

Simple app with MongoDB backend and React FrontEnd

Hugo Pelayo and Guillermo Quintanar

Backend

node JS

express



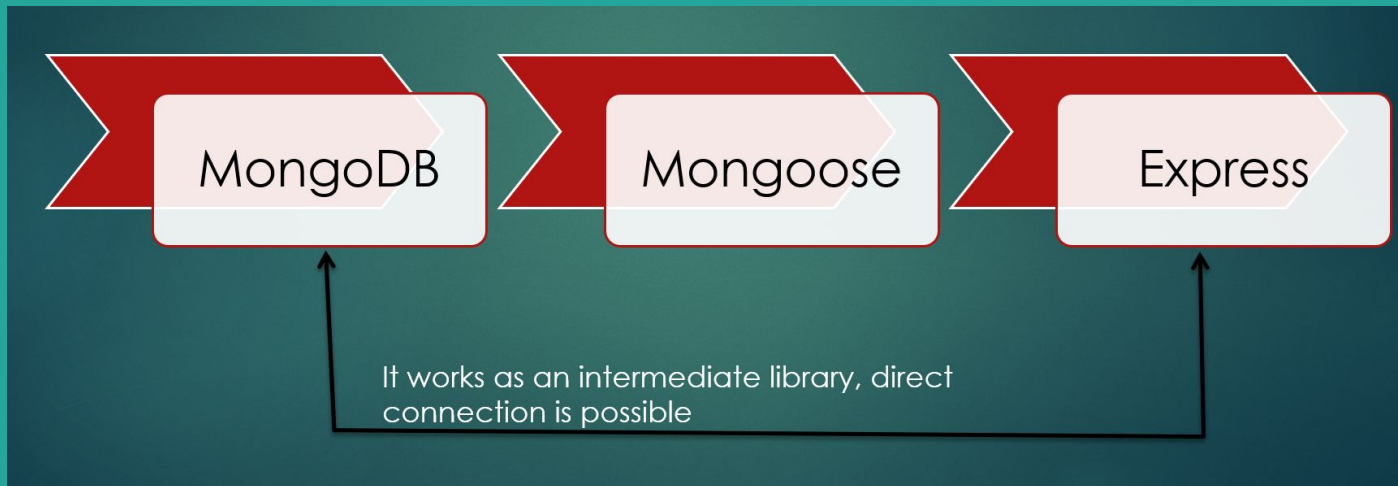
mongoDB



mongoDB



Mongoose



Express

The logo for Express.js, featuring the letters "JS" in a bold, black, sans-serif font, centered within a bright yellow square.

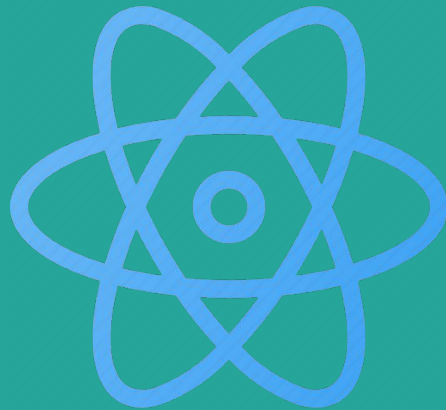
Express

```
const aplicacion = new express();  
  
aplicacion.use(cors());  
  
aplicacion.use(morgan("dev"));  
  
aplicacion.use(express.json());  
  
aplicacion.use(cookieParser());  
  
aplicacion.use("/api",router);  
  
aplicacion.use("/api",ruta);  
  
aplicacion.use("/api",rutaAlumno);  
  
export default aplicacion;
```

Cookies Bcrypt



Frontend Tech



Specify the sites for our routes

```
import TaskPage from "../pages/TaskPage.jsx";
import TaskFormPage from "../pages/TaskFormPage.jsx";
import ProfilePage from "../pages/ProfilePage.jsx";

import { AuthProvider } from "../context/AuthContext.jsx";

export default function App() {
  return (
    <AuthProvider>
      <BrowserRouter>
        <Routes>
          <Route path="/" element={ <h1 className="text-4xl font-bold">Home Page</h1> } />
          <Route path="/login" element={ <LoginPage/> } />
          <Route path="/register" element={ <RegisterPage/> } />

          <Route path="/tasks" element={ <TaskPage/> } />
          <Route path="/add-task" element={ <TaskFormPage/> } />
          <Route path="/tasks/:id" element={ <TaskFormPage/> } />
          <Route path="/miPerfil" element={ <ProfilePage/> } />
        </Routes>
      </BrowserRouter>
    </AuthProvider>
  );
}
```



You, 4 days ago (November 26th, 2023 9:41 PM)

acabada practica 3 psps

4031923 < > | Connect to GitHub...

- <Route path="/login" element={ <LoginPage/> } />

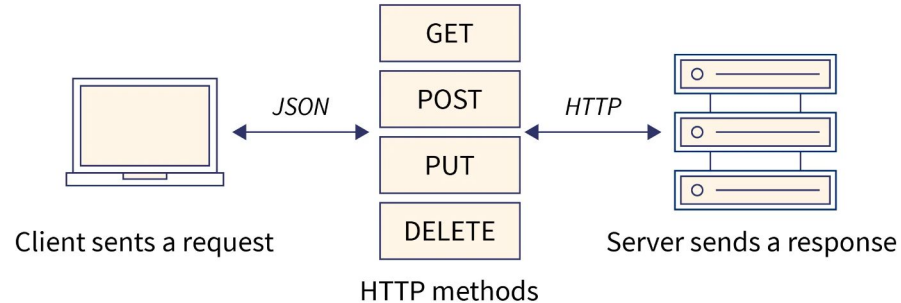
+ <Route path="/" element={ <h1 className="text-4xl font-bold">Home Page</h1> } />

Changes 70b7f3e ↔ 4031923 |

You, 4 days ago

Now we need to send the data from our frontend to our backend.

We use axios for this, but there's a built-in alternative in the browser (fetch).



To manage our frontend context we use providers, which are functions that pass data to the different components of our app.

```
export const AuthContext = createContext();

export const useAuth = () => {
  const context = useContext(AuthContext);

  if (!context) {
    throw Error("useAuth must be used within an AuthProvider");
  }

  return context;
};

export const AuthProvider = ({ children }) => {
  const [user, setUser] = useState<User>();
  const [isAuthenticated, setIsAuthenticated] = useState<boolean>();
  const [errors, setErrors] = useState<string>();

  const signup = async (user) => {
    try {
      const res = RegisterRequest(user);
    } catch (error) {
      setErrors(error.message);
    }
  };

  return (
    <AuthContext.Provider value={{ user, setUser, isAuthenticated, setIsAuthenticated, errors, setErrors, signup }}>
      {children}
    </AuthContext.Provider>
  );
};
```

(alias) useState<T>(initialState: (() => T) | T): [T, React.Dispatch<React.SetStateAction<T>>] (+1 overload)
import useState
Returns a stateful value, and a function to update it.
@version — 16.8.0
@see — <https://react.dev/reference/react/useState>

Finally we just need to secure our routes

```
    <Route element={<ProtectedRoute/>}>
      <Route path="/tasks" element={ <TaskPage/> } />
      <Route path="/add-task" element={ <TaskFormPage/> } />
      <Route path="/tasks/:id" element={ <TaskFormPage/> } />
      <Route path="/miPerfil" element={ <ProfilePage/> } />
    </Route>
  </BrowserRouter>
```



Great Experience!



PRESENTACIÓN
PATROCINADA POR:



**Loterías y Apuestas
del Estado**