1. **Tecnologías y Frameworks** **utilizados.**

**Backend:**

El backend está desarrollado con Node.js y el framework Express.js, lo que permite una gestión eficiente de rutas, peticiones y middleware. También se emplean las siguientes librerías:

* Axios: Para realizar solicitudes HTTP a servicios externos.
* Bcrypt.js: Para el cifrado de contraseñas y mejorar la seguridad.
* Cloudinary: Para el almacenamiento y gestión de imágenes en la nube.
* Cookie-parser: Para manejar cookies en las peticiones HTTP.
* CORS: Para permitir peticiones desde distintos orígenes.
* Dotenv: Para la gestión de variables de entorno y mejorar la seguridad del proyecto.
* Jsonwebtoken: Para la autenticación de usuarios mediante tokens JWT.
* Multer: Para la carga y gestión de archivos.
* Nodemailer: Para el envío de correos electrónicos en el proceso de recuperación de contraseñas.
* PG (PostgreSQL): Para la conexión y gestión de la base de datos.
* UUID: Para la generación de identificadores únicos en la aplicación.

**Frontend:**

El frontend está desarrollado con React.js y utiliza Vite como herramienta de desarrollo, lo que permite una carga rápida y optimización del código. Además, se han utilizado las siguientes librerías:

* @dotlottie/player-component: Para animaciones en formato Lottie.
* Axios: Para la comunicación con el backend mediante solicitudes HTTP.
* Chart.js & react-chartjs-2: Para la visualización de gráficos en la interfaz.
* JsBarcode: Para la generación de códigos de barras.
* JSPDF & JSPDF-AutoTable: Para la generación de documentos PDF.
* Lucide-react: Para la integración de iconos en la interfaz.
* PDFMake: Para la creación y personalización avanzada de documentos PDF.
* UUID: Para la generación de identificadores únicos.
* XLSX: Para la manipulación y generación de archivos Excel