

Documentación de tecnologías usadas

Para este test de tareas, he realizado un Stack MERN de JavaScript usando MongoDB, Express, React y NodeJs, como tecnologías bases para el desarrollo de la misma.

MongoDB → <https://www.mongodb.com/>

Express → <https://expressjs.com/es/4x/api.html>

React → <https://reactjs.org/>

Node → <https://nodejs.org/en/>

Ademas de estas tecnologías me he apoyado en paquetes externos para ayudar en ciertas funcionalidades.

Para el **BackEnd**:

Dependencias:

Cors → <https://github.com/expressjs/cors>

Express-Validator → <https://express-validator.github.io/docs/>

Moment-Timezone → <https://momentjs.com/timezone/>

Mongoose → <https://mongoosejs.com/>

Dependencias de desarrollo:

Morgan → <https://github.com/expressjs/morgan>

Nodemon → <https://nodemon.io/>

Para el **FrontEnd**:

Create React App → <https://github.com/facebook/create-react-app>

Toast Notifications for ReactJs → <https://www.npmjs.com/package/react-notify-toast>

React Router DOM → <https://www.npmjs.com/package/react-router-dom>

Resumen

He usado **MongoDB** como gestor de bases de datos, **Express** para crear y manejar las API, **React** como framework FrontEnd, **NodeJs** como plataforma para JavaScript del lado del servidor, **Cors** para evitar problemas relacionados a las peticiones, **Express-Validator** para validar los campos en el BackEnd, **Moment-Timezone** para las fechas formateadas a una localización específica, Mongoose gestionar el manejo del gestor de base de datos, **Morgan** para controlar las peticiones que recibe el BackEnd, **Nodemon** para tener el servidor de NodeJs en caliente (que se refresque automáticamente), **Create React App** para crear la aplicación web con React, **Toast Notifications for ReactJs** para enviar mensajes al usuario de las respuestas del BackEnd por la petición realizada y **React Router DOM** para la creación del enrutado en React.