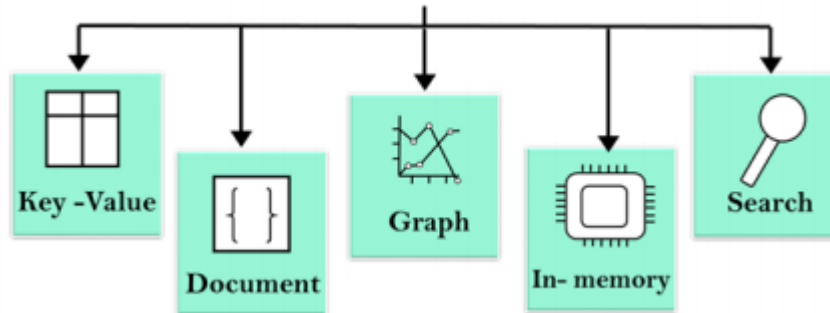
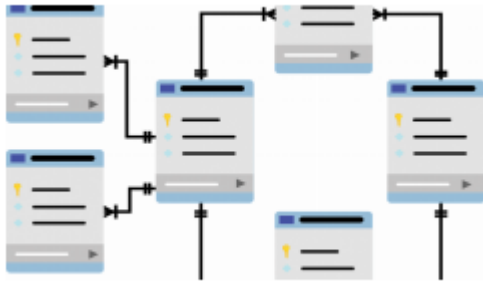
Tecnologías: Lado server

- Servidores web
 - NodeJS (Javascript)
 - Apache web server (PHP)
 - Tomcat/Jeti (Java)
 - Apache web server (C/C++/Python)



Tecnologías: Lado server

- Bases de datos
 - Relacionales
 - MySQL, MariaDB, Oracle, SQLServer
 - No Relacionales
 - MongoDB, Cassandra, Redis



Tipos de aplicaciones en servidor

- Servidor de contenido web (HTML,CSS,JS)
- Servicio web RESTful (JSON)



Herramientas seleccionadas

- Cliente:
 - HTML5
 - CSS3: Materialize
 - JS: Typescript.
- IDE: Visual Studio Code

