**Descripción Completa del Proyecto**

**Nombre del Proyecto**

**Sistema de Gestión de Cédulas**

**Propósito**

El proyecto es una aplicación interna diseñada para el SAIME (Servicio Administrativo de Identificación, Migración y Extranjería) de Venezuela. Su objetivo principal es gestionar documentos de identificación, organizar registros y administrar usuarios de manera eficiente.

**Tecnologías Utilizadas**

**Lenguaje de Programación**

* **TypeScript**: Se utiliza TypeScript como lenguaje principal, lo que permite un desarrollo más robusto gracias a su tipado estático. Esto ayuda a prevenir errores comunes y facilita el mantenimiento del código.

**Framework**

* **Next.js**: Es un framework basado en React que permite la creación de aplicaciones web modernas con renderizado híbrido (estático y dinámico). Next.js también proporciona un sistema de enrutamiento basado en archivos y soporte para API integradas.

**Frontend**

* **React**: La interfaz de usuario está construida con React, lo que permite una experiencia interactiva y dinámica para los usuarios.
* **Tailwind CSS**: Se utiliza Tailwind CSS para el diseño y la estilización de la aplicación. Este framework de CSS basado en utilidades permite crear interfaces modernas y responsivas de manera eficiente.

**Backend**

* **Next.js API Routes**: Las rutas API integradas en Next.js se utilizan para manejar la lógica del backend, como la autenticación, la gestión de usuarios y la interacción con la base de datos.

**Base de Datos**

* **Firebase Firestore**: Se utiliza Firestore como base de datos NoSQL para almacenar y gestionar los datos de los documentos, usuarios y configuraciones del sistema. Firestore es una solución escalable y en tiempo real que facilita la sincronización de datos entre el cliente y el servidor.

**Autenticación**

* **NextAuth.js**: Se utiliza NextAuth.js para manejar la autenticación de usuarios. Este sistema permite la autenticación basada en credenciales y la gestión de sesiones de usuario.

**Dependencias Principales**

1. **next**: Framework principal para el desarrollo de la aplicación.
2. **react**: Biblioteca para construir la interfaz de usuario.
3. **react-dom**: Biblioteca para renderizar componentes React en el DOM.
4. **firebase**: SDK para interactuar con Firebase Firestore.
5. **next-auth**: Manejo de autenticación en Next.js.
6. **bcryptjs**: Para la encriptación y comparación de contraseñas.
7. **tailwindcss**: Framework CSS para el diseño de la interfaz.
8. **autoprefixer**: Herramienta para agregar prefijos CSS automáticamente.
9. **postcss**: Procesador CSS utilizado junto con Tailwind CSS.

**Estructura del Proyecto**

├── app/

│   ├── api/               # Endpoints de la API

│   │   ├── auth/          # Autenticación con NextAuth.js

│   │   ├── documentos/    # Gestión de documentos

│   │   ├── organizacion/  # Organización de registros

│   │   └── usuarios/      # Gestión de usuarios

│   ├── administracion/    # Páginas de administración

│   ├── organizacion/      # Página de organización

│   ├── registro/          # Página de registro de documentos

│   └── layout.tsx         # Layout principal de la aplicación

├── public/                # Archivos estáticos (imágenes, íconos, etc.)

│   └── saime-seeklogo.png # Logo del proyecto

├── styles/                # Archivos CSS globales

├── firebaseConfig.ts      # Configuración de Firebase

├── next.config.mjs        # Configuración de Next.js

├── tailwind.config.js     # Configuración de Tailwind CSS

├── postcss.config.js      # Configuración de PostCSS

├── package.json           # Dependencias y scripts del proyecto

└── README.md              # Documentación del proyecto

**Características del Proyecto**

1. **Gestión de Documentos**
   * Registro de documentos con información como nombre, cédula, tipo de trámite, fecha de emisión, estado, y ubicación.
   * Búsqueda de documentos por nombre o cédula.
   * Organización de documentos en carpetas y archivadores.
2. **Gestión de Usuarios**
   * Creación, edición y eliminación de usuarios.
   * Asignación de roles (Administrador, Usuario).
   * Autenticación segura con contraseñas encriptadas.
3. **Interfaz Intuitiva**
   * Diseño responsivo y moderno utilizando Tailwind CSS.
   * Navegación sencilla con un menú lateral que incluye enlaces a las principales secciones del sistema.
4. **Seguridad**
   * Contraseñas encriptadas con bcryptjs.
   * Manejo de sesiones de usuario con NextAuth.js.
   * Variables de entorno para proteger claves y configuraciones sensibles.
5. **Despliegue**
   * Desplegado en **Netlify**, una plataforma que permite el hosting de aplicaciones modernas con soporte para Next.js.
   * Configuración para desactivar la optimización de imágenes en Netlify (unoptimized: true en next.config.mjs).

**Flujo de Trabajo**

1. **Inicio de Sesión**
   * Los usuarios inician sesión con su correo y contraseña.
   * El sistema valida las credenciales y crea una sesión segura.
2. **Gestión de Documentos**
   * Los usuarios pueden registrar nuevos documentos, buscar documentos existentes y organizarlos en carpetas.
3. **Administración**
   * Los administradores pueden gestionar usuarios y configurar parámetros del sistema.
4. **Organización**
   * Los documentos se organizan en carpetas y archivadores para facilitar su búsqueda y acceso.