



Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación

Mauricio Gaona
mauricio.gaona@correounivalle.edu.co

Profesor

2023-I

Desarrollo de Software I



01

RESUMEN

02

CARACTERÍSTICAS DEL
FRONTEND Y EL BACKEND

03

EXPOSICIONES

04

ESTADO DEL PROYECTO

Conceptos

Resumen





Preguntas ?





Ítems que deberían tener en cuenta para la primera presentación de avance del proyecto

Fecha de entrega mayo 24 de 2023

1. Título del Sistema
2. Objetivos (principales funcionalidades solicitadas)
3. Empresas a las que le sirve el sistema
4. Idea del proyecto: Presenta un resumen de no más de 100 palabras que resumen de la idea del proyecto.
5. Fechas críticas: Define las fechas críticas de la aplicación tales como inicio y finalización del proyecto y fechas estimadas de los releases. (Entrega final del proyecto junio 28)
6. Miembros del equipo de desarrollo: El equipo se organiza a través de asignación de los 3 roles básicos de la metodología distribuidos entre los integrantes del grupo
7. Product backlog (o enlace a donde este): Backlog priorizado y estimado.
8. Prácticas ágiles a usar durante el desarrollo: Presenta un listado de las prácticas ágiles que acordaron usar durante el desarrollo.
9. Tecnologías a usar : Define mediante un listado todas las tecnologías de desarrollo de la aplicación
10. Diagrama de bloques(módulos de la aplicación): Presenta un diseño inicial de los módulos de la aplicación que representan el estilo arquitectural de la aplicación.
11. Modelo de datos (modelo relacional): Presenta un modelo de datos inicial.
12. Diseño de las interfaces de la aplicación (Interfaces del primer sprint)
13. Release plan: Presenta un release plan inicial donde presenta los sprints a realizar, HU de cada sprint, responsable de cada HU y fecha estimada de entrega de los sprints y releases del proyecto.
14. Detalles de la Historias de usuario : Presenta el detalle de las HU de al menos el primer sprint en un formato definido.
15. Acrónimos: Presenta una lista de los acrónimos usados con su respectiva definición (si los hay)
16. Primer Sprint del proyecto.



Fecha de Exposiciones, Taller y Proyecto

Grupo	Tema	Fecha exposición
	Pruebas unitarias en Python y Django	Mayo 3 de 2023
	Estándares de codificación en Python y Django	Mayo 3 de 2023
	Taller Django	Mayo 10 de 2023
	Tablero Kanban	Mayo 17 de 2023
	Chrome DevTools	Mayo 17 de 2023
	Primera entrega del proyecto	Mayo 24 de 2023
	Git y git flows (incluir git tags)	Mayo 31 de 2023
	*Web services con Django	Mayo 31 de 2023
	Motion UI	Junio 7 de 2023
	*WAF (Web Applications Firewall)	Junio 7 de 2023
	Desarrollo de aplicaciones web usando ChatGPT	Junio 14 de 2023
	Entrega final del proyecto	Junio 28 de 2023



Preguntas ?





Modelo de Despliegue

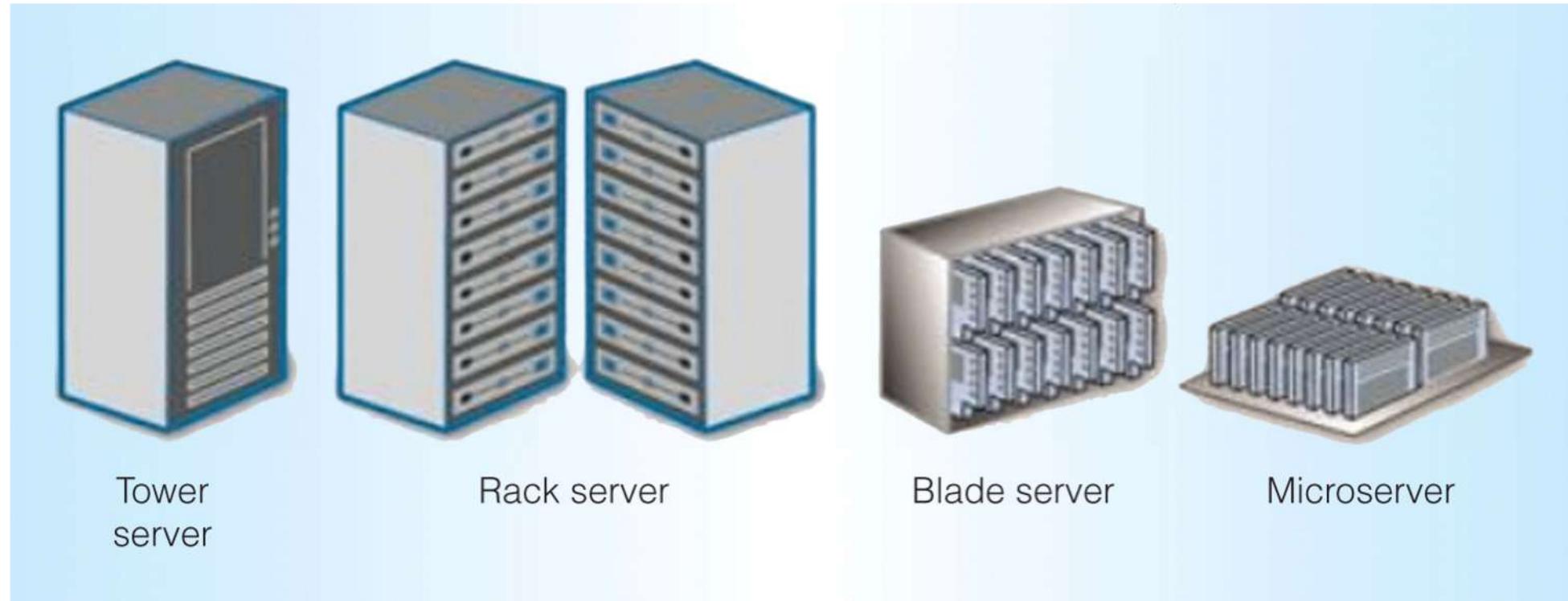
Representa la plataforma o el conjunto de servidores requeridos por una aplicación de software para ponerla en funcionamiento para ser usada por los usuarios finales.



Modelo de despliegue

Servidor (Host)

Evolución del hardware para servidores



Un equipo

Data Center local (on-premise)

Data Center externo

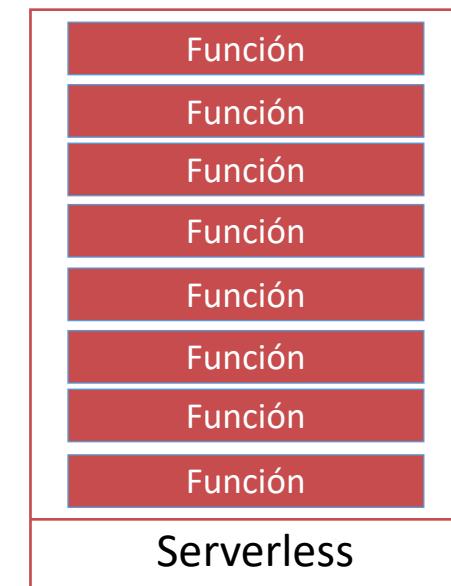
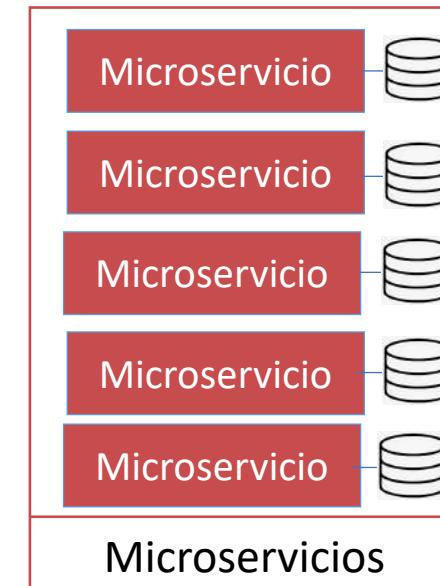
Cloud Data Center

Donde colocar los servidores ?

Desarrollo de Software I

Modelo de despliegue

¿Como construir aplicaciones para ser desplegadas: Estilo arquitectural



Servidores físicos o virtuales

Plataformas en la nube

Desarrollo de Software I

Modelo de despliegue

Ambientes para desarrollo, pruebas y despliegue de aplicaciones



Modelo de despliegue

Como calcular el tamaño de los servidores de los nodos para despliegue de aplicaciones (RAM, Procesadores, Espacio de almacenamiento, Ancho de banda)

Factores a tener en cuenta para determinar el tamaño de un servidor:

- Carga (tráfico) : Número de peticiones que depende del número de usuarios concurrentes
- Recurso que requiere la aplicación: RAM, Procesadores, Almacenamiento, Ancho de banda
- Tipo de Aplicación: Cantidad de recursos que consume. Nota: si la aplicación usa muchos videos estos se deben colocar en un servidor de videos como youtube, Vimeo, etc.
- Escalabilidad requerida : Crecimiento por demanda
- Presupuesto

Modelo de despliegue

Despliegue de aplicaciones (Deploy)

El despliegue de aplicaciones consiste en copiar los archivos de una aplicación en un servidor o nodo donde funcionará la aplicación. (configurado)

Que archivos copiar:

- Programas ejecutables o scripts
- Librerías requeridas con sus versiones específicas
- Otras librerías o paquetes (Ej: librerías gráficas, etc)
- Archivos de datos requeridos
- Bases de datos y drivers de conexión

Modelo de despliegue

Despliegue continuo: Despliegue por Release, por Sprint o Hot fix

Métodos para hacer Deploy:
Manual

Automático: **IaC** (Infrastructure as a Code)

Instalar desde un repositorio (EJ. Git pull)
`git clone https://github.com/github/miapp.git`



Infraestructure as a Code (IaC)
Infraestructura como código (IaC) significa administrar su infraestructura de TI mediante archivos de configuración.

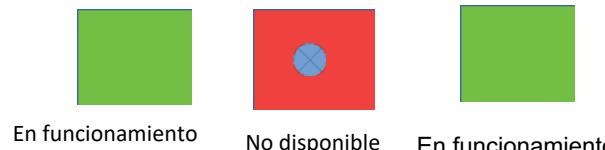


Los procesos de desarrollo ágil intentan que el “Deploy”
cada vez sea más automático

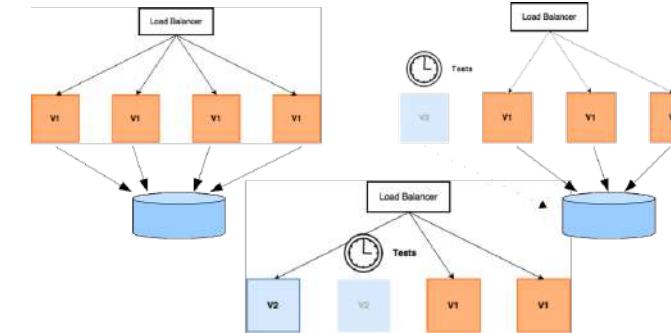
Desarrollo de Software I

Despliegue de aplicaciones Web: Estrategias para actualizar el servidor

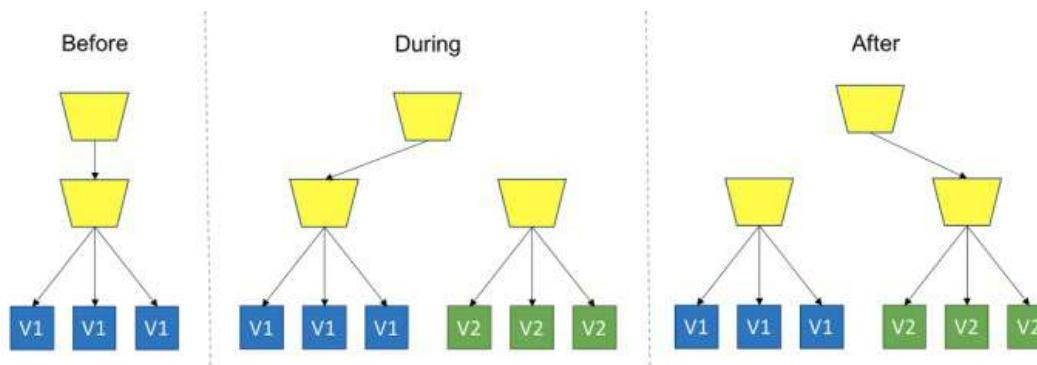
Tipo Stand Alone (Stop-Start)



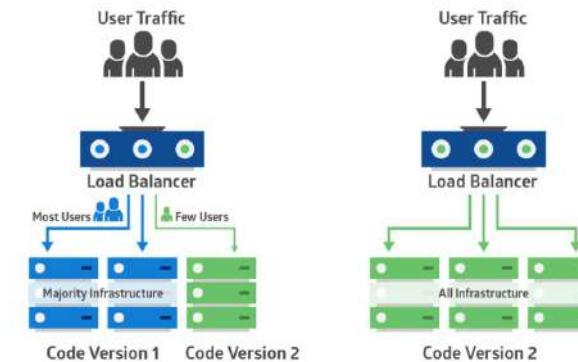
Rolling upgrade



Despliegue Blue/Green



Canary Deployment





Preguntas ?





Introducción a las Arquitecturas Ágiles de Software

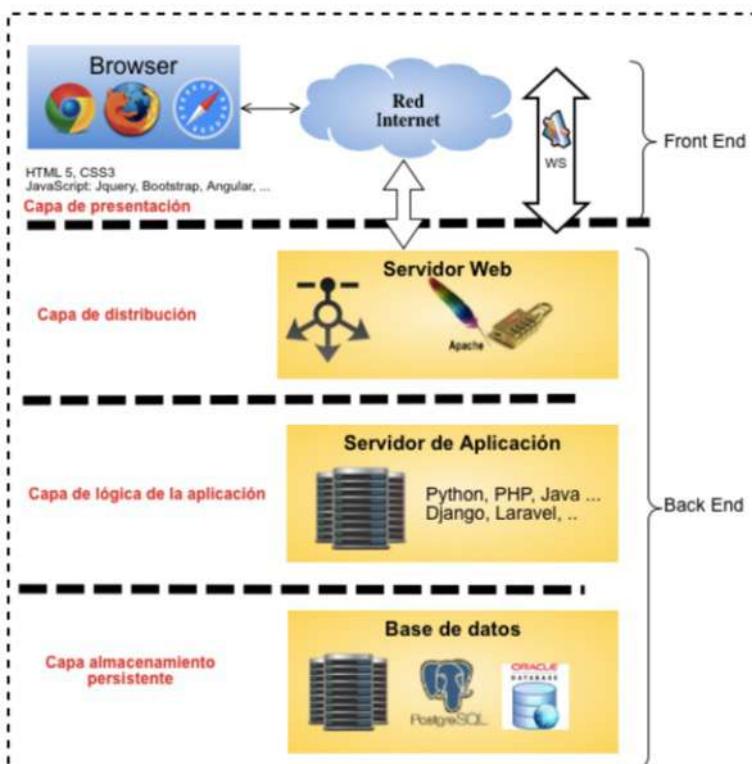


Desarrollo de Software I

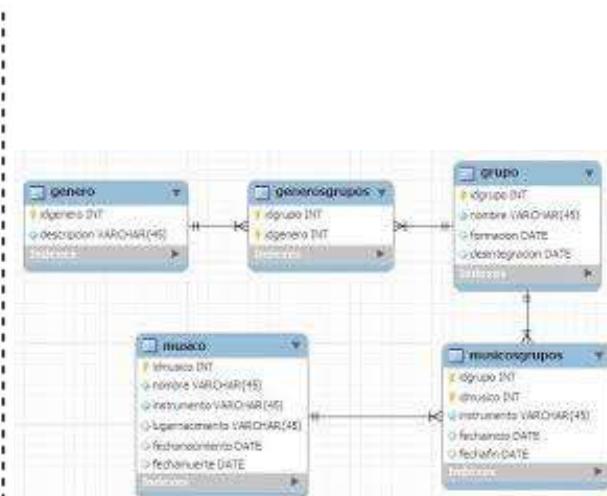
Iniciando un proyecto de software usando prácticas ágiles

Modelos arquitecturales del sistema

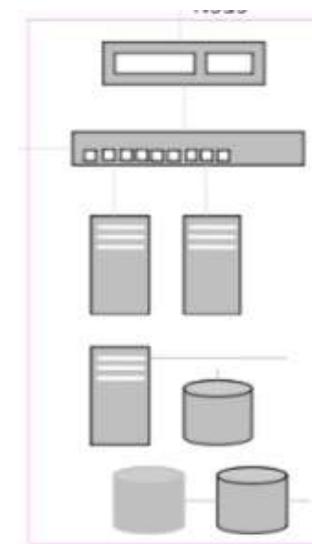
“Simple Sketches for Diagramming your Software Architecture “



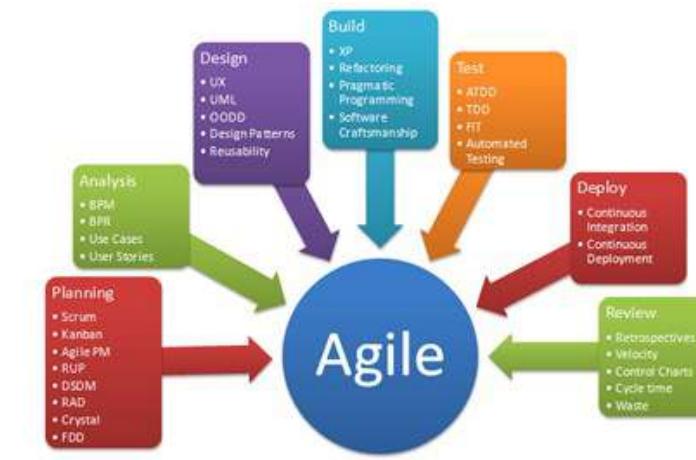
Estilo arquitectural de la aplicación



Modelo de datos (inicial)



Modelo de despliegue



Resultados de las prácticas ágiles

Arquitectura ágil de software

Documentación de una arquitectura ágil

Documento entre 4 y 6 páginas con enlaces

Sección	Descripción	Contenido
Descripción del producto	Descripción del producto (Máximo 100 palabras)	<ol style="list-style-type: none">1. Que hace mi producto2. Objetivos del producto y valor del negocio.3. Problemas que el producto soluciona.4. Tipos de empresas objetivo y usuarios objetivo5. [Enlaces a documentación relevante del producto o productos similares.]
Visión general de la arquitectura del producto	Descripción general de la arquitectura del sistema con sus versiones y vistas conceptuales	<p>Vistas :</p> <ul style="list-style-type: none">• Descripción general de la arquitectura (100 palabras)• Diseño de la interfaz de usuario (las más importantes)• Diseño para el desarrollo (equipos y herramientas usadas para el desarrollo, control y seguimiento)• Principales Componentes del sistema (Diagrama de Bloques o Diagrama de Microservicios)• Modelo de datos (alto nivel)• Diagrama de despliegue

Documentación de una arquitectura ágil

Documento entre 4 y 6 páginas

Sección	Descripción	Contenido
Requerimientos (Historias de usuario)	Descripción de alto nivel de las funcionalidades	<ul style="list-style-type: none">• Product Backlog• Detalles de las historias de usuario (link)
Prácticas ágiles en el proceso de desarrollo	Descripción corta de las prácticas ágiles usadas	<ul style="list-style-type: none">• Resultados de las prácticas ágiles realizadas (link) <p>Para las ceremonias videos o actas de lo presentado o realizado en la reunión</p>
Acrónimos	Acrónimos usados en el documento	<ul style="list-style-type: none">• Descripción corta



Preguntas ?



Acuerdos iniciales para el desarrollo de una App Web

Frontend (UI)



Acuerdos iniciales para el desarrollo de una App Web

Diseño Front end basado en Plantillas (Templates)



Desarrollo web **Front-end**: el desarrollo de frontend implica la construcción mediante páginas web las interfaces de usuario para aplicaciones web.

En el desarrollado front-end se implementan la estructura, el diseño, el comportamiento y la animación de todo lo que se puede presentar en un navegador para sitios web, aplicaciones web o aplicaciones móviles.

Se usa principalmente HTML, CSS y JavaScript.



Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)



```
<audio autoplay="autoplay" controls="controls">  
<source src="music.ogg" />  
<source src="music.mp3" />  
</audio>
```

Almacenamiento: cache del navegador, **localStorage**, **Web Storage**

Etiquetas de estructura:

< menu >

Etiquetas semánticas:

WebSocket (<ws://...>)

Pull y Push

<article>
<aside>
<details>
<figcaption>
<figure>
<footer>
<header>
<mark>
<nav>
<section>
<summary>
<time>





Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

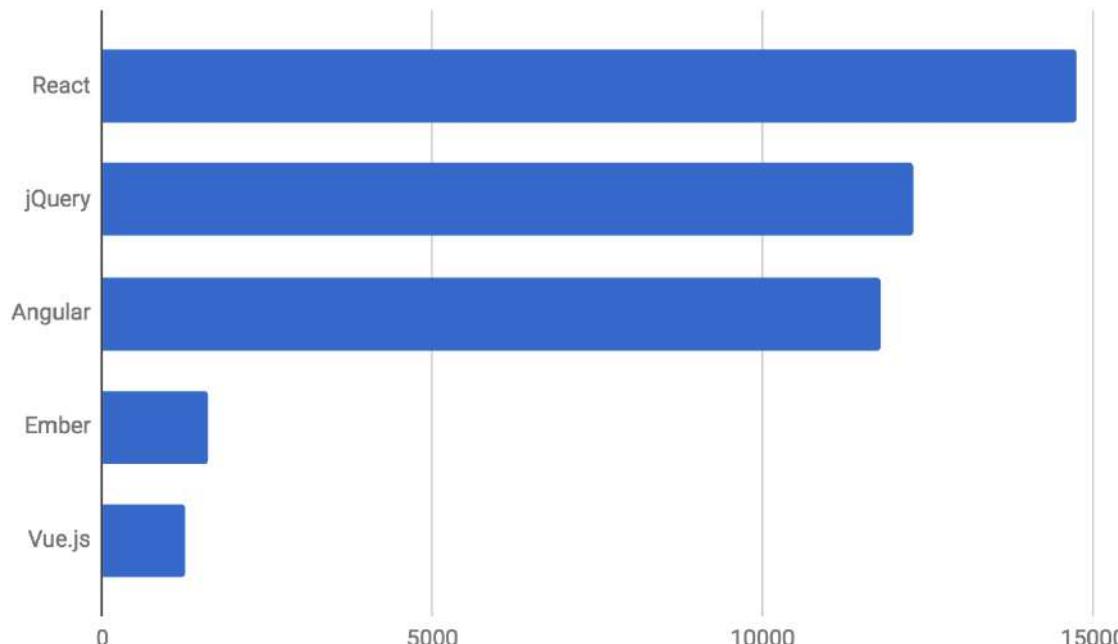
Librerías de HTML, CSS y JavaScript

Bootstrap: Bootstrap es una colección de piezas de código reutilizables escritas en HTML, CSS y JavaScript. Sin embargo, es un marco de desarrollo web front-end gratuito y de código abierto para crear sitios web y aplicaciones web responsivos. [Foundation, Bulma, Skeleton, Pure, ...]

https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp

Principales tecnologías en JavaScript: Jquery, React, Angular y ... Vue

ECMA-262 → ECMAScript → JavaScript (Vanilla JavaScript)



Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

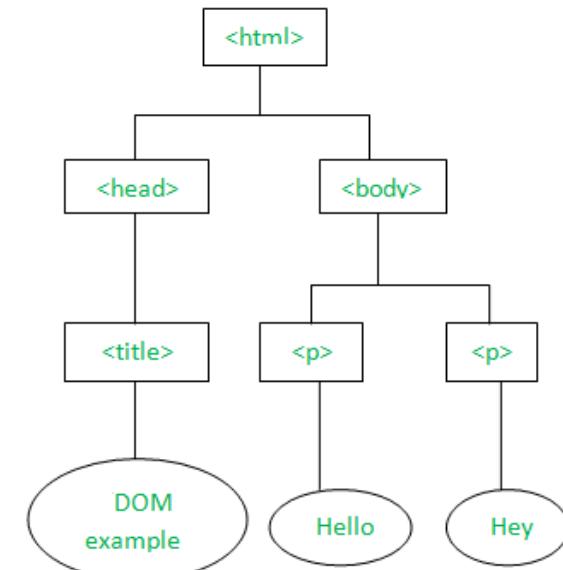
DOM (Document Object Model)

Es una API definida para representar e interactuar con cualquier documento HTML o XML.

El DOM es un modelo de documento que se carga en el navegador web y que representa el documento HTML como un árbol de nodos, en donde cada nodo representa una parte del documento

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Estructura de árbol del DOM</title>
  </head>
  <body>
    <h1>Estructura de árbol del DOM</h1>
    <h2>Aprende sobre el DOM</h2>
  </body>
</html>
```

```
<html>
<head>
<title>
  DOM example
</title>
</head>
<body>
  <p id = "para1">Hello</p>
  <p id = "para2">Hey</p>
</body>
</html>
```



Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

DOM (Document Object Model)

Como seleccionar Elementos en el Documento

Hay diferentes métodos para seleccionar un elemento en el documento de HTML.

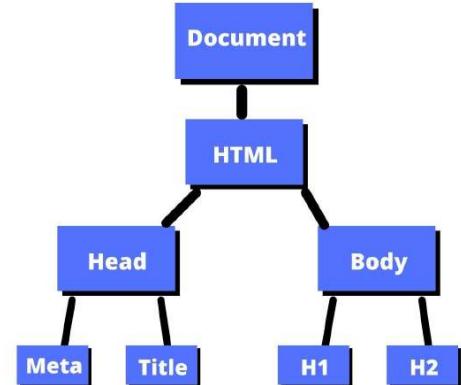
Ejemplo:

- getElementById()
- querySelector()
- querySelectorAll()
- appendChild()

```
const paragraph1 = document.getElementById("para1");
console.log(paragraph1);
```

<https://single-spa.js.org/docs/examples/>

<https://react.microfrontends.app/planets/7>



Ejemplo:

```
<p id="para1">Este es mi primer párrafo.</p>
<p id="para2">Este es mi segundo párrafo.</p>
```

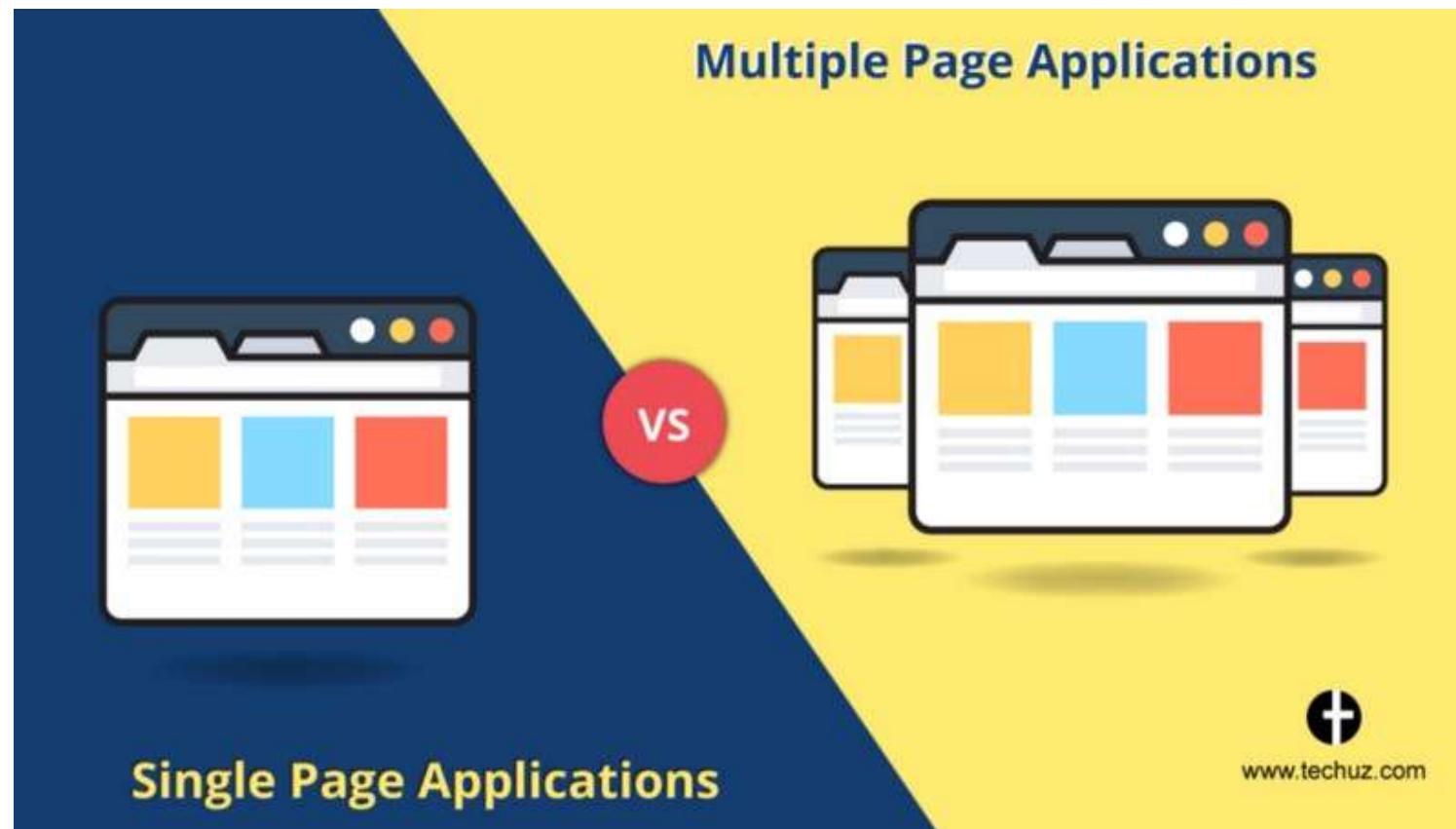
Console

```
<p id="para1"> Este es mi primer párrafo.</p>
```

Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

Tipo de frontend en las aplicaciones Web: SPA o MPA



Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

SPA (Single Page Application)

¿Qué es la aplicación de una sola página?

Las aplicaciones de una sola página consisten en presentar todas las funcionalidades una sola página, usando principalmente Ajax.

Los SPA presentan el contenido de una manera fácil, elegante y efectiva, ya que carga todo el contenido en una sola página en lugar de que el usuario tenga que navegar por diferentes páginas.

Los SPA son más rápidos que las aplicaciones web tradicionales porque ejecutan gran parte de la lógica en el navegador web en lugar de hacerlo en el servidor. Y después de la carga inicial de la página, solo se envían datos de ida y vuelta en lugar de todo el HTML.





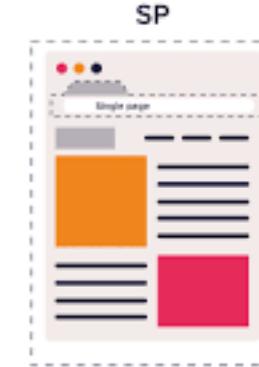
Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

SPA (Single Page Application)

Ventajas de las aplicaciones de una sola página

1. Las aplicaciones de una sola página suelen ser rápidas.
2. Facilitar tener páginas responsivas.
3. Se puede pasar a las aplicaciones móviles reutilizando el mismo código de backend.



Desventajas de las aplicaciones de una sola página

1. Los SPA se desempeñan mal en el motor de búsqueda.
2. Las aplicaciones de una sola página proporcionan un solo enlace para compartir y requieren más memoria.
3. Los SPA son más difíciles de depurar.
4. Se requiere implementar lógicas complicadas para mejorar la seguridad.

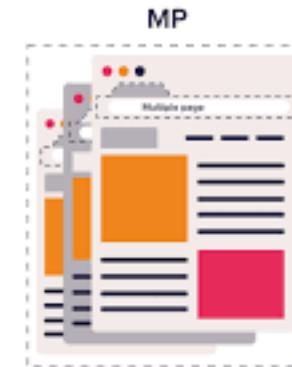
Ejemplos:

Facebook, Twitter, Gmail, Google Docs, etc.

Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

MPA : Multi Page Application



Las aplicaciones multipágina son las aplicaciones web tradicionales que recargan toda la página y muestran la nueva cuando un usuario interactúa con la aplicación web.

Cada vez que se intercambian datos de un lado a otro, se solicita una nueva página del servidor para que se muestre en el navegador web. Este proceso lleva tiempo generar las páginas en el servidor, enviarlas a un cliente y mostrarlas en el navegador, lo que puede afectar la experiencia del usuario si la página a traer es muy pesada.



Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

MPA : Multi Page Application

Beneficios de las aplicaciones de múltiples páginas

1. Fáciles de indexar por los motores de búsqueda.
2. Consumen menos memoria y se pueden tener muchas páginas.
3. Navegación más fácil (mapa del sitio).
4. Facilitan algunos aspectos de seguridad.



Inconvenientes de las aplicaciones de múltiples páginas

1. Menor rendimiento
2. Más tiempo de desarrollo
3. Mantenimiento más costoso



Desarrollo de Software I

Acuerdos iniciales para el desarrollo de una App Web

Diseño de Frontend (UI)

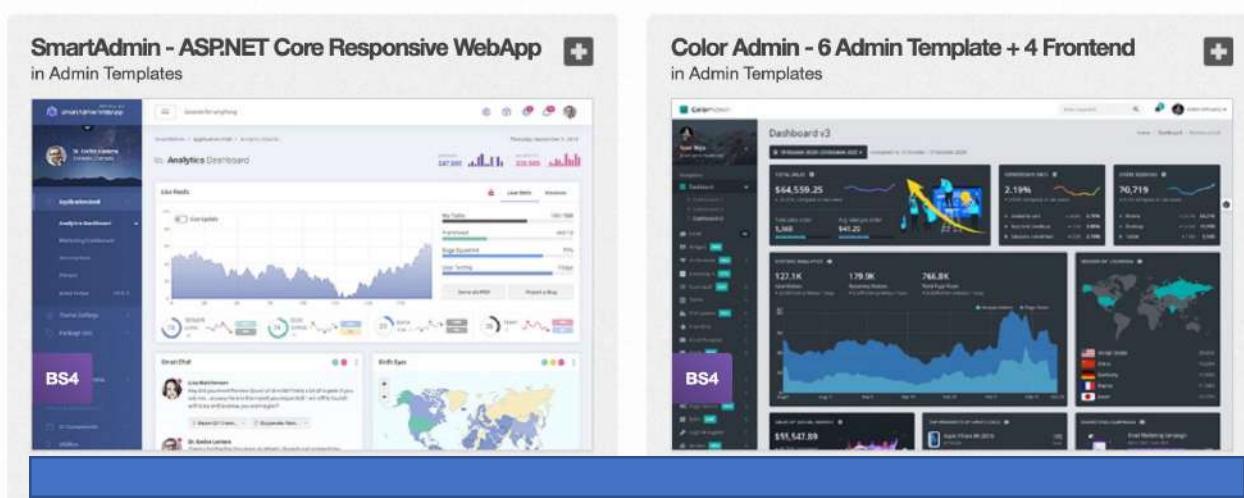
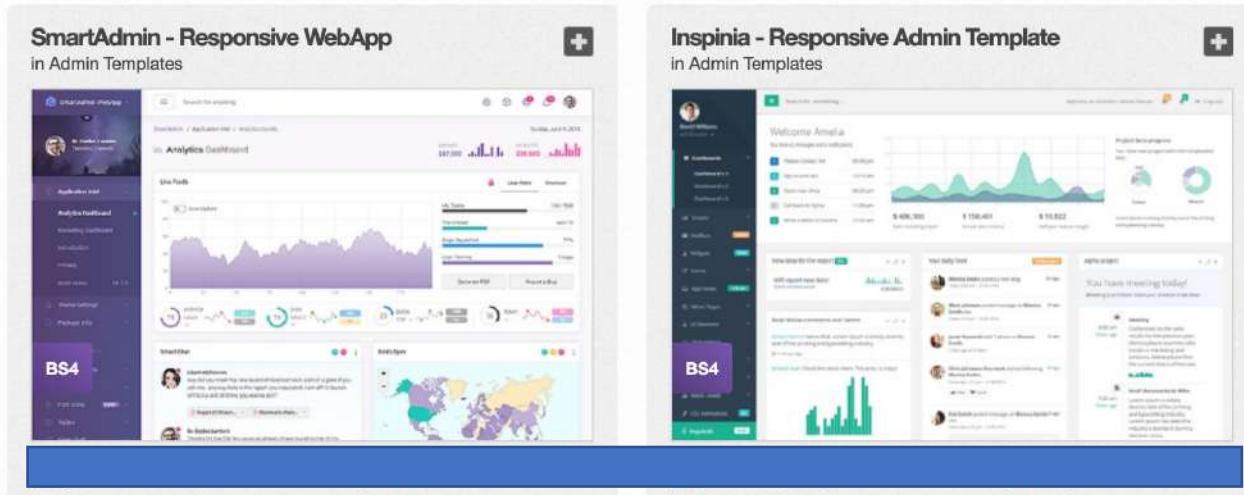
Plantillas

<https://wrapbootstrap.com/category/templates/admin-templates>

Best free Bootstrap templates for ReactJS

<https://www.codeinwp.com/blog/best-free-bootstrap-templates-for-reactjs/>

<https://material-ui.com/store/collections/free-react-dashboard/>







Backend

- El backend es toda la tecnología requerida para procesar una solicitud entrante, generar y enviar la respuesta al cliente.
- Típicamente el backend tiene tres partes:
 - **Aplicación**: Conjunto de scripts o programas responsables de toda la lógica de la aplicación también se incluyen APIs si se requiere.
 - **Bases de datos** : Donde se almacena la información.
 - **Servidor (host)** : Donde se aloja la aplicación, el servidor web y el motor de bases de datos; una aplicación web usa uno o varios servidores para su ejecución y depende de su arquitectura de despliegue.



Backend

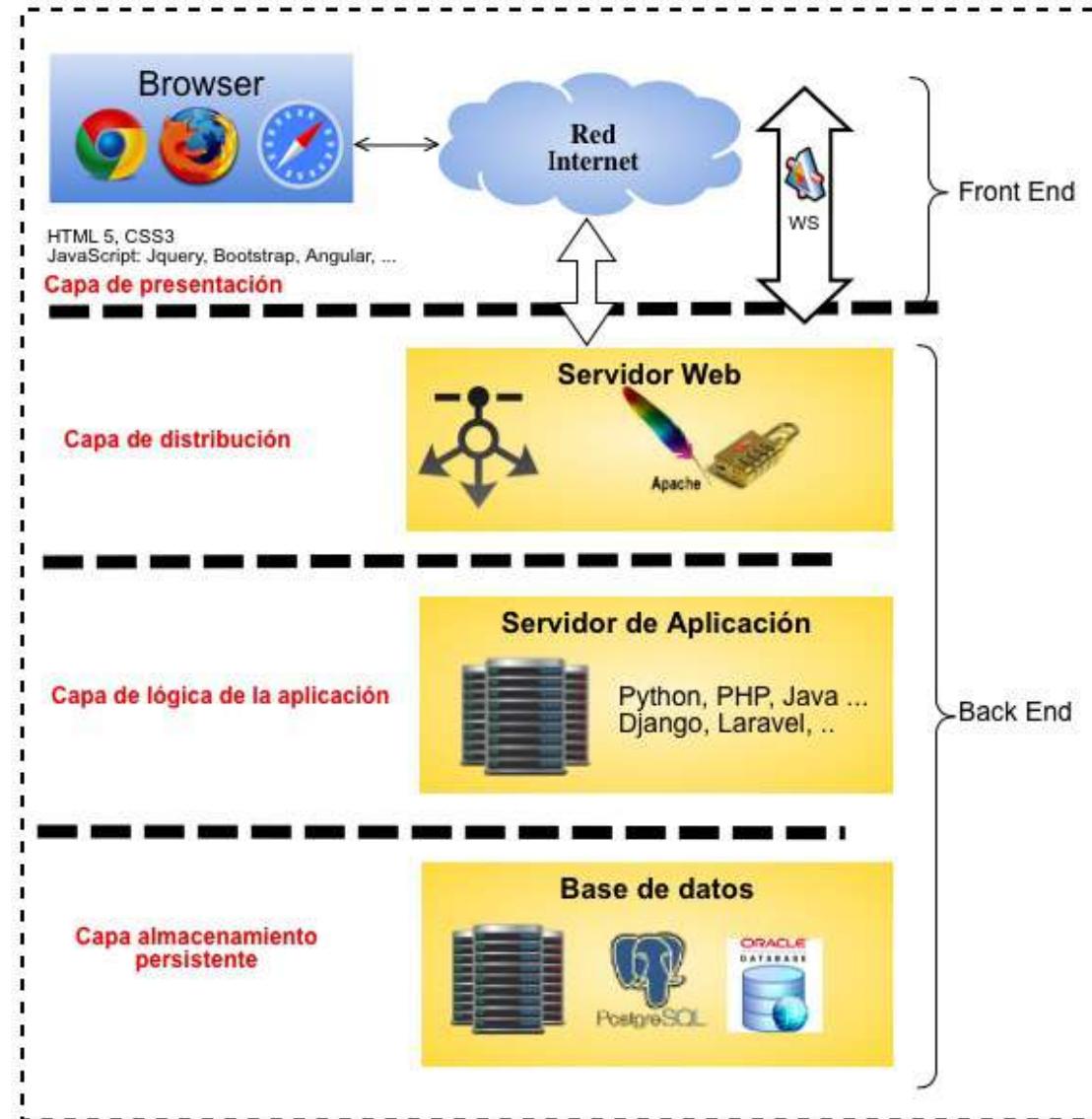
Ejemplo de procesos que se ejecutan en el backend

- Procesar la entrada del usuario
- Procesamiento de una solicitud de una página web entrante
- Ejecutar scripts (Python, PHP, ASP, JSP, etc.) para generar HTML
- Acceder a datos, desde una base de datos mediante consultas
- Almacenar o actualizar registros en una base de datos
- Cifrar y descifrar datos
- Manejo de cargas y descargas de archivos

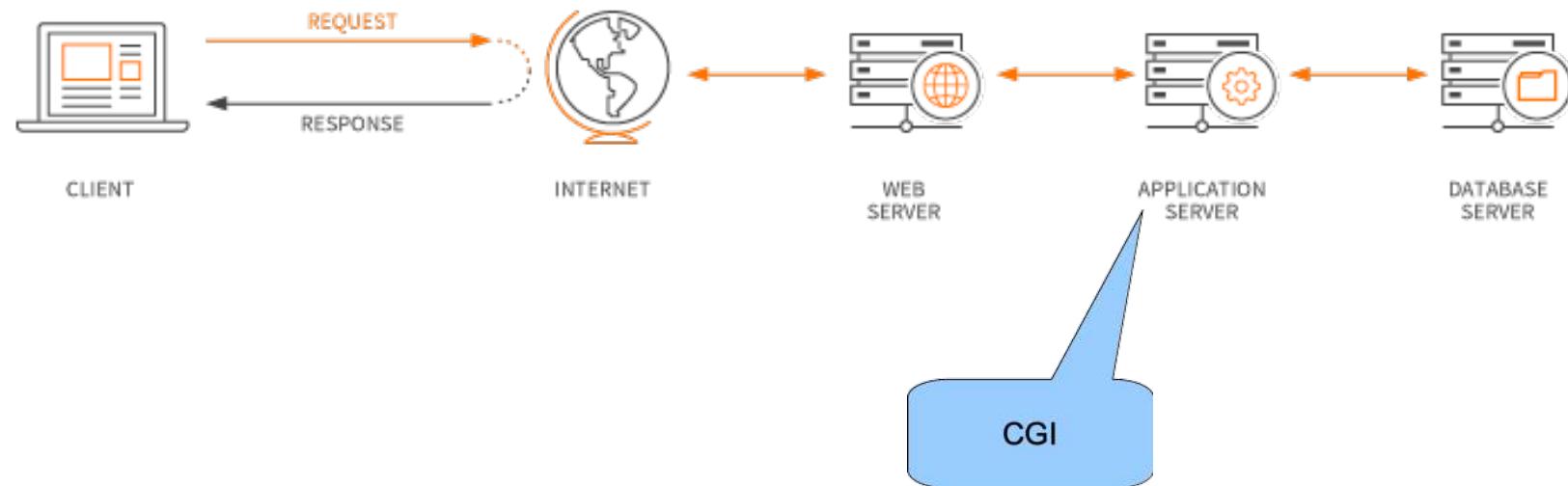


Backend

Vista por capas de una aplicación web.



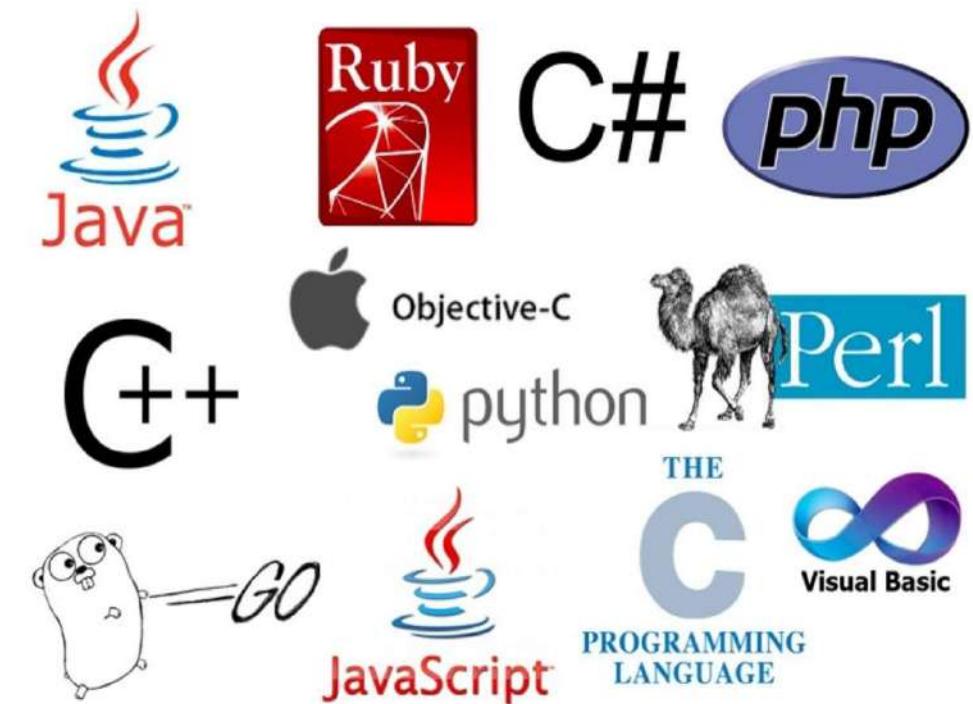
Ciclo Petición-Respuesta (Request-Response) en una aplicación web



Del lado Servidor, son numerosos los lenguajes de programación y frameworks disponibles.

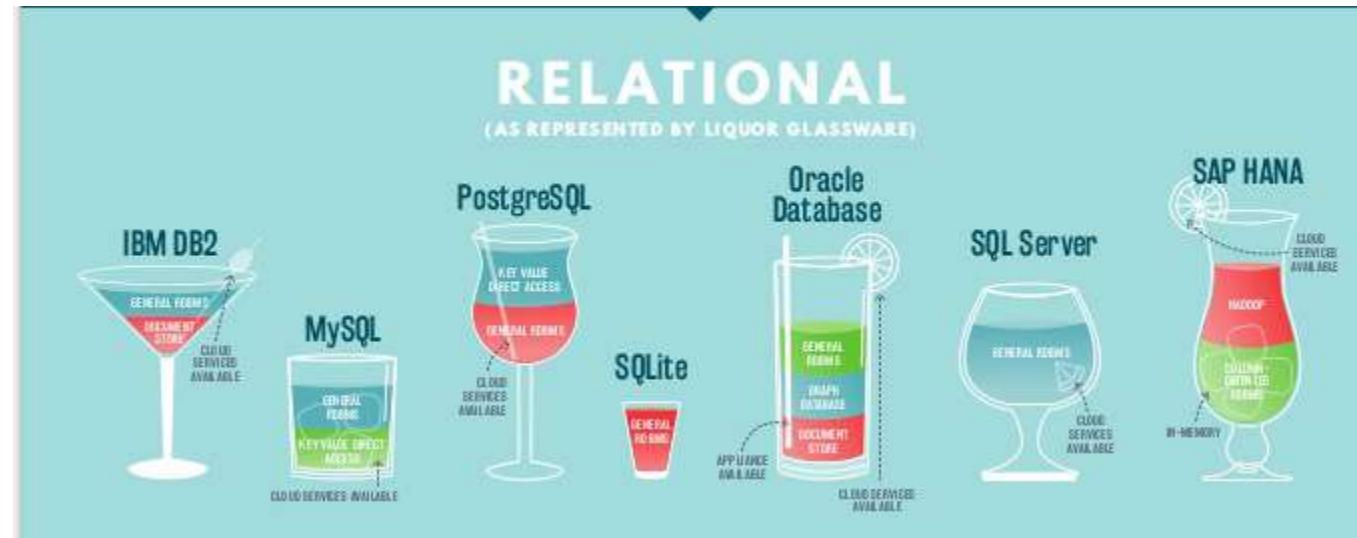
Criterios para seleccionar el lenguaje y frameworks:
(Algunos criterios)

- Dominio del problema
- Características del lenguaje
- Experiencia técnica de los desarrolladores
- Soporte y reputación de los lenguajes
- Herramientas asociadas (Frameworks, pruebas, calidad del código, etc)



Backend

Bases de datos



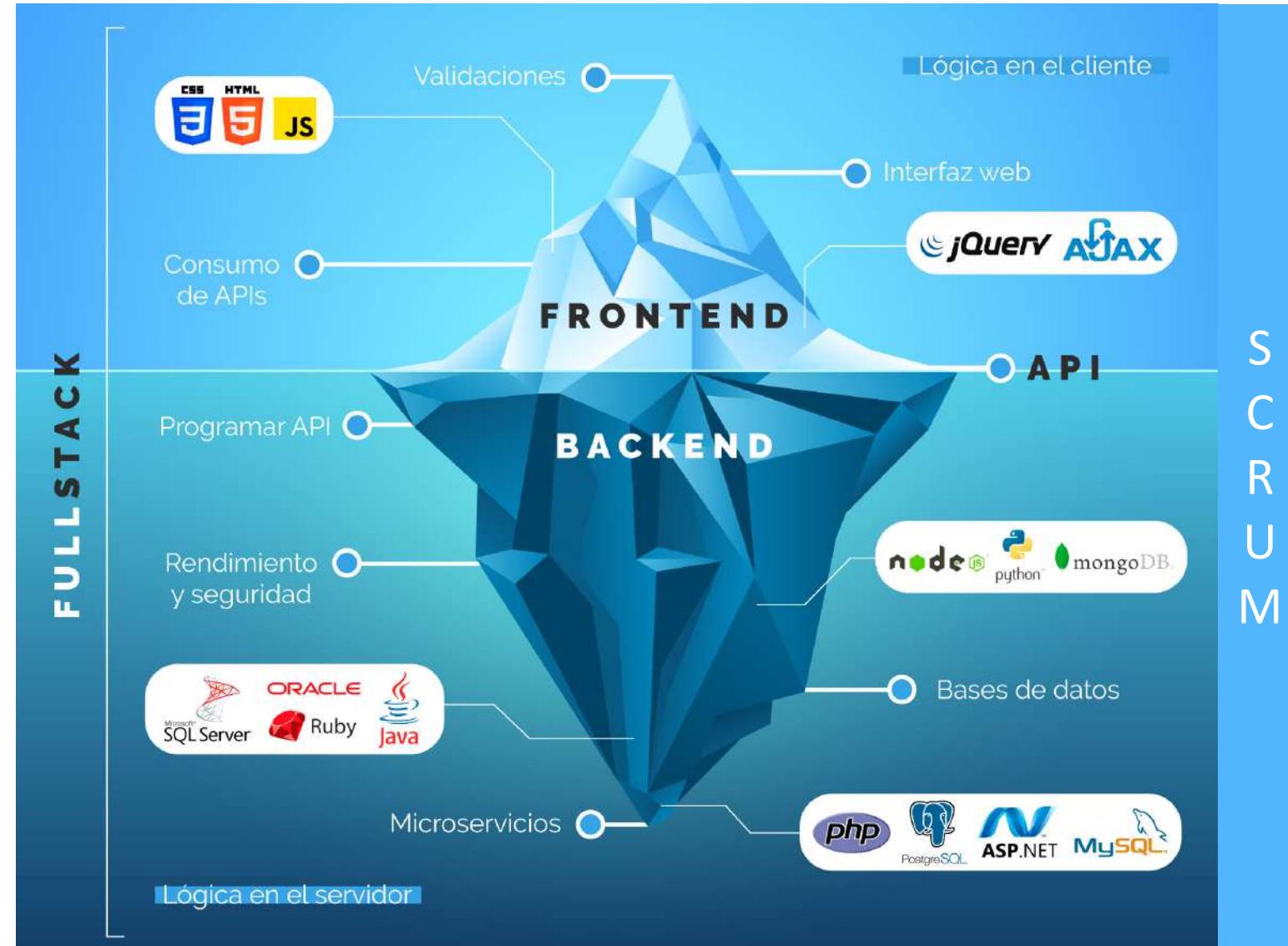
Backend

Bases de datos



Desarrollo de Software I

Desarrollador FullStack





Exposiciones

Pruebas unitarias en Python y Django.

Estándares de codificación en Python y Django.



Equipos para el proyecto del curso y exposiciones:

Trabajo en equipos del proyecto del curso



Desarrollo I

Economy of the
European Union

Gracias

