



Bootcamp full stack web

PROYECTO FINAL



Presentamos nuestro proyecto “El Último & Me Voy”, una aplicación web de “Blogging” enfocada en temas relacionados con la programación y, más concretamente, con la programación web.



¿De que está compuesta?

<backend>



<frontend>



<DDBB>



¿Por qué NODE?

Velocidad.

Node.js está construido sobre el motor de JavaScript V8 de Google Chrome, por eso su biblioteca es muy rápida en la ejecución de código.

Asíncrono.

Asíncrono y controlado por eventos. Un servidor basado en Node.js no espera que una API devuelva datos. El servidor pasa a la siguiente API después de llamarla, y un mecanismo de notificación de eventos ayuda al servidor a obtener una respuesta de la llamada a la API anterior.

Escalable.

Utiliza un modelo de un solo subproceso con bucle de eventos. El servidor responde sin bloqueos, esto hace que el servidor sea altamente escalable comparando con los servidores tradicionales como el Servidor HTTP de Apache.

Sin búfer.

Las aplicaciones de Node.js generan los datos en trozos (chunks), nunca los almacenan en búfer.



¿Por qué REACT?

Componentes.

Las aplicaciones se realizan con la composición de varios componentes. Estos componentes encapsulan un comportamiento, una vista y un estado.

Desarrollo Declarativo.

Nosotros contamos con un estado de la aplicación y sus componentes reaccionan ante el cambio de ese estado. Los componentes tienen una funcionalidad dada y cuando cambia una de sus propiedades ellos producen un cambio.

Flujo de datos uni-direccional.

En este modelo de funcionamiento, los componentes de orden superior propagan datos a los componentes de orden inferior.

DOM Virtual.

El resultado es que las operaciones permiten actualizaciones de hasta 60 frames por segundo, lo que producen aplicaciones muy fluidas, con movimientos suavizados.



¿Por qué REACT?

Isomorfismo

Las aplicaciones se realizan con la composición de varios componentes. Estos componentes encapsulan un comportamiento, una vista y un estado.

JSX

ReactJS no retorna HTML. El código embebido dentro de Javascript, parece HTML pero realmente es JSX. Son como funciones Javascript, pero expresadas mediante una sintaxis propia de React llamada JSX.

Estados .

React permite crear componentes de diversas maneras, pero hay una diferencia entre componentes con y sin estado

Ciclo de vida de los componentes

Métodos que se ejecutan cuando pasan cosas comunes con el componente, que nos permiten suscribir acciones cuando se produce una inicialización

¿Por qué MONGO?

MongoDB almacena datos en documentos flexibles similares a JSON, por lo que los campos pueden variar entre documentos y la estructura de datos puede cambiarse con el tiempo

¿Por qué Atlas?

Esta decisión la tomamos al principio, y fué para poder colaborar todos juntos con los mismos datos, nos permitía trabajar en la nube sin necesidad de descargarnos contenido, y poder hacer modificaciones o consultas.



Comenzamos....

Todo el proyecto se ha elaborado en un mismo repositorio, cada miembro del equipo se creó una rama, desarrollaba una parte de la aplicación y cuando estaba terminado lo subía a la rama master, los demás actualizábamos nuestras correspondientes ramas con la información actualizada. Esto ha propiciado un entorno de trabajo dinámico y ágil.



....el backend

SE DECIDIÓ COMENZAR EXCLUSIVAMENTE POR EL BACKEND, ELABORANDO UN API REST

<GET>

Usuarios:

- Mostrar usuario por su Nickname
- ID de Usuario.

Artículos:

- Mostrar todos los artículos
- Mostrar articulo por ID
- Orden por fecha de publicación

<POST>

Usuarios:

- Registrar un usuario
- Iniciar sesión con usuario.
- Restablecer contraseña.
- Seguir o dejar de seguir a un usuario.
- Añadir o eliminar un artículo de favoritos

Artículos:

- Crear un artículo
- Crear un nuevo artículo en respuesta a otro artículo.
- Realizar búsqueda de artículos.

Comentarios:

- Añadir un comentario a un artículo
- Responder a un comentario de un artículo



Necesitará incluir el valor Bearer <token> del usuario

...el backend

<PATCH>

Usuarios:

- Actualizar un usuario

Artículos:

- Actualizar un artículo

<DELETE>

Usuarios:

- Eliminar un usuario (y los anuncios que hubiese creado):

Artículos:

- Eliminar un artículo:

Comentarios:

- Eliminar un comentario (*tanto el creador del comentario como el creador del artículo*)

EL backend también dispone de:

Swagger
Test



Necesitará incluir el valor Bearer
<token> del usuario **CREADOR**



...el frontend

LA APLICACIÓN DISPONE DE DIFERENTES PÁGINAS O RUTAS PÚBLICAS Y PRIVADAS

<COMPONENTES COMUNES>

- Menú principal
- Cambio de idioma (inglés y español)
- Búsqueda
- Crear artículo (accesible sólo estando logueado)
- Home
- Usuario (si no se está logueado mostrará un popup para: iniciar sesión, registrarse o recuperar contraseña. Cuando se ha iniciado sesión, muestra las opciones "Perfil" y "Cerrar sesión")
- Pié de página (muestra los miembros que han realizado este proyecto y links a sus redes)

<PÚBLICAS>

- Home
- Búsqueda de artículos
- Detalle de artículos
- Perfiles de usuarios
- Recuperar contraseña

<PRIVADAS>

- Crear artículo
- Mi cuenta

Ee envían **emails**:

- Recuperar contraseña,
- Un usuario realiza un comentario en tu artículo
- Un usuario que sigues, crea un artículo.



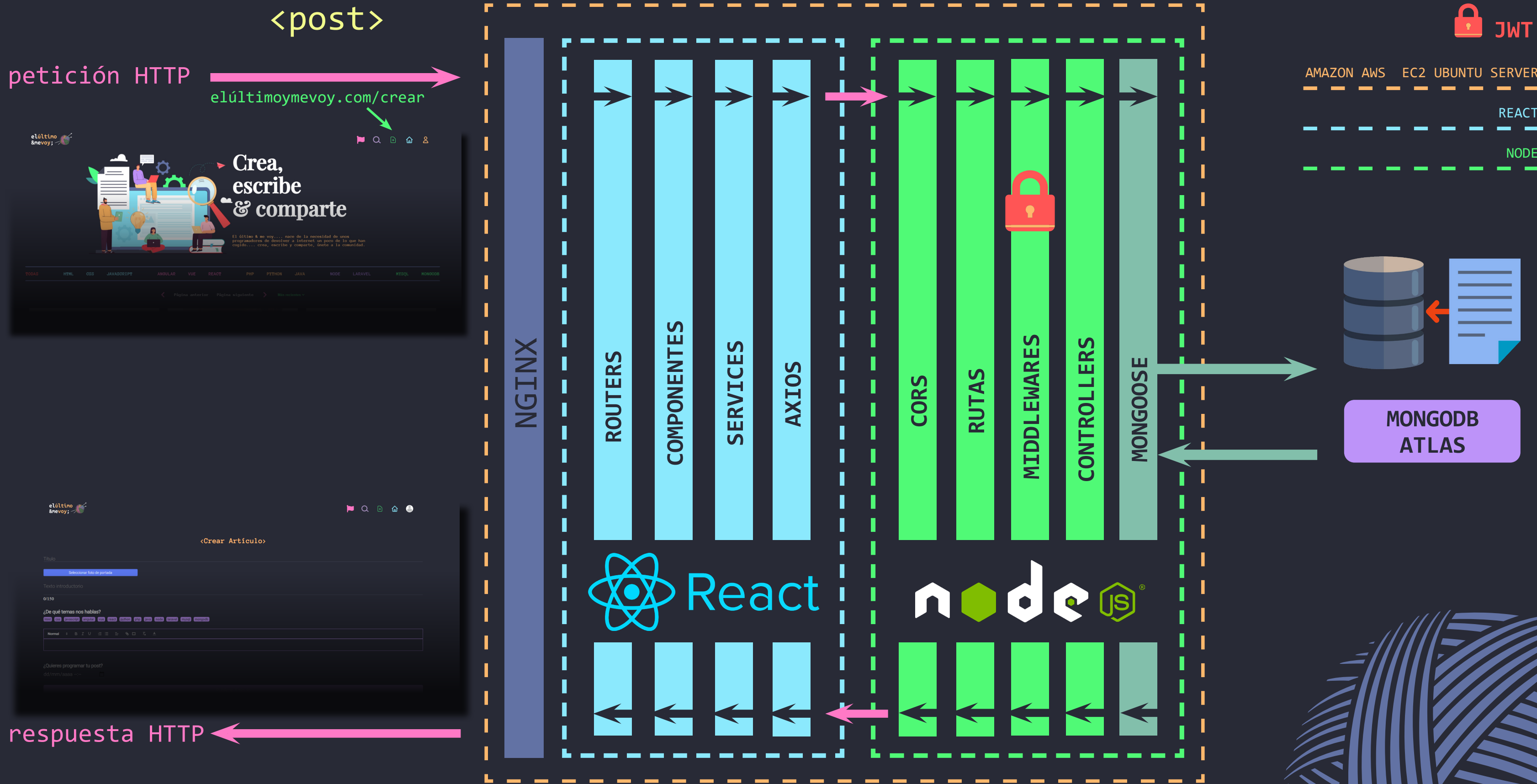
¿Qué te ha aportado el desarrollar este proyecto?

- Afianzar los conocimientos que se han impartido durante el bootcamp
- Aprender a trabajar en equipo
- Ver diferentes puntos de vista
- Aprender a adaptarte a una metodología de trabajo diferente a la tuya

A nivel personal ha sido muy enriquecedor poder trabajar con personas tan dedicadas y profesionales

PD: También nos hemos reído mucho XD.....





**¿Qué es lo que no volverías a hacer
de la misma manera?**



**¿Qué cosas seguirías haciendo
en el futuro para mejorar el proyecto?**

