

Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación



Proyecto 2

Luis Pedro García 19344
Andres Paiz Caniz 191142

Tecnologías usadas:

- Vue.js
- MongoDB
- Express.js
- Axios
- etc.

¿Qué son, por qué la escogimos y las usaríamos nuevamente?

Andres:

vue

A primera instancia se preguntaría porque escogimos este framework para trabajar en lugar de otros, en este caso durante el desarrollo del proyecto no éramos muy experimentados con desarrollo web por lo cual decidimos escoger vue ya que, Vue.js es conocido por su curva de aprendizaje suave y su sintaxis intuitiva. Es fácil de entender y comenzar a trabajar con él, lo que lo convierte en una excelente opción tanto para desarrolladores principiantes como experimentados. Vue.js es extremadamente flexible y se puede utilizar para crear proyectos de cualquier tamaño, desde pequeñas aplicaciones interactivas hasta aplicaciones empresariales complejas. También se puede integrar fácilmente en proyectos existentes, lo que permite una transición gradual y sin problemas. Esto fue muy útil en nuestro caso ya que se debía de integrar porciones diferentes a un proyecto existente.

Algunas razones por la que se utilizó para el E-commerce:

Interfaz de usuario interactiva: Vue.js es conocido por su capacidad para crear interfaces de usuario interactivas y altamente responsivas. En un sitio de comercio electrónico, esto es crucial para brindar una experiencia de usuario fluida y atractiva. Con Vue.js, podemos crear componentes reutilizables para diferentes elementos de la interfaz, como el carrito de compras, filtros de productos, selección de tallas y colores, etc.

Estado de la aplicación gestionado: Vue.js utiliza su sistema de gestión de estado centralizado, Vuex, que facilita la administración de los datos de la aplicación en un

sitio de comercio electrónico. Se puede mantener un seguimiento del carrito de compras, el historial de pedidos, la información de usuario, entre otros datos relevantes, y mantenerlos sincronizados en toda la aplicación.

En general, muchas personas eligen Vue.js y lo consideran para futuros proyectos debido a su eficiencia de desarrollo, facilidad de aprendizaje, flexibilidad, escalabilidad, comunidad activa y rendimiento sólido. Sin embargo, la elección de la biblioteca o framework adecuado siempre depende de los requisitos y las necesidades específicas del proyecto. Dependiendo del proyecto personalmente considero que lo podríamos utilizar otra vez dada la facilidad que se tuvo al utilizarlo aunque hoy en día tengo mayor práctica con React por lo cual no estaría 100% seguro de utilizarlo de nuevo.

mongodb

MongoDB es una base de datos NoSQL que utiliza un modelo de documentos flexible en lugar de un esquema rígido. En un sitio de comercio electrónico, donde los datos pueden variar y evolucionar con el tiempo, esta flexibilidad es muy valiosa. Se puede almacenar fácilmente diferentes tipos de datos, como productos, usuarios, pedidos, comentarios, etc., sin tener que preocuparse por un esquema de base de datos fijo.

Para la base de datos del servidor se utilizó MongoDB, en términos coloquiales se utilizó mongo ya que es menos “lío” con el formato de los datos, al ser no relacional. Lo cual nos resultaba más fácil de utilizar y verdaderamente facilitaba la parte de backend en la que honestamente no estábamos muy acostumbrados a manejar. Igualmente evita la necesidad de conocer SQL para trabajar.

MongoDB ofrece una amplia gama de operaciones de búsqueda y consulta avanzadas. Esto es especialmente útil en un sitio de comercio electrónico, donde necesitas buscar productos, filtrar resultados, realizar búsquedas por palabras clave, aplicar clasificaciones y más. MongoDB admite índices, consultas geoespaciales y una sintaxis de consulta flexible que permite realizar consultas complejas de manera eficiente.

MongoDB cuenta con características integradas para garantizar la alta disponibilidad y la tolerancia a fallos en tu sitio de comercio electrónico. Puedes

configurar réplicas para tener copias redundantes de tus datos, lo que proporciona una mayor disponibilidad y garantiza que tu sitio siga funcionando incluso en caso de fallos en el hardware o los servidores. Lo cual es muy importante en nuestro caso, siendo un proyecto de “principiantes”.

Honestamente la razón #1 para utilizar mongo fue que hemos trabajado múltiples veces en múltiples cursos y nunca nos ha causado problemas. Es muy confiable por experiencia personal, lo he utilizado bastantes veces por mi lado y lo volvería a utilizar.

Node.js

Node.js es un entorno de ejecución de JavaScript basado en el motor de JavaScript V8 de Google Chrome. Permite ejecutar código JavaScript en el lado del servidor, lo que lo hace especialmente adecuado para el desarrollo de aplicaciones web y de red. A diferencia de la ejecución tradicional de JavaScript en el navegador, Node.js permite a los desarrolladores utilizar JavaScript para construir aplicaciones de servidor completas.

Node.js tiene una multitud de bibliotecas y módulos para utilizar, igualmente tiene una gran popularidad por lo cual siempre existen tutoriales o ayuda adicional fácil de encontrar y por supuesto una documentación extensiva.

La principal ventaja de usar Node es el hecho de que no es necesario aprender PHP, el cual yo personalmente odio apasionadamente (o Python o Java). Permite que toda la aplicación pueda estar escrita en JavaScript.

Obviamente el hecho de ya haber trabajado con Node fue una de las otras razones por la que lo escogimos, es fácil de usar, tenemos experiencia y resulta muy eficiente. Node.js se integra bien con bases de datos NoSQL, como MongoDB, por lo cual no fue una duda utilizar ambos.

Dadas todas estas razones volvería a utilizar Node.js en algún otro proyecto, sobre todo porque nunca he tenido problemas con él e incluso si los tengo encontrar el fix es bastante rapido y facil.

Luis Pedro:

Express.js:

Express es uno de los frameworks de backend más populares para Node.js. Es conocido por ser minimalista y flexible, y por que proporciona un conjunto sólido de funciones para aplicaciones web y móviles. Entre sus funcionalidades más importantes está que nos da una capa de abstracción sobre el servidor HTTP de Node.js, cosa que hace más fácil la creación de páginas.

Primero que nada, escogimos esta tecnología por su simplicidad. Como mencionado anteriormente, express es minimalista y no impone una estructura predeterminada, lo que la hace una tecnología versátil. Además, express ofrece mucha flexibilidad para crear diferentes tipos de páginas web. Con express se pueden construir aplicaciones de una sola página o hasta proyectos complejos con múltiples páginas.

Después, otra razón por la que escogimos express es por su popularidad. Es evidente que una tecnología popular tendrá más personas trabajando con ella, y por eso, habrá más soluciones fácilmente disponibles para problemas recurrentes. Con respecto a la popularidad, eso también significa que habrá una mayor cantidad de middleware, que nos permitirían agregar funcionalidades a nuestro proyecto.

Creemos que sí volveríamos a usar esta tecnología de nuevo ya que no nos dió muchos problemas y los pocos que tuvimos, los pudimos arreglar con la vasta documentación, foros y tutoriales que hay allá afuera.

El proceso de crear el backend fue muy fluido, fácil de entender, flexible, y más importante, funcional.

Axios:

Axios es un cliente HTTP basado en promesas para Javascripts que tiene la capacidad de hacer solicitudes HTTP desde el navegador y manejar la transformación de los datos de solicitud y respuesta. (Codebots, 2023)

Nosotros usamos axios para unir el front-end con el back-end mediante la realización de solicitudes HTTP al API del servidor. Decidimos usar esta tecnología

porque primero que nada tiene una sintaxis muy sencilla que nos facilitó la realización de solicitudes. Además, vimos que axios nos daba métodos convenientes para enviar datos en formato JSON, que era como teníamos nuestros productos.

Axios también es compatible con Node.js, que fue otra de las razones principales por las cuales decidimos usarlo.

Por último, decidimos usarlo porque ya que axios se basa en promesas, se facilitaba la gestión de las respuestas asíncronas, utilizando métodos como `async/await` y/o `then/catch`.

Definitivamente usaríamos axios de nuevo ya que fue una tecnología útil y fácil de usar para unir nuestro front-end con nuestro back-end. Además, su interfaz intuitiva y su integración con promesas hacen que el código sea fácil de leer, mantener y que sea reusable.

Nodemon:

Nodemon es una herramienta de desarrollo para Node.js que facilita la tarea de reiniciar automáticamente una aplicación cuando se detectan cambios en los archivos.

Nos dimos cuenta que esta sería una funcionalidad muy útil para no tener que estar reiniciando el server a cada rato, por lo que decidimos usar esta tecnología. Su reinicio automático revisa los archivos del proyecto en busca de cambios y automáticamente reinicia la aplicación cuando se detecta una modificación.

Por último, decidimos usar esta tecnología en vez de otras por su facilidad a la hora de configurarlo, que básicamente solo era instalarlo.

También, creemos que sí volveríamos a usar esta tecnología por que nos ahorró mucho tiempo y sería útil para cualquier otra aplicación que requiera backend.

Manual de Usuario:

El sitio es una tienda de zapatos, al iniciar la página se puede observar el catálogo de zapatos en la frontpage, cada producto incluye su rating, precio y un botón para agregarlo al carrito de compras. Al escoger un producto podemos ver sus detalles y luego agregarlo al carrito. Al oprimir en el carrito podemos ver los artículos que se encuentran dentro del carrito, y al mismo tiempo finalizar nuestra compra o remover artículos (el total a pagar se puede observar al final de la lista de artículos).