

Proyecto DWEC – Aplicación Meteorológica con React y AEMET

Descripción del proyecto

En este proyecto he desarrollado una aplicación web en React que consulta información meteorológica de la API oficial de la AEMET, mediante un backend en Node.js con Express que actúa como puente entre el frontend y la API oficial.

La aplicación permite:

- Consultar la predicción diaria y por horas para el código de municipio dado.
- Mostrar información meteorológica relevante: estado del cielo, temperatura máxima y mínima, probabilidad de precipitación, viento y fecha del pronóstico.
- Gestionar estados de carga, error y ausencia de datos.

Se ha diseñado con dispositivos móviles en mente utilizando un diseño responsive y también implementando un botón para alternar entre el tema de modo claro y oscuro.

Instrucciones de instalación y ejecución

Requisitos previos:

- Node.js instalado
- npm o yarn

Clonar el repositorio:

```
git clone https://github.com/4149822/aemet
cd aemet
```

Instalación y ejecución del backend:

Ir a la carpeta del backend:

```
cd backend
```

Instalar dependencias:

```
npm install
```

Crear un archivo .env copiando el de ejemplo:

```
cp .env.example .env
```

Poner tu API Key de AEMET en el .env creado:

```
AEMET_API_KEY=TU_API_KEY_AQUI
```

Iniciar el servidor:

```
npm start
```

Instalación y ejecución del frontend:

Ir a la carpeta del frontend:

```
cd frontend
```

Instalar dependencias:

```
npm install
```

Iniciar la aplicación:

```
npm run dev
```

Navegar a la dirección dada por la consola.

Decisiones técnicas tomadas

- Como pide el enunciado, se utiliza un backend intermedio para proteger la clave de la API y procesar los datos antes de enviarlos al frontend.
- No se han utilizado librerías externas de diseño ya que es un proyecto sencillo.
- Se han simplificado algunos resultados meteorológicos al mostrarlos para no saturar la interfaz, siempre priorizando claridad y compresión sobre cantidad de datos mostrados.
- He utilizado emojis para representar los estados del cielo y he añadido algunos iconos a la interfaz de [IconPark Outline](#) para mejorar la consistencia visual.
- Se han separado la lógica en componentes funcionales y hooks utilizando `useState` y `useEffect`.
- Se manejan errores en el backend de manera que únicamente se manden al frontend los relevantes para este.

Conclusiones tras la realización del proyecto

La aplicación cumple con los requisitos mínimos del enunciado y se han implementado algunas mejoras adicionales sugeridas.

Aunque el proyecto podría ampliarse y mejorarse mucho, el tiempo disponible para esta segunda evaluación no ha permitido hacerlo.