

Nicolás Artemio Dume

FRONT END DEVELOPER

Desarrollador Front End autodidacta, apasionado por resolver problemas y ayudar a las personas. Mi objetivo a futuro es convertirme en un destacado Tech Lead, contribuyendo al crecimiento y éxito de equipos de desarrollo.

EXPERIENCIA

Portafolio

Proyecto personal creado desde cero :

- Utilice Vite para acelerar la creación y la optimización del proyecto.
- En este proyecto, utilicé una arquitectura dividida en tres carpetas principales: "components", que contiene componentes reutilizables; "pages", donde se encuentran las páginas principales de la aplicación; y "routes", que alberga la configuración de las rutas.
- React y Typescript para que el código sea más sólido y mantenible.
- React Router DOM para la gestión de las rutas.
- Para el diseño utilice Tailwind CSS para estilos rápidos y AOS para añadir animaciones.

Backend - Api

- Desarrollé una API eficiente y versátil diseñada para facilitar la comunicación fluida entre el backend y el frontend.
Esta API utiliza Node.js, Express.js, Nodemon y TypeScript para ofrecer un backend ágil y fácil de mantener.
- Inicié el proyecto instalando las dependencias esenciales y posteriormente diseñé una arquitectura organizada, dividida en tres componentes clave:
 - Middlewares: Configuré middlewares para gestionar la validación de datos, para manejar las configuraciones de CORS, garantizando la seguridad y la permisividad adecuadas en las solicitudes desde el frontend.
 - Routes: Cree y configure mis endpoints utilizando el método GET.
 - Services: Desarrollé lógica de negocio y manipulación de datos, utilizando tipado en TypeScript para mejorar la claridad del código.

Games - App

- Desarrollé una aplicación que incluye tres juegos: Memotest, Adivina el Personaje y WPM. Al iniciar el proyecto, instalé las dependencias y configuré Webpack y Babel desde cero. Utilicé tecnologías como React y React Router DOM. Además, organicé el proyecto en una estructura modular, dividiendo cada juego en componentes separados con su lógica correspondiente dentro de la carpeta 'components'.
- También establecí carpetas adicionales, como 'models' reservada para información adicional, 'pages' cada juego tiene su propia página facilitando la navegación, 'routes' contiene archivos que definen las rutas de la aplicación y por último 'utils/hooks' En esta carpeta, implementé custom hooks personalizados que mejoran la eficiencia y promueven la reutilización del código.

Buenos Aires, Olavarría
dumenicolas45@gmail.com

[Linkedin](#)

[Github](#)

[Web](#)

[YouTube](#)

HERRAMIENTAS

Html
Css
Javascript
React
Typescript
Node.js
Express
Git
Linux

CURSOS

- Krowdy academy
(Despliegue, monitoreo, Docker, Kubernetes, Mongo db, React).
- Plan 111 mil (Java, Mysql).

IDIOMAS

Español: Nativo
Ingles: B1

Habilidades Blandas

Facilidad de comunicación.
Responsabilidad.
Solución de problemas.
Creatividad.
Multitarea.
Aprendizaje activo.

Todo - App

- Para este proyecto utilice Vite para una configuración rápida y eficiente.
- React y Typescript para que el código sea más sólido y mantenible, integre ESLint para garantizar la consistencia y calidad del código.
- Además, he incorporado todomvc-app-css para aprovechar estilos predeterminados y garantizar una apariencia coherente y atractiva para la aplicación.
- La estructura que has mostrado es bastante clara y sigue las mejores prácticas para organizar componentes en archivos individuales.
- En cada componente de la aplicación, la información crucial se transmite a través de props.
- La app te deja hacer de todo con tus tareas. Puedes crear nuevas, tacharlas cuando las completes y ver solo las que están pendientes o ya hiciste. También tienes la opción de borrar una tarea en particular o todas de una vez.

Blog Cod demo

- En este proyecto, he simulado un blog de CoD. Para la creación, he utilizado Vite como entorno de desarrollo, React para la construcción de la interfaz de usuario, React Router DOM para la gestión de rutas, y Pinterest como referencia para recrear el diseño original. Además, he empleado Tailwind CSS para facilitar el diseño y he implementado un menú responsivo para mejorar la experiencia del usuario.
- La estructura del proyecto se divide en las siguientes carpetas:
- Components: Aquí, he organizado los componentes en secciones clave, como el encabezado (header), el pie de página (footer) y la barra de navegación (navbar).
- Pages: Esta carpeta está dividida en subcarpetas que representan las diferentes secciones del blog, tales como 'about', 'games', y 'home'.
- Routes: En esta carpeta, se encuentra el archivo 'approute', que gestiona y define las rutas de la aplicación.

Hangman Game

- He desarrollado el clásico juego del ahorcado utilizando Vite, React y TypeScript. La estructura del juego se ha organizado en componentes, incluyendo 'hangmandrawing' para representar las partes del cuerpo mediante estilos CSS, 'hangmanword' para la lógica principal de mostrar las letras de la palabra a adivinar, y 'keyboard', que recibe las letras como propiedades y renderiza botones correspondientes a cada letra del teclado. Este enfoque modular facilita el mantenimiento y la comprensión del código del juego.