

Sprint 5 - IA generativa

1. Explicación de la elección del modelo de IA

En un primer momento pensé en usar ChatGPT por ser el más conocido, pero tras las recomendaciones de mis compañeros, me pasé a Claude.

Aun así, terminé usando ambas IAs para diferentes fines.

- ChatGPT me parece que te puede resolver dudas concretas como las de enfrentarte a una librería o proyecto nuevo, donde te puede hacer un resumen y darte los primeros pasos para que te encamine en el buen camino.
Pero para cosas más grandes no parece ser muy consistente, ya que debes ir pidiéndole a cada petición que recuerde o tenga en cuenta algo ya mencionado anteriormente.
- Claude por otro lado desde el primer momento me pareció más acertado para proyectos de más allá de 2 archivos, ya que se le nota más consistencia de respuesta a respuesta y es capaz de generar archivos más grandes y centrados.

2. Registro de las interacciones con la IA

Comencé preguntándole a ChatGPT que me describiera cómo era el proceso de usar la librería de React para empezar a tener una idea de a qué me enfrentaba.

He desarrollado el backend de una aplicación web en PHP y Laravel con arquitectura API REST, y tengo que crear el frontend de este proyecto usando la librería de React. Me podrías explicar cómo debería usar esta librería para poder conseguirlo? Nunca he usado React y querría saber cómo es trabajar con esa librería a la hora de generar el frontend de una API.

Buenísima combinación: **Laravel como API + React como frontend** es un stack muy usado y potente. Como vienes de backend, lo más importante no es aprender "React entero", sino entender **cómo piensa React y cómo se comunica con tu API REST**.

Voy a explicártelo en 4 partes:

1. Qué es React y cómo se trabaja con él
2. Estructura típica de un frontend con React
3. Cómo conectar React con tu API Laravel
4. Flujo real de trabajo (mentalidad backend → frontend)

1. Qué es React realmente

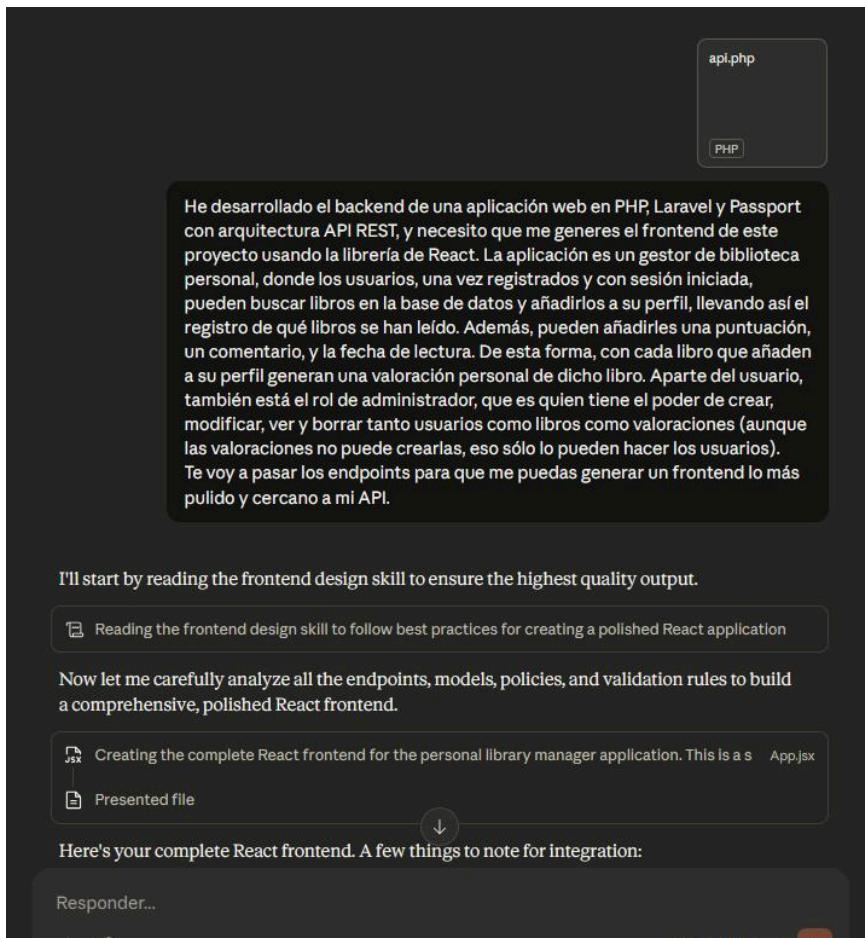
React no es un framework completo, es una librería para construir interfaces por componentes.

En lugar de hacer HTML + JS suelto, trabajas así:

- Todo son **componentes**
- Cada componente es una función
- Cada componente puede tener:
 - estado (`state`)
 - efectos (`useEffect`)
 - props (datos que recibe)

De esta forma empecé a aprender la estructura de un proyecto en React, que usa componentes como los de Blade, los pages a modo de vistas...

Pasé entonces a Claude y le describí el proyecto, sus funcionalidades e ideas generales y le pasé los endpoints para afinar mejor en los resultados.



Me devolvió un único archivo App.jsx con todo el proyecto de frontend, más de 1600 líneas de código. Una animalada.

```
1620     <div className="user-avatar">{user.name.charAt(0).toUpperCase()}</div>
1621     <div className="user-info">
1622       <div className="name">{user.name}</div>
1623       <div className="role">{user.is_admin ? "Administrador" : "Us</div>
1624       </div>
1625     </div>
1626     <button className="logout-btn" onClick={logout}>
1627       <Icon name="logout" style={{ width: 14, height: 14 }} /> Cerra</button>
1628   </div>
1629 </aside>
1630
1631
1632 <main className={`main-content ${sidebarOpen ? "" : "expanded"}`}>
1633   <div className="top-bar">
1634     <div style={{ display: "flex", alignItems: "center" }}>
1635       <button className="hamburger" onClick={() => setSidebarOpen(!s</div>
1636       <h2>{routeTitles[route] || "Biblioteca"}</h2>
1637     </div>
1638   </div>
1639   <div className="page-body">
1640     {renderPage()}
1641   </div>
1642 </main>
1643 </div>
1644 );
1645 }
1646
1647 // -----
1648 // DEFAULT EXPORT
1649 // -----
1650 export default function MyLibraryApp() {
1651   return (
1652     <AuthProvider>
1653       <Root />
1654     </AuthProvider>
1655   );
1656 }
1657
```

Eché un vistazo al código y era capaz de ver los diferentes métodos que se usarían como componentes, el diseño de los iconos, partes de HTML dentro de código JavaScript... E intenté separarlo en distintos archivos independientes, aprovechando para crear la estructura de carpetas que me sugirió antes ChatGPT.

Tras unos cuantos intentos fallidos, al no estar acostumbrado todavía a este tipo de proyectos, le pedí a Claude que me separara el código en sus respectivos archivos. Ajusté las rutas, y corregimos un par de problemas menores:

1. Al iniciar sesión, los usuarios no administradores eran redirigidos a la pantalla de gestión de usuarios, donde podían hacer las funciones de administrador. La solución fue sencilla, necesitando cambiar un par de líneas en el archivo Root.

The screenshot shows a dark-themed code editor interface. At the top, there is a message box with yellow text on a black background: "Hay un problema en alguna parte de este código, porque cuando inicia sesión un usuario no administrador, lo primero que ve al entrar es la ventana de usuarios, donde puede ver, crear, editar y borrar usuarios. ¿Dónde puede estar el error?" Below this, the main code editor area displays the following text:

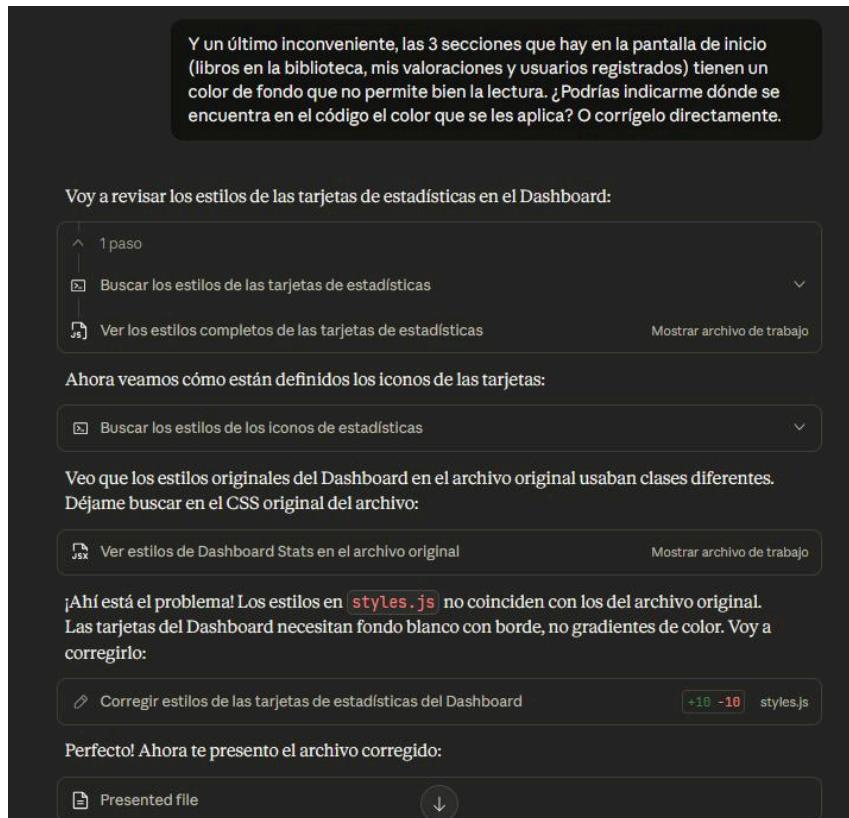
```
El problema está en el archivo Root.jsx. Cuando un usuario no administrador inicia sesión, la ruta por defecto se establece como "dashboard", pero el problema está en la función renderPage() que no está validando correctamente los permisos de administrador.
```

Below the code, there is a note: "Déjame verificar y corregir el archivo Root.jsx:"

At the bottom of the editor, there are several buttons and status indicators:

- A "Ver el archivo Root.jsx para identificar el problema" button with a file icon.
- A "Mostrar archivo de trabajo" button.
- A note: "Ahora veo el problema. La función renderPage() en las líneas 54-55 renderiza las páginas de usuarios y valoraciones sin verificar si el usuario es administrador. Necesito agregar validación de permisos. Aquí está la corrección:"
- A "Agregar validación de permisos de administrador en renderPage" button with a checkmark icon.
- A "Root.jsx" file status indicator showing "+13 -13".
- A note: "Perfecto, ahora te presento el archivo corregido:"
- A "Presented file" button with a document icon.
- A note: "¡Listo! El problema estaba en la función renderPage() del archivo Root.jsx."
- A note: "El error: Las rutas "users" y "allreviews" renderizaban directamente las páginas de administrador sin verificar los permisos del usuario."
- A note: "La solución: Ahora las líneas 54-55 verifican si el usuario es administrador antes de mostrar esas páginas:"

2. Y un problema gráfico que no pude ver de dónde venía, pese a buscar la clase o los ids de los elementos problemáticos en el navegador y en los archivos de estilo. En la ventana de inicio se muestran 3 paneles con estadísticas varias, pero su color de fondo dificultaba la lectura. Terminé preguntándole a Claude dónde estaba el problema.



3. Análisis del código generado por la IA

Pese a generar en primera instancia un archivo abrumador, se podía intuir los diferentes bloques de cada elemento. Y aunque no estoy del todo familiarizado con Javascript, sí que podía ver a grandes rasgos qué hacia cada bloque. También podía intuir que el código estaba bastante sintetizado/factorizado, muy compacto todo.

Aun así, tuve que ajustar algunas pocas líneas de código, como por ejemplo la carga de los estilos. Mientras iba ajustando las rutas de los archivos ya divididos, vi que el main.jsx no cargaba correctamente el archivo de index.css, y al corregirlo, se podían ver correctamente los iconos en el frontend.

4. Descripción del proceso de conexión entre el frontend y el backend

La verdad es que el frontend funcionaba desde la primera versión, con algunos ligeros errores. Por lo pude entender y ver, cuando el usuario hace clic en algún botón, React ejecuta su función, envía una petición HTTP, la API procesa la lógica y consulta la base de datos, devolviendo un JSON que React recibe y guarda el estado, y finalmente React actualiza la vista.

5. Reflexión sobre el proceso de aprendizaje

Mi conclusión tras usar la IA para estos temas es que debes ser muy preciso con lo que quieres que te devuelva. Y, aun con esas, hay que revisar, probar y ajustar para que todo esté bien.

Es similar a la programación, en la que tienes que tener claro qué lequieres comunicar a un ordenador y cómo lo haces para conseguir tus objetivos.

Y eso es algo que se va adquiriendo con la práctica, al adquirir experiencia e ir desarrollando un instinto que te guíe y diga qué está bien y qué está mal.