

¿Qué es REACT?

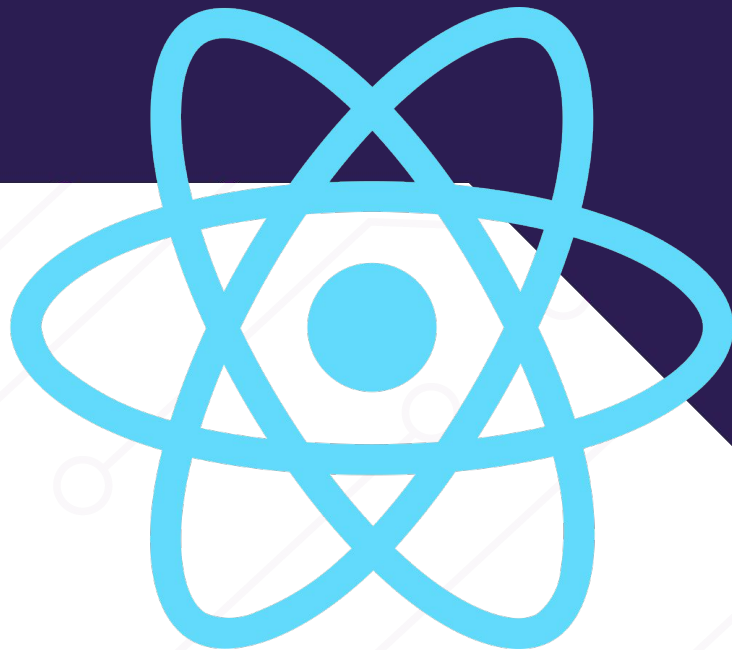
DEV.F
DESARROLLAMOS(PERSONAS);

dev

REACT

Es una biblioteca JavaScript de código abierto para desarrollar interfaces de usuario. Fue lanzado en 2013 y desarrollado por Facebook, quien actualmente también lo mantiene junto con una comunidad de desarrolladores y empresas independientes.

Desde su lanzamiento, su uso se ha incrementado notablemente, convirtiéndose, a día de hoy, en una de las tecnologías front-end más utilizadas.





Framework

Conjunto de herramientas que trabajan en un proyecto completo bajo reglas específicas.

Tiene funcionalidades integradas para que no necesites librerías externas.

La compatibilidad de sus funcionalidades está asegurada.

El framework define la manera en la que debes implementar y desarrollar el proyecto



Librería

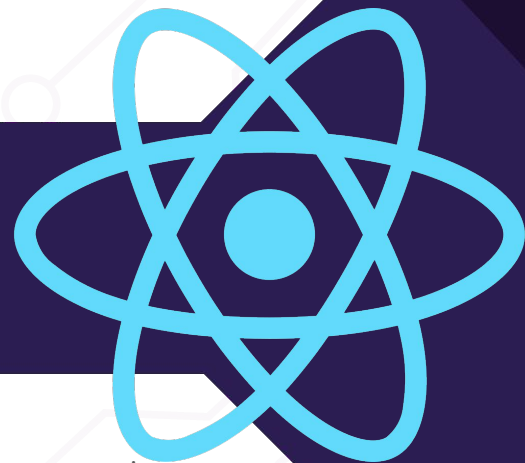
Herramienta con una sola utilidad específica.

Da la libertad de usar las librerías que desees en la estructura que quieras.

Se debe de controlar la compatibilidad de cada librería con las otras.

Se pueden usar varias librerías según las necesidades

¿Cómo funciona REACT?



Para comprender cómo funciona React es clave tener conocimiento de tres conceptos básicos:

- HTML: la semántica, estructura e información de la página web; es decir, su esqueleto.
- CSS: la apariencia de nuestra página web.
- JavaScript: básicamente es el cerebro de nuestra página. Determina qué hacer en función de lo que sucede en ella.

Sin embargo, antes de React estos conceptos funcionaban por separado, en diferentes archivos y carpetas, por lo que escalar y extraer diversas partes del código para reutilizar o migrar funcionalidades era más complicado.

¿Qué es DOM?

Las siglas DOM significan Document Object Model, o lo que es lo mismo, la estructura del documento HTML. Una página HTML está formada por múltiples etiquetas HTML, anidadas una dentro de otra, formando un árbol de etiquetas relacionadas entre sí, que se denomina árbol DOM (o simplemente DOM).



¿Qué es JSX?

Consiste en una extensión de la sintaxis de JavaScript que permite que compilamos un objeto JavaScript para mapear un elemento del DOM (Objeto del Documento). A través de JSX se crea un DOM virtual (que es una representación en memoria del DOM) que pesa muy poco y usa menos recursos.

Con esto, cuando React es informado de un cambio de estado, vuelve a ejecutar esas funciones y determina una nueva representación virtual de dicha página. Posterior a esto, traduce automáticamente ese resultado en los cambios del DOM necesarios, reflejando así la nueva presentación de la página.

¿Por qué debes conocer React JS?

Principalmente por las novedades que otorga, como por ejemplo:

- Databinding unidireccional: promueve el flujo de datos en un solo sentido facilitando la detección de errores en aplicaciones complejas, simplificando la carga de datos y haciendo más sencilla la fase de testeo.
- DOM virtual: que evita renderizar todo el DOM cada vez que hay un cambio en la interfaz de la aplicación.

Sin embargo, antes de sumergirte en el conocimiento acerca de qué es React JS y utilizarlo, es clave que estés actualizado en lo que respecta a JavaScript, ya que es uno de los pilares de esta librería.

Para ayudarte con esto y para que te conviertas en un experto que está a la vanguardia con esta tecnología que permite mayor rapidez con menos recursos informáticos, te invitamos a inscribirte a nuestro curso de FrontEnd con JavaScript.

The logo for DEV.FL, featuring the text "DEV.FL" in a bold, white, sans-serif font. The "F" is stylized with a grid-like pattern. The logo is centered within a dark blue diamond shape.

DEV.FL

Single Page App

¿Qué es una SPA?

¿Qué son o cómo se comen esas cosas que nos está enseñando el sensei Dani?

Una **Single-Page Application (SPA)** es un tipo de aplicación web que ejecuta todo su contenido en una sola página.

Funciona cargando el contenido **HTML**, **CSS** y **JavaScript** por completo al abrir la web. Al ir pasando de una sección a otra, solo necesita cargar el contenido nuevo de forma dinámica si este lo requiere, pero no hace falta cargar la página por completo. Esto mejora los tiempos de respuesta y agiliza mucho la navegación, favoreciendo así a la experiencia de usuario.

Ventajas y desventajas

Que sea una única página no significa que tenga un scroll eterno o que el contenido no esté estructurado. La realidad es que, dentro de esa propia página, podemos obtener **distintas vistas** para cada apartado de la web. Estas serían equivalentes a las diferentes páginas de las que se compone una **MPA (Multi Page Application)**.

Está claro que la característica más destacada de las SPA es la navegación ágil y fluída. Todo el contenido inicial se carga una única vez, por lo que el tiempo de espera en el paso de una sección a otra es prácticamente inexistente. Estas utilizan menos recursos del servidor, lo que, además de aumentar la velocidad, es beneficioso también para conexiones de Internet lentas. En definitiva, el single-page resuelve uno de los puntos clave de la experiencia de usuario

Tech Stacks

DEV.F
DESARROLLAMOS(PERSONAS);

dev

¿Qué es un stack de tecnología?

Un stack tecnológico, también llamado stack de soluciones o ecosistema de datos, es una lista de todos los servicios que se utilizan para desarrollar y ejecutar una aplicación.

Conjunto de tecnologías que utiliza una organización para crear una aplicación, dichas tecnologías pretenden cubrir las necesidades mediante lenguajes de programación, frameworks, librerías, patrones, servidores, soluciones UI/UX, software y más herramientas utilizadas por sus desarrolladores.



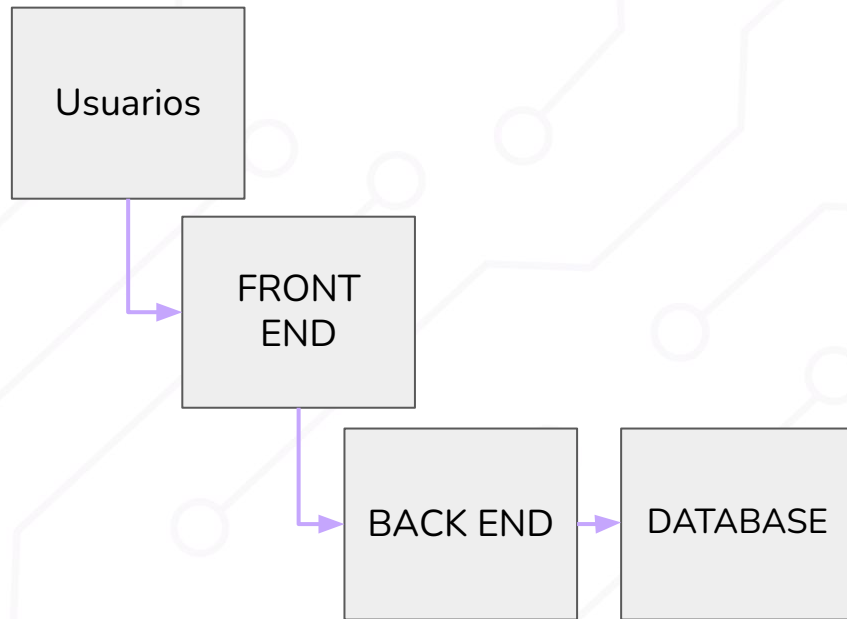
¿Qué contiene un stack tecnológico?

DEV.F
DESARROLLAMOS(PERSONAS);

dev

¿Qué contiene un stack tecnológico?

Todos los stacks tecnológicos se dividen entre el back-end y el front-end, además conocido como server-side y client-side. Una vez que una solicitud de trabajo necesita de un ingeniero con vivencia en la zona de back-end, front-end y full-stack, se refiere a el fragmento de la tecnología en la que el solicitante se especializa de manera particular.



¿Cómo construir un stack tecnológico?

DEV.F
DESARROLLAMOS(PERSONAS);

dev

how to build a tech stack?

Para formar un tech stack es una buena iniciativa aprender a disponer de una mezcla reducida de herramientas para mejorar el manejo y quitar las deficiencias del modelo de negocio con novedosas herramientas

Sin embargo, para diseñar un stack tecnológico eficaz no basta con escoger los instrumentos de programa e implantarlas.

- ❑ ¿Qué tipo de herramientas de software necesita la empresa?
- ❑ ¿Qué procesos van a mejorar con la automatización?
- ❑ ¿En qué medida van a ser idóneas las herramientas que has elegido a largo plazo?
- ❑ ¿Las herramientas van a funcionar correctamente en conjunto?

Tipos de stacks en la Industria

- LAMP
- ASP.NET
- MEAN
- MEARN



LAMP Stack

¿Qué es LAMP?

Arquitectura de LAMP

- Linux
- Apache
- MySQL
- PHP

¿Cómo funciona?

PHP > MySQL > Apache > Navegador

Beneficios para el desarrollador

- Open Source
- Pocas Restricciones
- Simplifica el Desarrollo



ASP.NET Stack

¿Qué es ASP.NET?

Arquitectura de ASP.NET

- Windows Server
- Internet Information Services (IIS)
- MSSQL

¿Cómo funciona?

Web Forms/MVC/API > MSSQL > Navegador

Beneficios para el desarrollador

- Desarrollo ágil y Fiable.
- Componentes prefabricados.
- Administrador de paquetes.



MEAN Stack

¿Qué es MEAN?

Arquitectura de MEAN

- Mongo
- Express
- Angular
- NodeJS

¿Cómo funciona?

Angular > MVC > Express > Mongo

Beneficios para el desarrollador

- Open Source
- Pocas Restricciones
- Se domina un solo lenguaje para frontend y backend



MERN Stack

¿Qué es MERN?

Arquitectura de MERN

- Mongo
- Express
- ReactJS
- NodeJS

¿Cómo funciona?

ReactJS > MVVM > Express > Mongo

Beneficios para el desarrollador

- Open Source
- Popular
- Se domina un solo lenguaje para frontend y backend



**Y como me dijo el del transporte
público**

**hasta aquí llegamos
joven**



DEV.F.
DESARROLLAMOS(PERSONAS);

dev