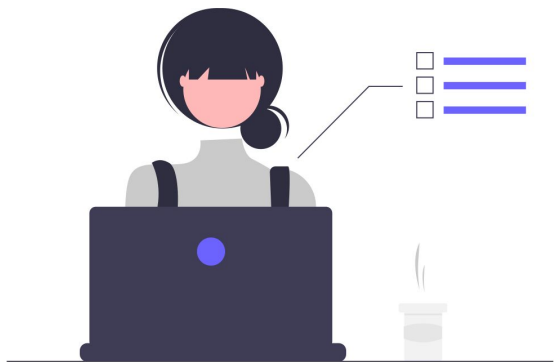


¿Por qué Node.js para fullstack?

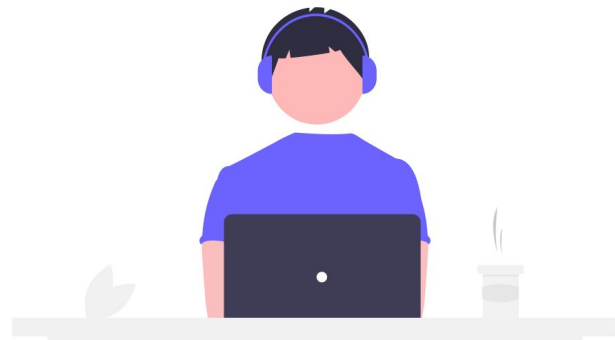
DEV.F
DESARROLLAMOS(PERSONAS);

dev

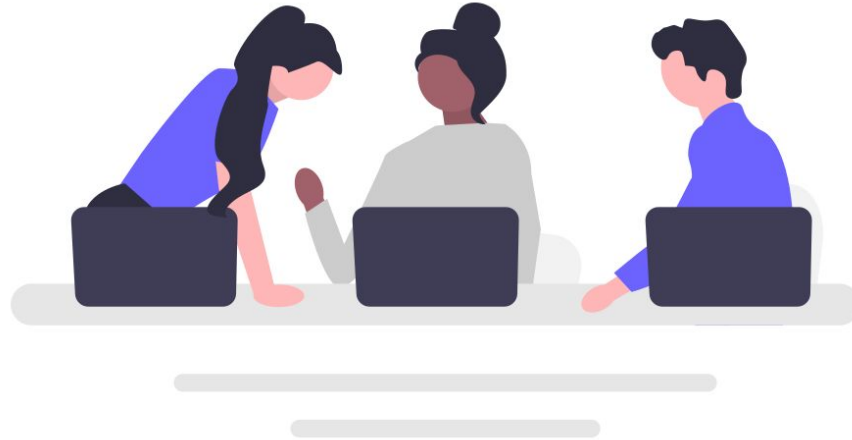
Desarrollador vs Programador



Los desarrolladores suelen tener un ámbito de trabajo más amplio que los programadores, suelen gestionar proyectos de software.



Los programadores generalmente se enfocan principalmente en escribir, depurar y probar su código. Un programador puede tener un mayor conocimiento técnico.



Los desarrolladores y programadores comparten algunas de sus responsabilidades laborales. Ambos profesionales crean, revisan, prueban y solucionan problemas de código, pueden organizar datos de software, utilizar métodos de encriptación o seguridad para proteger su software y verificar la compatibilidad de su software con varios sistemas operativos.

En la actualidad JavaScript es posiblemente la manera más sencilla y rápida de transformarse en un Full-stack developer. Era esperable que mucha gente se encontrará buscando probar Node.js justo luego de los fundamentos de JavaScript.



Un desarrollador Full Stack JavaScript es el profesional capaz de producir proyectos tanto en el Front End como en el Back End de una aplicación o desarrollo web, usando lenguaje de programación JavaScript, siendo capaz de dar un plan de inicio a fin.

Por tener los 3 conocimientos, pertenece a los perfiles más valorados en el mercado.




The logo consists of two overlapping yellow squares. The top square contains the text 'ES6' in a bold, black, sans-serif font. The bottom square, which is partially obscured by the top one, contains the text 'JS' in the same bold, black, sans-serif font. The background of the slide features a light gray circuit-like pattern with lines and circles, and a purple line with a circle at the top left.

ES6

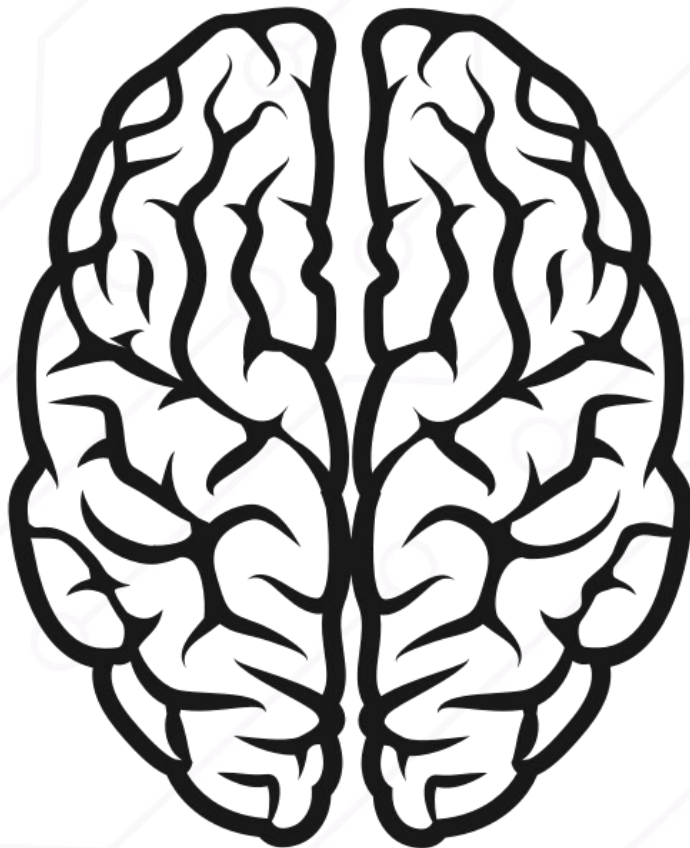
JS

Con el razonamiento Front-End tendrás la posibilidad de llevar a cabo los nuevos estándares ECMA de JS en la construcción de una app web para manejar información y datos, consumir APIs y usar pruebas unitarias. Además construirás interfaces de cliente declarativas para la web con React.



Además con el aprendizaje obtenido en Back-End usarás la consola, administrarás servidores y bases de datos, desarrollarás y publicarás APIs.

Con ambos conocimientos en development anteriormente mencionados, como Full Stack tendrás la posibilidad de generar comunicación entre una aplicación del todo desplegada y en producción, una base de datos y el servidor.

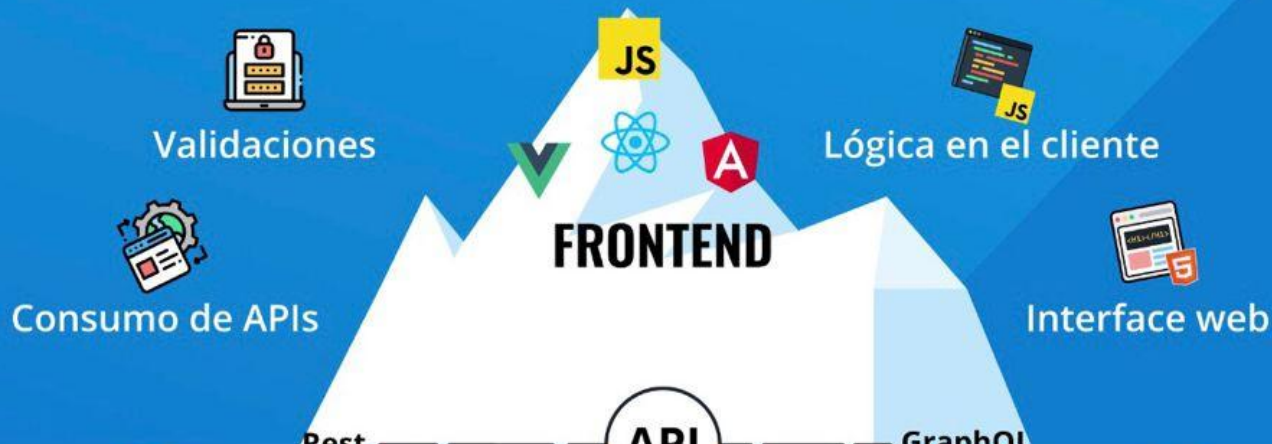


FRONT-END

DEV.F
DESARROLLAMOS(PERSONAS);

dev

¿QUÉ ES BACKEND Y FRONTEND?



FRONT-END Developer

Un desarrollador front-end trabaja la interfaz de usuario desde el punto de vista del código, para que la interacción con el sistema sea posible. Por lo tanto, se encarga de la parte visual de la web (de todo aquello que puedes ver en tu explorador) haciendo que su diseño sea intuitivo y atractivo.

front-end no es un diseñador, sino que recoge los documentos y directrices del equipo de diseño para trasladarlas a código

trabaja desde el lado del cliente, interpretando y adaptando lenguajes de desarrollo web para crear la experiencia final de ese cliente/usuario: es la forma en la que el diseño se implementa realmente en la web.

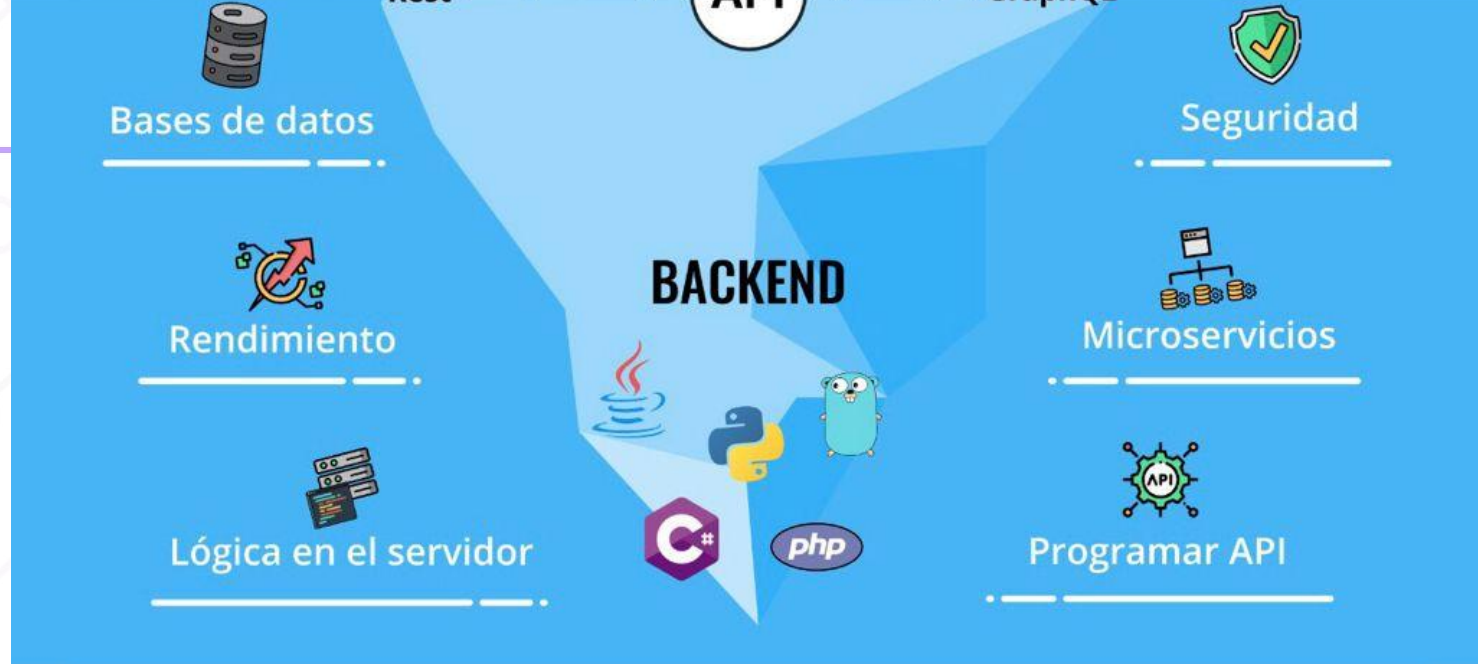
THE FRONT-END SPECTRUM



BACK-END

DEV.F
DESARROLLAMOS(PERSONAS);

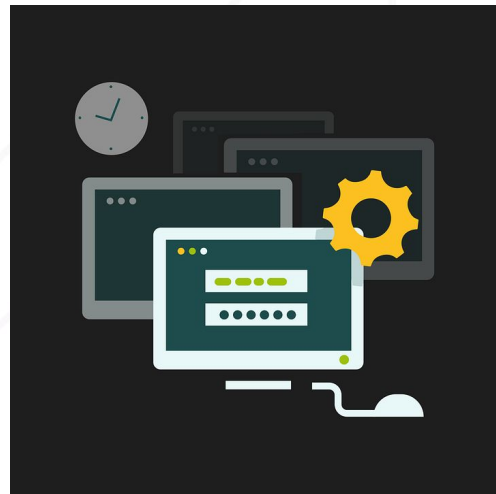
dev



BACK-END Developer

Un desarrollador backend es el individuo encargado del desarrollo de una web, app o programa, se ocupa de la “parte trasera”, de la espalda; o sea, para interactuar y trabajar con el servidor. Sus funcionalidades son diseñar, comprobar y componer todo para su buen funcionamiento.

- Optimizar las aplicaciones para que rindan mejor y de forma más rápida.
- Garantizar la seguridad de todo el entorno.
- Solucionar posibles errores que se vayan detectando.
- Gestionar e integrar las bases de datos necesarias.
- Trabajar mano a mano con el front end developer.
- Realizar controles de calidad sobre el producto.
- Elaborar mejoras para las actualizaciones.
- Codificar en diferentes lenguajes de programación.



Top 12 Backend Technologies To Look For In 2022





**System
architecture**



Servers



Database



Data analysis



Security



Frameworks



APIs



Scalability



**Operating
system**



Business logic

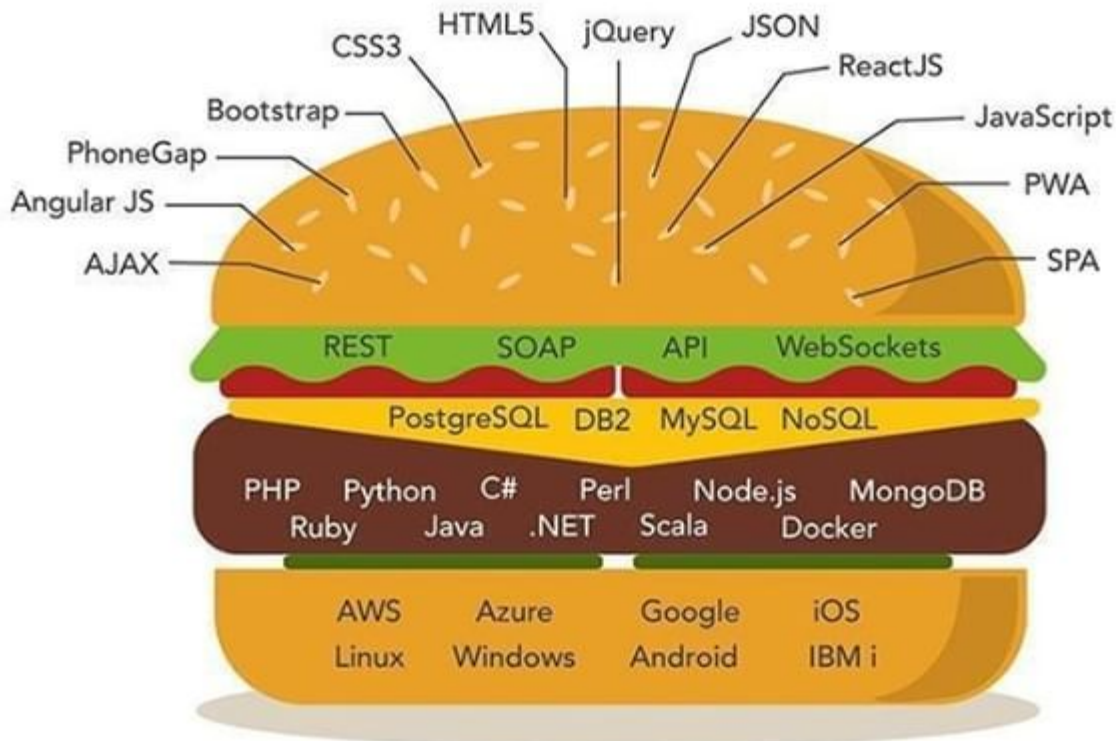
BACKEND IS:

FULLSTACK

DEV.F
DESARROLLAMOS(PERSONAS);

dev

"The Full-Stack"



Un desarrollador Full Stack JavaScript es el profesional capaz de crear proyectos tanto en el Front End como en el Back End de una aplicación o desarrollo web, utilizando lenguaje de programación JavaScript, siendo capaz de entregar un proyecto de inicio a fin. Por tener los tres conocimientos, es uno de los perfiles más valorados en el mercado.

Con el conocimiento Front-End se implementan los estándares ECMA de JS en la creación de una app web para manejar información y datos, consumir APIs y utilizar pruebas unitarias.

Se pueden desarrollar interfaces de usuario declarativas para la web con algún otro framework. En cuanto Back-End se utiliza la consola, administra servidores y bases de datos, desarrollar y publicar APIs.

Con los dos conocimientos en development antes mencionados, como Full Stack se crea comunicación entre una aplicación completamente desplegada y en producción, una base de datos y el servidor.

Patrones de arquitectura

DEV.F
DESARROLLAMOS(PERSONAS);

dev

¿Qué es un patrón de arquitectura?

Un patrón de diseño es básicamente una forma reutilizable de resolver un problema común en el desarrollo de software.

Si la forma de solucionar este problema se puede extraer, explicar y reutilizar en múltiples ámbitos y en muchas ocasiones, en ese caso nos hemos encontrado con un patrón de diseño.



Por qué utilizar patrones de diseño

Utilizar patrones de diseño es interesante porque de esta forma vas a poder realizar aplicaciones más robustas y más fáciles de mantener, gracias a estas reglas que se van a cumplir al utilizar estos patrones.

Los patrones nos ayudan a cumplir unos principios y reglas del diseño del software que hacen que se obtenga un código mucho más robusto.



¿Qué tipos de patrones de diseño podemos encontrar?

Se puede hablar de tres grandes bloques de patrones de diseño:

- **Patrones creacionales**, que nos van a solucionar problemas de creación de instancias, además de encapsular y abstraer dicha creación.
- **Patrones estructurales**, que describen cómo utilizar estructuras de datos complejas a partir de elementos más simples.
- **Patrones de comportamiento**, que especifican el comportamiento entre objetos de nuestro programa.
- **Patrones de arquitectura**, paquete de decisiones de diseño que se encuentra repetidamente en la práctica, tiene propiedades bien definidas que pueden ser reutilizadas y describe una clase de arquitecturas.

O'REILLY®



Fundamentals of Software Architecture

An Engineering Approach

Mark Richards & Neal Ford

CENGAGE
Learning

Humberto Cervantes Maceda
Perla Velasco-Elizondo
Luis Castro Careaga

ARQUITECTURA DE SOFTWARE

Conceptos y ciclo de desarrollo



DEV.F

The logo consists of the text 'DEV.F!' in a bold, white, sans-serif font. The 'F' is stylized with three small squares at its base. The logo is centered within a dark blue diamond shape.

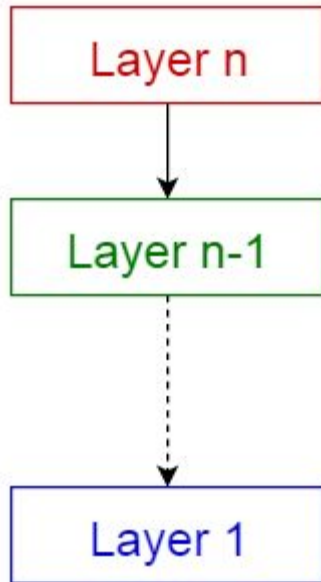
DEV.F!

**Patrones comunes de
arquitectura de software
de la muerte pero en
realidad si uno le sabe
puede adaptarse a ellos.**

Patrón de capas

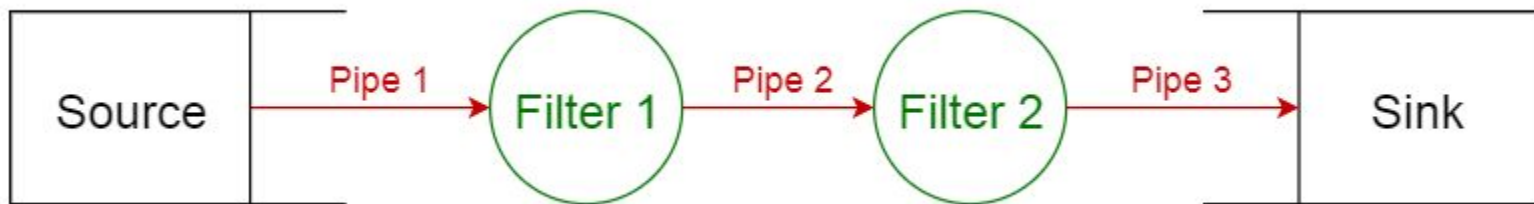
Se utiliza para estructurar programas que se pueden descomponer en grupos de subtareas, cada una de las cuales se encuentra en un nivel particular de abstracción. Cada capa proporciona servicios a la siguiente capa superior.

- Capa de presentación
- Capa de aplicación
- Capa de lógica de negocios
- Capa de acceso a datos



Patrón de filtro de tubería

Este patrón se puede usar para estructurar sistemas que producen y procesan una secuencia de datos. Cada paso de procesamiento se incluye dentro de un componente de filtro . Los datos que se procesarán se pasan a través de las tuberías . Estas tuberías se pueden utilizar para el almacenamiento en búfer o con fines de sincronización.

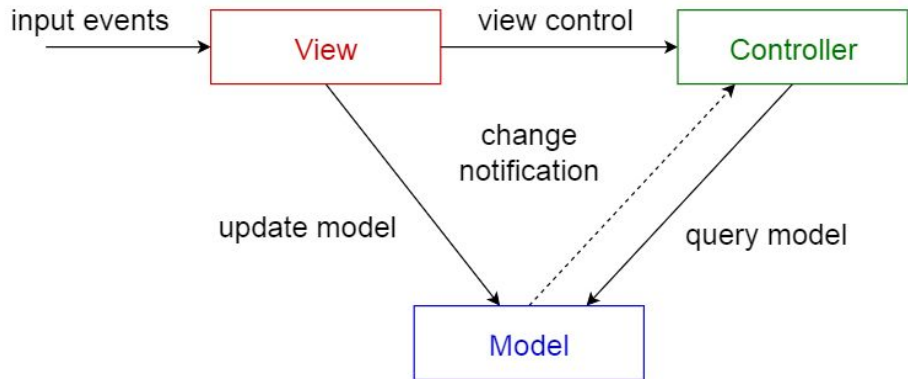


Patrón MVC

Este patrón, también conocido como patrón MVC, divide una aplicación interactiva en 3 partes.

1. modelo — contiene la funcionalidad y los datos básicos.
2. vista : muestra la información al usuario (se puede definir más de una vista).
3. controlador : maneja la entrada del usuario.

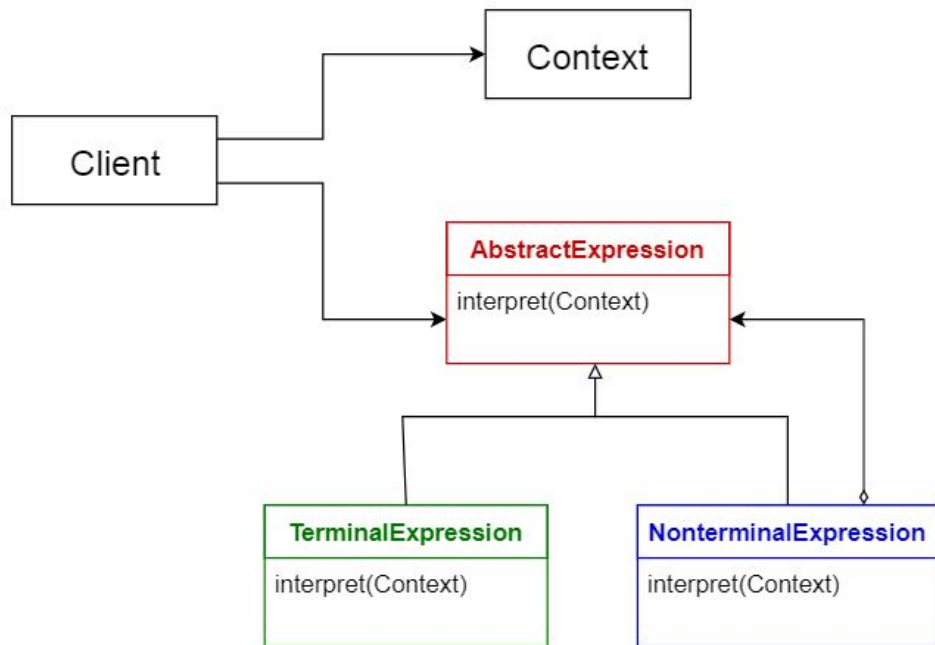
Esto se hace para separar las representaciones internas de información de las formas en que se presenta y acepta la información del usuario. Desacopla los componentes y permite la reutilización eficiente del código.



Patron interprete

Este patrón se usa para diseñar un componente que interpreta programas escritos en un lenguaje dedicado.

Específica principalmente cómo evaluar las líneas de programas, conocidas como oraciones o expresiones escritas en un idioma particular. La idea básica es tener una clase para cada símbolo del idioma.



Patron de arquitectura Cliente-Servidor

Uno de los estilos arquitectónicos distribuidos más conocidos, el cual está compuesto por dos componentes, el proveedor y el consumidor. El proveedor es un servidor que brinda servicios o recursos los cuales son consumidos por el Cliente.

En una arquitectura Cliente-Servidor existe un servidor y múltiples clientes que se conectan al servidor para recuperar todos los recursos necesarios para funcionar, en este sentido, el cliente solo es una capa para representar los datos y se detonan acciones para modificar el estado del servidor, mientras que el servidor es el que hace todo el trabajo pesado.

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.



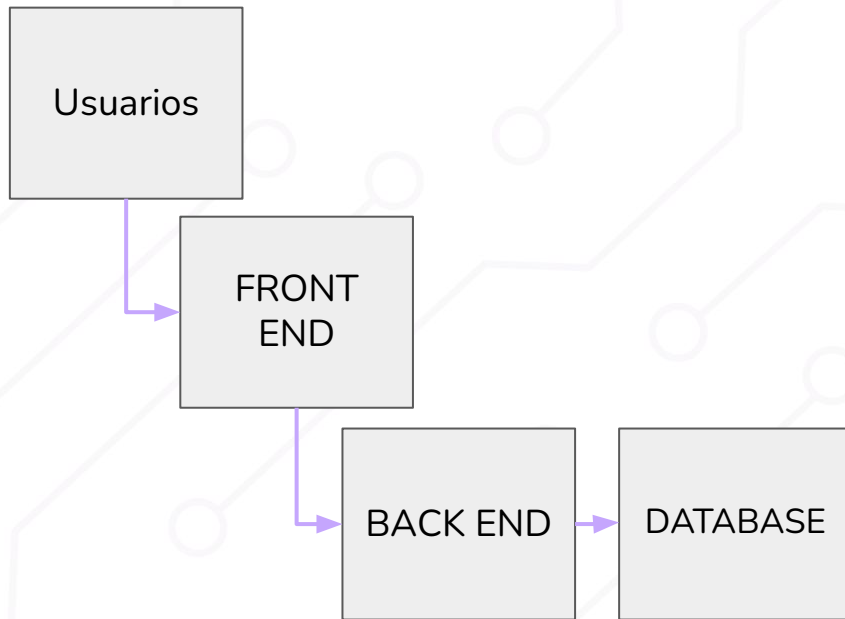
¿Qué contiene un stack tecnológico?

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 MEAN

- 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

