

Bootcamp desarrollo Frontend

Modulo 1 – Arquitectura de aplicaciones web



ESCUELA DE POSTGRADO
Y EDUCACIÓN CONTINUA
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Desarrollo Profesional

Profesor Carlos Marín Díaz

carlosmarind@gmail.com



ESCUELA DE POSTGRADO
Y EDUCACIÓN CONTINUA
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Desarrollo Profesional

Diseño y arquitectura de aplicaciones

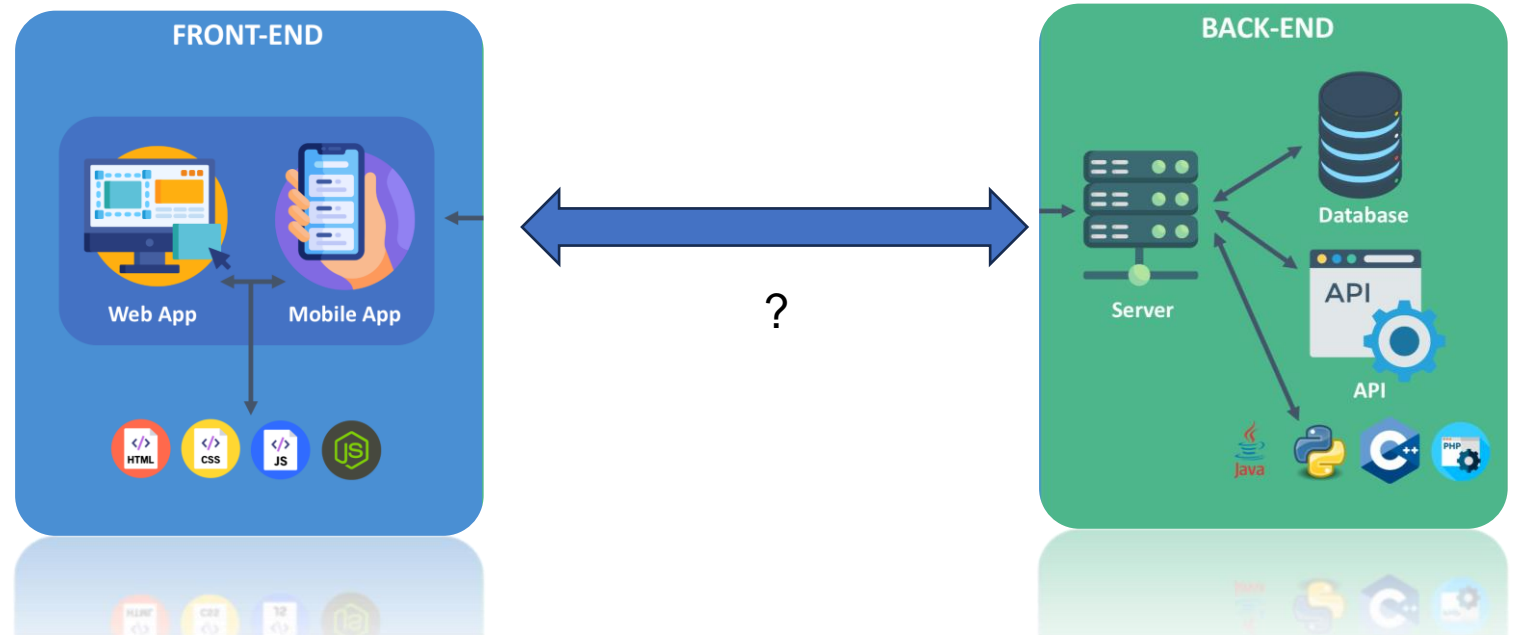
- En el principio de los tiempos, nuestra aplicación web era un solo paquete o modulo, el cual presentaba un módulo de interfaz de usuario y otro de acceso de datos.

Aplicación web o de escritorio



Diseño y arquitectura de aplicaciones

- Luego la rompimos en dos a modo de separar responsabilidades.
¿ahora estando separadas se comunican... como lo hacen?



Diseño y arquitectura de aplicaciones

- ¿Como se comunica un cliente con un servidor?



Cliente web
(frontend)



HTTP/HTTPS

Servidor (Backend)



Base de datos



API
(otro backend)

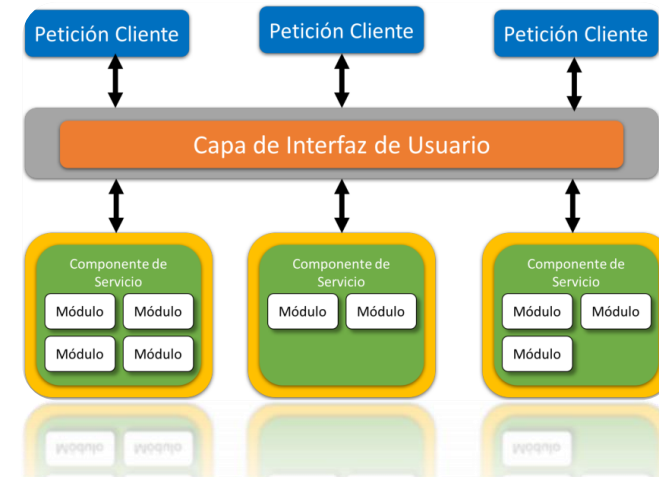


Mensaje
Json

```
{  
  "Empleado": {  
    "Nombre": "Juan",  
    "Edad": 30,  
    "Ciudad": "Madrid"  
  }  
}
```

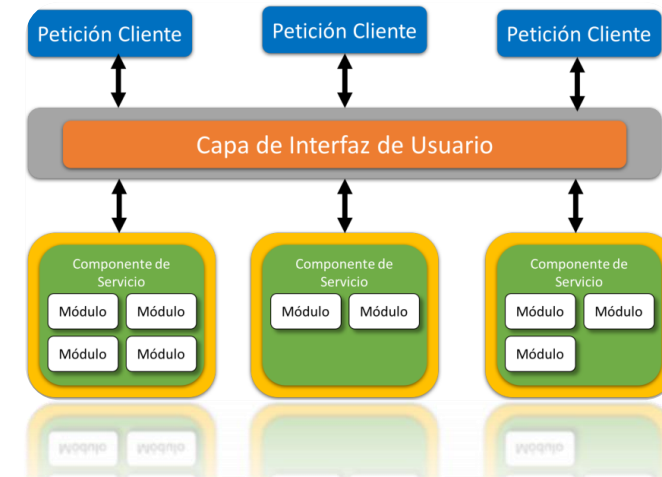
Arquitectura de aplicaciones web

Ahora que conocemos más de HTML, CSS y JavaScript, conocemos como se complementan, y sabemos cómo podemos comenzar a construir una página web básica, es importante que consideremos las distintas formas en que podemos diseñar y construir nuestras aplicaciones web. A esto llamaremos la arquitectura de nuestra aplicación, y mediante esta arquitectura, estableceremos definiciones iniciales y futuras de ordenamiento de ficheros, carpetas, lógica de negocio, tecnologías, librerías y una serie de directrices que serán las bases de como trabajaremos nuestro proyecto y como este evolucionara en el tiempo. Estas definiciones nos sirven para poner de acuerdo a nuestro equipo de desarrollo, facilitar el desarrollo, la mantenibilidad, la seguridad y el escalamiento entre otros beneficios.



Arquitectura de aplicaciones web

Existen distintos tipos de arquitecturas que se encargan de definir distintos aspectos de las aplicaciones y sistemas. Ahora mismo nos enfocaremos en la arquitectura de aplicaciones web en particular, lo cual nos servirá como herramienta para modelar nuestra aplicación web como interfaz de usuario y entender como esta puede o debe interactuar con el resto de la aplicación o sistema. En la arquitectura web, definiremos como se organizan y comunican los componentes de nuestra aplicación, como se dividirán y donde ira cada uno.

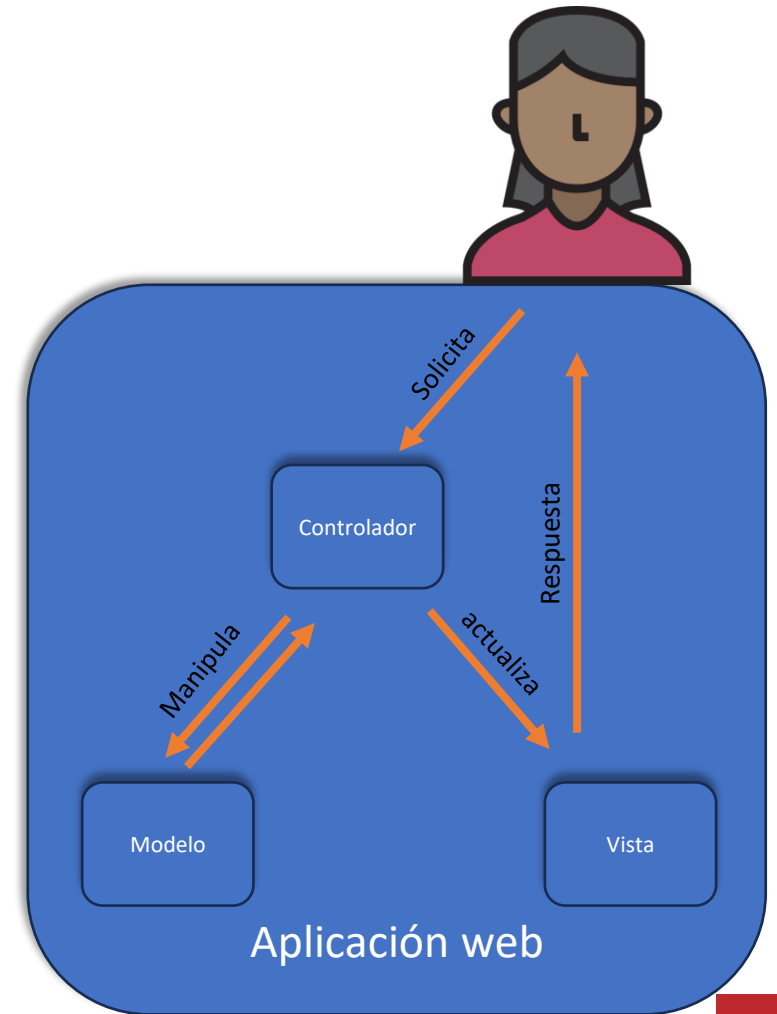


Arquitectura de aplicaciones web

Modelo vista controlador - MVC

Mediante este patrón de diseño arquitectónico, podemos separar nuestra aplicación en 3 componentes principales, cada uno de ellos con un propósito y finalidad clara. En este modelo de arquitectura de aplicaciones web, nuestra aplicación es una sola aplicación la cual internamente se divide en componentes para el que cada uno asume un rol en particular:

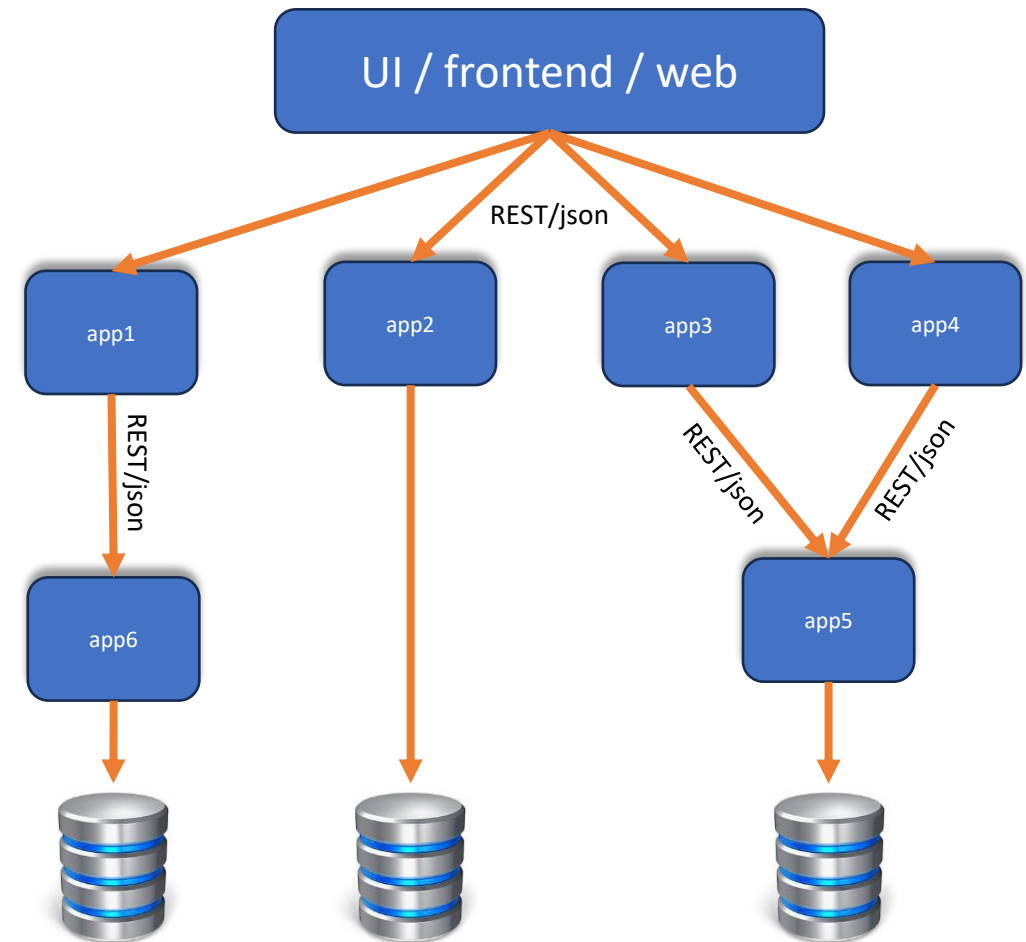
- **Controlador:** Es donde escribiremos el código que se encarga de recibir las solicitudes del usuario y mediante el cual se permite la interacción del usuario con el sistema. También es quien interactúa directamente con el modelo para consulta o actualización de información y la vista para actualizar las interfaces de usuario. Es posible incorporar lógica en el controlador basada principalmente en como este interactúa con el usuario como interfaz.
- **Modelo:** es el lugar donde escribiremos nuestra lógica de negocio y donde residirán nuestros datos o conexión a ellos. Aquí deben ir todas las reglas de nuestro negocio, procesos de información, manipulación y administración de los datos.
- **Vista:** Este componente es finalmente el que representa la interfaz de usuario y con el cual interactúan. Se define en HTML, considerando estilos y lógica de interacción con el usuario, y recibe las solicitudes del cliente probablemente a través de formularios web enviándolas posteriormente a un controlador.



Arquitectura de aplicaciones web

Arquitectura de microservicios

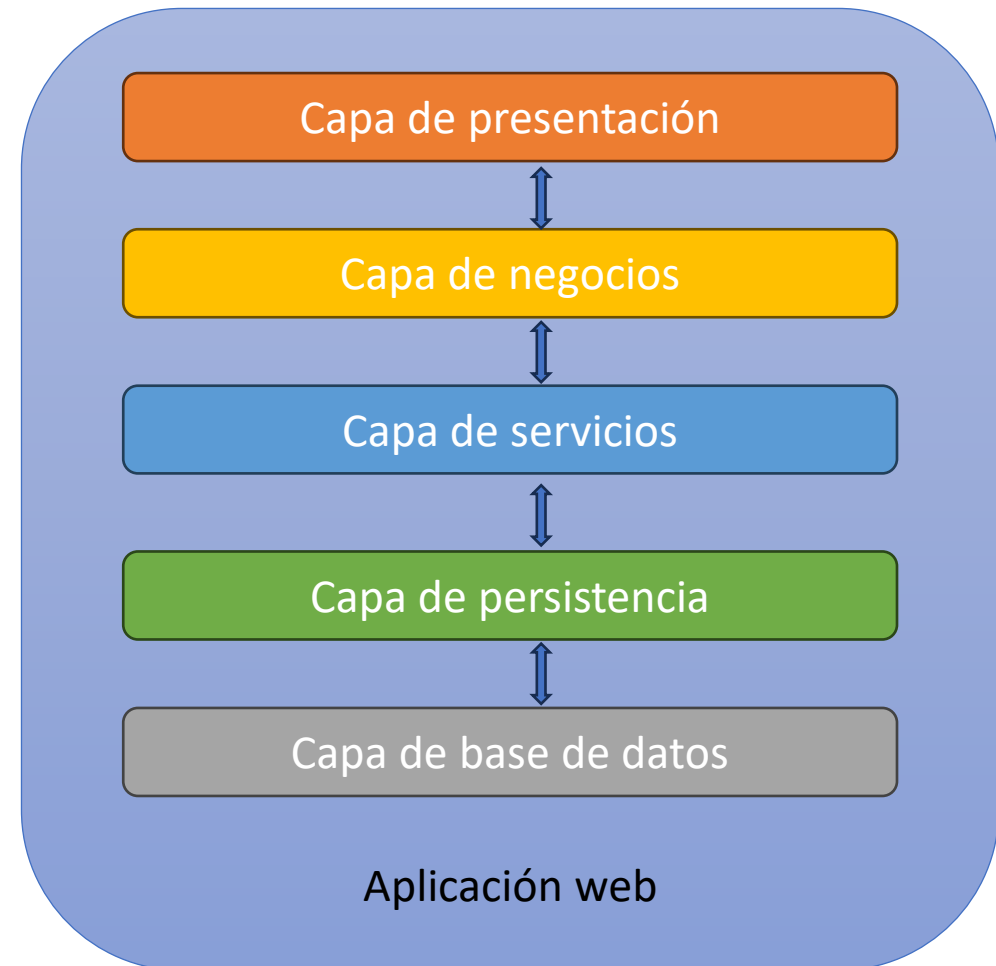
Este enfoque de construcción de aplicaciones busca descomponer la misma en bloques más pequeños e independientes unos de otros, y que puedan interactuar entre sí. Cada uno realiza una tarea específica y se comunican entre si generalmente a través de servicios web, como api 's REST. Al armar aplicaciones pequeñas con tareas específicas y particulares para cada una, nos da la oportunidad desarrollar la aplicación con equipos independientes, nos da escalabilidad, y por último el mantenimiento y puesta en producción es más simple dado que una modificación afecta solo al microservicio particular.



Arquitectura de aplicaciones web

Arquitectura de capas

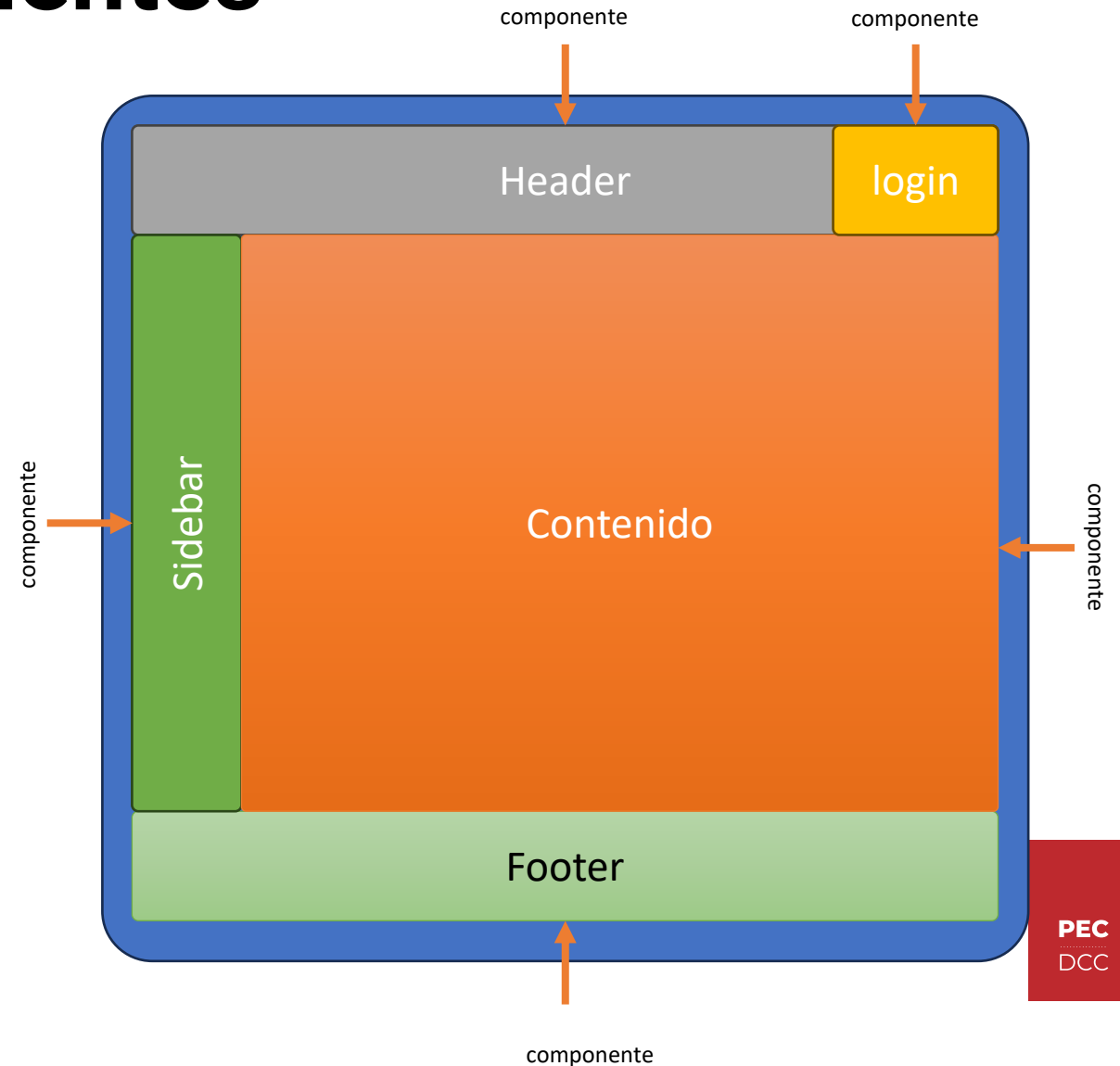
De este modo, ordenamos nuestra aplicación mediante el uso de capas lógicas, preocupándonos de que cada una de estas capas tenga una función específica y además de que cada capa solo se pueda comunicar con la capa anterior o siguiente. De esta forma separamos las responsabilidades, y cualquier cambio se facilita debido a la separación de responsabilidades.



Arquitectura de aplicaciones web

Arquitectura de componentes

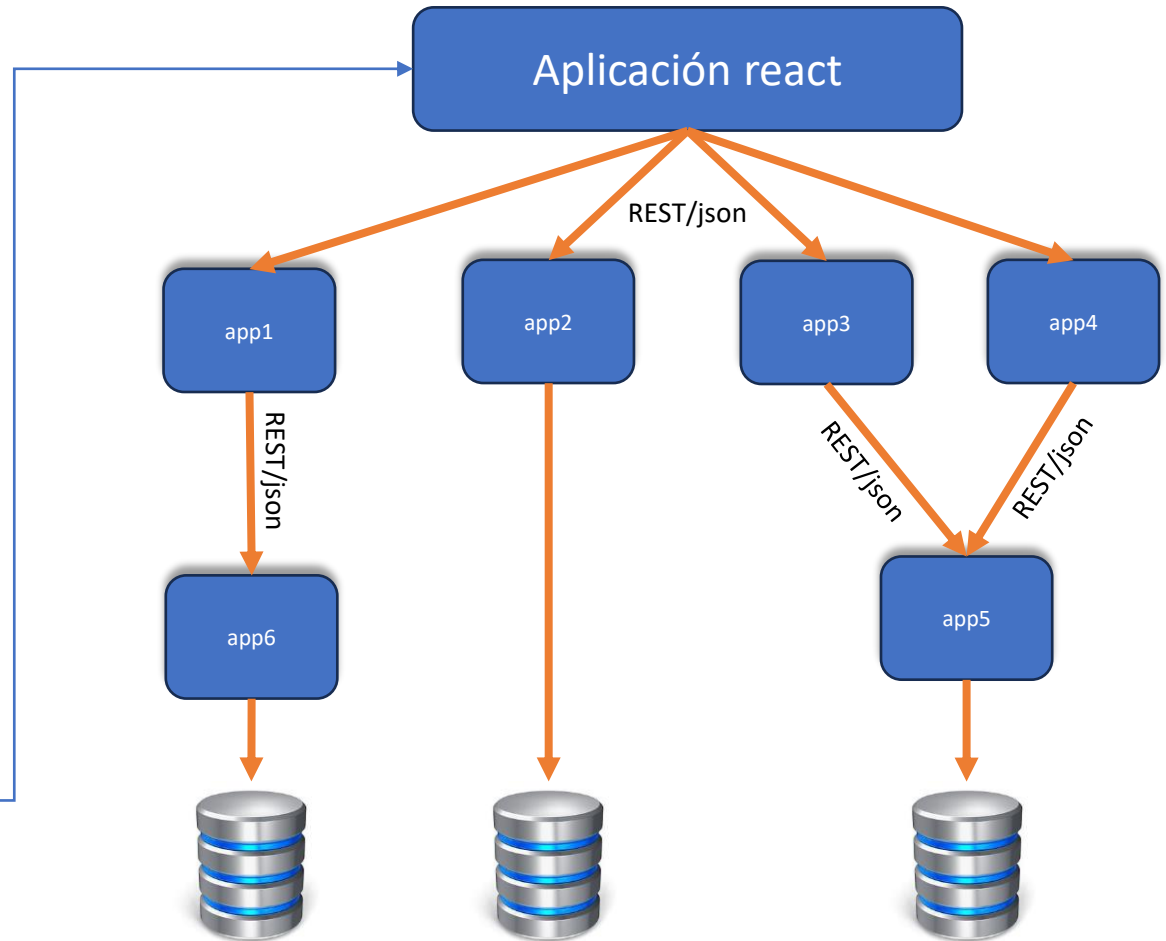
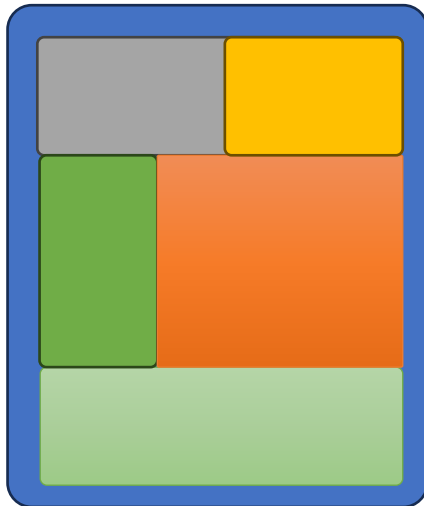
En este modelo de arquitectura web, nuestra aplicación se construye teniendo en mente la creación de componentes independientes y reutilizables, cada uno de ellos con su propia lógica, interfaz de usuario y datos asociados. Haciendo uso de estos componentes formamos una aplicación cada vez más grande. En este modelo, nosotros nos centraremos en la arquitectura web de componentes orientado particularmente al frontend o interfaz de usuario, donde existen frameworks o librerías que nos interesa conocer... por ejemplo react.



Arquitectura de aplicaciones web

Arquitectura de componentes

Como hablamos de arquitectura de componentes orientada al frontend. Nada impide utilizar un patrón de arquitectura de aplicación web que mezcle componentes y microservicios, por ejemplo.



Arquitectura de aplicaciones web

Arquitectura de componentes

Si bien es cierto que podemos implementar un desarrollo web de frontend basado en componentes utilizando distintas librerías o frameworks de mucho valor y amplio uso en el mundo del desarrollo de software como react y angular, hay tecnologías en común que se utilizan para aplicar este patrón de arquitectura en el frontend:

- **Herramientas de construcción y automatización:** utilidades que nos permitan agilizar el desarrollo, como por ejemplo npm, o npm scripts.
- **CSS en módulos:** posibilidad de aplicar estilos asociados únicamente a componentes individuales.
- **Bibliotecas de componentes:** Permiten la creación, organización y reutilización de componentes individuales, hacerlo de forma independiente por componente y luego juntarlos para armar una aplicación.
- **Pruebas unitarias:** Permite generar pruebas en base a componentes individuales, sin necesidad de probar toda la aplicación, existen librerías como jest, mocha o chai que nos permiten hacerlo.
- **Gestión de estados:** Proveen la capacidad de conocer el estado cualquier componente en un momento dado y también de modificarlo.
- **Gestión de rutas:** Cuando una aplicación tiene muchas páginas o vistas, se necesita de un enrutador que nos pueda dirigir de una a otra como usuario a través del navegador.
- **Bundles y transpiladores:** Herramientas que nos permitan generar un paquete final el cual reúna todos nuestros componentes y lo presente al navegador como un paquete unificado. También puede optimizar nuestro código o hacerlo compatible con versiones de navegadores más antiguas al generar el paquete final.

Arquitectura de aplicaciones web

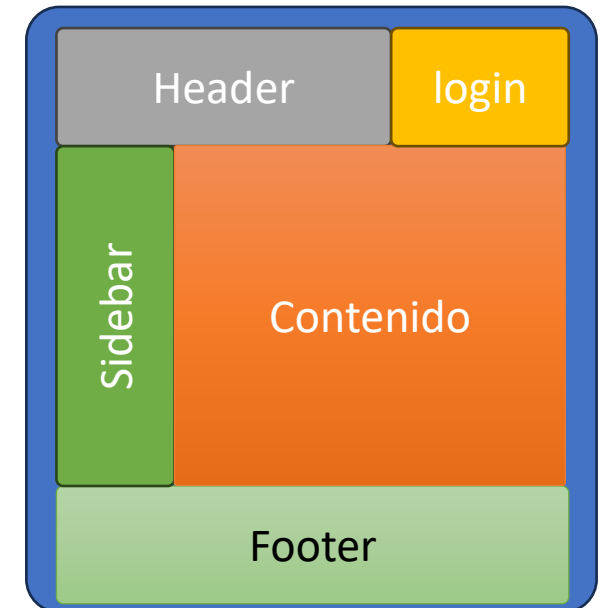
Arquitectura de componentes

Actividad en clase.

Construyamos dos páginas web utilizando HTML y css las cuales contengan la estructura mencionada antes.

Además, cada una de ellas debe contar con los siguientes criterios.

- Debe tener una lista de entradas en el sidebar como entradas de menú (enlaces). Uno de estos enlaces debe llevar a la segunda página web, y viceversa.
- Debe tener una imagen y un texto del sitio en el header
- Debe contar con una lista de enlaces en el footer además de contar con enlaces a redes sociales de Facebook, twitter e Instagram , los enlaces a redes sociales deben ser con imágenes. Al menos uno de los enlaces del footer debe llevar a la página 2 y viceversa.
- En el contenido deben incorporar una imagen y un texto de tipo "ipsum lorem". Deben variar la imagen y el texto en ambas paginas para poder diferenciarlas.





dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE



fcfm

ESCUELA DE POSTGRADO
Y EDUCACIÓN CONTINUA
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Desarrollo Profesional