

Proyecto DWECE – Aplicación Meteorológica con React y AEMET

1. Introducción

Este proyecto está planteado como un **trabajo práctico integrador** para el alumnado de **2º de Desarrollo de Aplicaciones Web**, dentro del módulo de **DWECE**.

El objetivo no es únicamente que la aplicación “funcione”, sino que el alumnado **demuestre criterio técnico**, capacidad de organización y comprensión real de cómo se construye una aplicación web moderna que consume datos de una API externa mediante una arquitectura cliente–servidor.

Para ello, se desarrollará una **aplicación en React** que consulte información meteorológica de la **API oficial de AEMET**, utilizando un **backend propio en Node.js con Express** que actuará como intermediario.

2. Objetivos didácticos

Al finalizar este proyecto, el alumnado deberá ser capaz de:

- Desarrollar una aplicación web en **React** estructurada en componentes.
 - Consumir datos de una API REST real de forma asíncrona.
 - Comprender y aplicar una arquitectura **frontend–backend** correctamente desacoplada.
 - Implementar un backend sencillo en **Node + Express**.
 - Gestionar estados de carga, error y ausencia de datos.
 - Aplicar criterios básicos de **usabilidad y diseño responsive**.
 - Documentar mínimamente un proyecto técnico.
-



C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
30019702@murciaeduca.es
www.cifpcarlos3.es



3. Tecnologías y herramientas obligatorias

El proyecto deberá realizarse utilizando obligatoriamente las siguientes tecnologías:

Frontend

- React (componentes funcionales)
- JavaScript ES6+
- HTML5 y CSS3

Backend

- Node.js
- Express
- API de AEMET (con clave personal)

Herramientas adicionales

- Git (control de versiones)
- Editor de código (VS Code o similar)

4. Descripción general del proyecto

La aplicación permitirá al usuario **consultar información meteorológica de España** a partir de distintos criterios (municipio, provincia, predicción diaria, etc.).

⚠ Aspecto clave del proyecto:

El frontend **NO debe comunicarse directamente con la API de AEMET**. Todas las peticiones deben realizarse:

React → Backend Express → API AEMET → Backend Express → React

Este punto es **obligatorio** y se tendrá muy en cuenta en la evaluación.

5. Requisitos funcionales mínimos

5.1 Estructura básica de la aplicación

La aplicación deberá contener, como mínimo:

- Una **cabecera o título** identificativo del proyecto.
- Un **formulario de búsqueda** meteorológica.
- Una **zona de resultados** claramente diferenciada.
- Mensajes visibles:
 - Cargando datos.
 - Error en la consulta.
 - Búsqueda sin resultados.

5.2 Tipos de búsqueda

La aplicación deberá permitir **al menos un tipo de búsqueda**, a elegir por el alumno:

- Por municipio
- Por provincia
- Por código de municipio
- Predicción diaria
- Predicción por horas

La implementación concreta es libre, pero deberá cumplir los siguientes criterios:

- La búsqueda debe ser **clara e intuitiva** para el usuario.
 - No se permitirán búsquedas vacías.
 - Los datos introducidos deberán validarse mínimamente.
-



C/Carlos III, 3
30201 CARTAGENA
TEL: 968 321301
30019702@murciaeduca.es
www.cifpcarlos3.es



ENSEÑANZAS
COFINANCIADAS POR
FONDO SOCIAL
EUROPEO

5.3 Información meteorológica

Se deberá mostrar información meteorológica **relevante y comprensible**, por ejemplo:

- Estado del cielo
- Temperatura máxima y mínima
- Probabilidad de precipitación
- Viento
- Fecha del pronóstico

No es obligatorio mostrar todos los datos disponibles, pero sí presentar la información de forma ordenada y coherente.

6. Backend (Node + Express)

Se proporcionará una **plantilla base de backend**, que deberá ser comprendida y adaptada por el alumno.

Requisitos obligatorios del backend

- La clave de la API de AEMET **no debe aparecer en el frontend**.
- El backend deberá:
 - Exponer rutas propias (por ejemplo `/api/tiempo/...`)
 - Consultar la API de AEMET
 - Procesar la respuesta
 - Devolver los datos al frontend en formato JSON

Se valorará positivamente:

- Uso de variables de entorno (`.env`)
 - Código organizado y legible
 - Manejo adecuado de errores
-

7. Frontend (React)

Requisitos mínimos

- Uso de **componentes funcionales**
- Uso de hooks (useState, useEffect, etc.)
- Separación lógica de componentes (buscador, resultados, tarjeta del tiempo...)
- Código ordenado y comprensible

No se evaluará el uso de librerías externas complejas; el objetivo es demostrar dominio de los **fundamentos de React**.

8. Diseño y experiencia de usuario

El diseño debe cumplir, como mínimo, los siguientes criterios:

- Interfaz clara y ordenada
- Tipografía legible
- Uso coherente del color
- Diseño **responsive** (móvil y escritorio)

Dentro de estos límites, el alumnado tiene **libertad creativa** para decidir el estilo visual de la aplicación.

9. Mejoras y ampliaciones (opcional)

Se valorará positivamente, aunque no es obligatorio, la inclusión de mejoras como:

- Varios tipos de búsqueda
- Selectores de provincias o municipios
- Iconos meteorológicos
- Histórico de búsquedas
- Modo oscuro
- Gráficas o visualización de datos

Estas mejoras **no compensarán** la falta de los requisitos mínimos.

10. Gestión de errores

La aplicación deberá gestionar correctamente:

- Errores de red
- Respuestas erróneas de la API de AEMET
- Búsquedas sin datos

El usuario **no debe ver errores técnicos sin tratar**.

11. Entrega del proyecto

La entrega se realizará mediante un **repositorio Git**, que deberá incluir:

- Código del frontend (React)
 - Código del backend (Node + Express)
 - Archivo **PDF** explicativo con:
 - Descripción del proyecto
 - Instrucciones de instalación y ejecución
 - Breve explicación de las decisiones técnicas tomadas
 - Conclusiones tras la realización del proyecto
-

12. Criterios de evaluación

En la evaluación se tendrá en cuenta:

- Funcionamiento correcto de la aplicación
- Uso adecuado de React y del backend
- Correcta comunicación frontend–backend
- Calidad y organización del código
- Gestión de errores
- Diseño y experiencia de usuario
- Mejoras y creatividad (si las hay)

13. Consideraciones finales

- El proyecto debe ser **original**.
- No se permite la copia directa de proyectos completos de Internet.
- El incumplimiento de los requisitos mínimos afectará negativamente a la calificación.

Este proyecto está pensado para que el alumnado **aprenda haciendo**, no para que memorice soluciones.

Se valorará especialmente el criterio técnico y el trabajo bien razonado.