

UTN RECONQUISTA

Asignatura: Gestión de desarrollo de Software

Docente: Eliana Suligoy

Alumnos: Dolzani Kevin, Lorenzón Nicolas, Zupel Sofia

Turno: Tarde

Introducción y Problema

En la actualidad, con los avances tecnológicos, las personas se informan cada vez más a través del celular, ya sea a través de redes sociales, aplicaciones o medios informativos. La mayoría de las personas consultan el clima a la hora de programar actividades, tanto para cosas básicas como realizar deporte o caminar, como para actividades con mayor antelación de tiempo como eventos, conciertos o fechas particulares.

Sin embargo, muchas aplicaciones meteorológicas presentan interfaces complicadas, exceso de información innecesaria o tiempos de carga prolongados. Esto nos llevó a desarrollar una solución más simple y directa: una aplicación web que muestre la información esencial del clima de manera clara y rápida, con un diseño moderno y fácil de usar.

Descripción

Como mencionamos en la introducción, nuestro trabajo consiste en el desarrollo de una aplicación web del clima llamada "Tiempo Ya". El objetivo es brindar datos meteorológicos en tiempo real para que las personas puedan consultar el clima de cualquier ciudad del mundo.

Características principales:

La aplicación incluye una pantalla de bienvenida con animaciones visuales (sol animado y nubes flotantes) que crea una primera impresión atractiva. Al presionar el botón "Comenzar", se accede mediante una transición animada a la interfaz principal de búsqueda.

En la interfaz principal, el usuario encuentra un campo de texto donde puede escribir el nombre de cualquier ciudad y un botón para consultar. La aplicación muestra los siguientes datos meteorológicos:

- Temperatura actual en grados Celsius
- Humedad relativa en porcentaje
- Índice UV para saber el nivel de radiación solar
- Estado del cielo (despejado, nublado, lluvia, etc.) en español

El diseño está orientado para que sea sencillo, fácil de consultar y entender. Se puede usar desde cualquier dispositivo (computadora, tablet o celular) ya que la interfaz se adapta automáticamente al tamaño de pantalla.

Objetivos Generales:

Desarrollar una aplicación web que proporcione información meteorológica en tiempo real de manera rápida y accesible, con una interfaz intuitiva que cualquier persona pueda usar sin dificultad.

Específicos:

Diseño accesible e intuitivo: Crear una interfaz que sea fácil de usar en diferentes dispositivos (computadoras, tablets y celulares).

Información confiable: Utilizar una fuente de datos meteorológicos reconocida (OpenWeatherMap API) para garantizar información exacta y actualizada.

Funcionalidades esenciales: Implementar la consulta de temperatura, humedad, índice UV e información del estado del cielo para cada ciudad buscada.

Optimización de rendimiento: Lograr que los tiempos de respuesta sean rápidos y que la aplicación consuma pocos recursos del dispositivo.

Experiencia visual agradable: Incorporar animaciones y un diseño moderno que hagan la aplicación visualmente atractiva sin complicar su uso.

Alcance

Lo que incluye el proyecto

El proyecto Tiempo Ya incluye un diseño web completo con pantalla de bienvenida animada y aplicación principal funcional, conectado a la API de OpenWeatherMap para obtener datos meteorológicos en tiempo real. La interfaz es completamente responsive, funcionando correctamente en computadoras, tablets y celulares. Los datos mostrados incluyen temperatura actual, índice UV, humedad relativa y estado del cielo, permitiendo la búsqueda por ciudad para consultar el clima de cualquier lugar del mundo. Además, incorporamos animaciones CSS que mejoran la experiencia visual, como nubes flotantes, sol animado y transiciones suaves entre pantallas.

Lo que NO incluye el proyecto

Para mantener el alcance del proyecto manejable, decidimos no incluir ciertas funcionalidades. No cuenta con pronóstico extendido de 7 días, mostrando únicamente el clima actual. Tampoco incluye geolocalización automática por GPS ni alertas o notificaciones sobre cambios climáticos. No guarda historial de búsquedas de ciudades consultadas anteriormente y solo funciona como aplicación web en el navegador, sin versión nativa para tiendas de aplicaciones móviles. Finalmente, al ser un proyecto académico, no incluye actualizaciones continuas ni soporte técnico posterior a la entrega.

Estas exclusiones se decidieron para mantener el proyecto dentro de un alcance manejable para el tiempo disponible y enfocarnos en ofrecer una funcionalidad principal de alta calidad.

Metodología de trabajo

El proyecto se llevó a cabo por tres integrantes. Para el desarrollo del sitio web utilizamos una metodología Scrum adaptada, debido al enfoque colaborativo que necesitábamos y la posibilidad de trabajar en secciones divididas por integrante.

Distribución del trabajo:

Dividimos las responsabilidades por áreas técnicas:

- HTML: Arquitectura y estructura de la aplicación
- CSS: Diseño visual, animaciones y adaptación responsive
- JavaScript: Funcionalidad, uso de la API y manejo de eventos

Cada uno se enfocó en su área pero trabajamos colaborativamente cuando era necesario integrar las partes o resolver problemas.

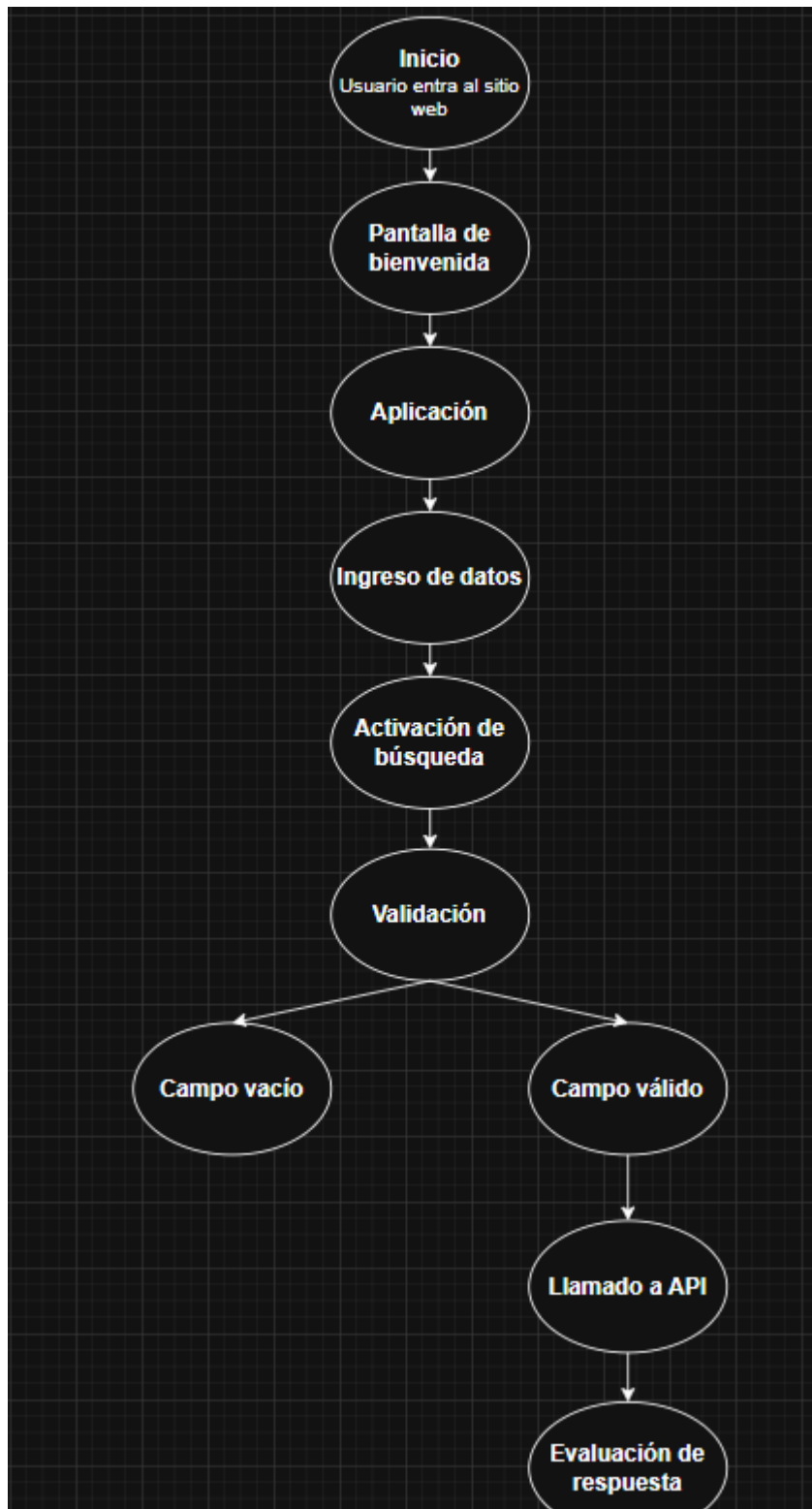
Proceso de desarrollo:

Trabajamos en ciclos cortos de aproximadamente una semana donde definíamos objetivos específicos. Durante la primera semana nos enfocamos en el diseño de la pantalla de bienvenida y la estructura básica del HTML, estableciendo las bases visuales del proyecto. En la segunda semana implementamos la funcionalidad de búsqueda y la conexión con la API de OpenWeatherMap, logrando obtener datos reales del clima. La tercera semana la dedicamos al desarrollo del diseño responsive y la optimización de las animaciones CSS para que funcionaran correctamente en diferentes dispositivos. Finalmente, en la cuarta semana realizamos pruebas exhaustivas, corrección de errores encontrados y los ajustes finales para pulir la aplicación antes de su entrega. Usamos desarrollo iterativo, donde cada componente creado fue probado individualmente antes de integrarlo al proyecto completo. Esto nos permitió detectar errores tempranamente y corregirlos sin afectar otras partes de la aplicación.

Herramientas utilizadas:

- Git y GitHub: Para trabajar colaborativamente en el código y hospedar el proyecto
- Visual Studio Code: Editor de código principal
- WhatsApp/Discord: Comunicación entre el equipo

Diagramas

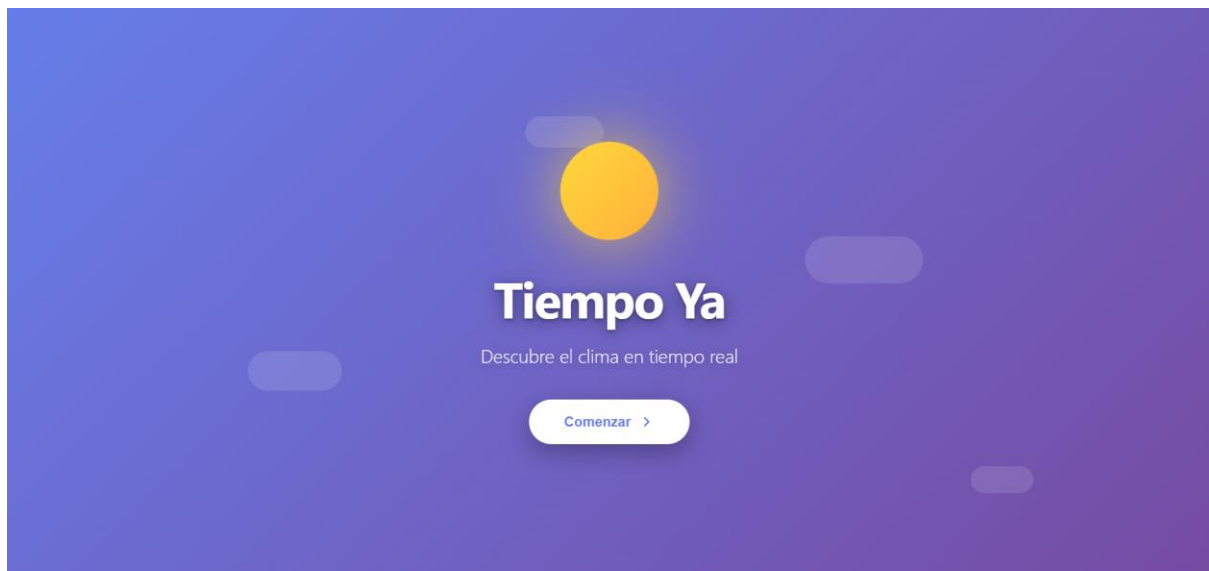


Funcionamiento

Paso 1: Acceso inicial

El usuario ingresa a la aplicación a través del navegador. Inmediatamente ve la pantalla de bienvenida que incluye:

- Un sol animado en el centro con efecto de pulsación
- Nubes que flotan de izquierda a derecha
- El título "Tiempo Ya"
- Un botón "Comenzar"



Paso 2: Inicio de la aplicación

Cuando el usuario hace clic en "Comenzar", la pantalla de bienvenida desaparece gradualmente y aparece la interfaz principal con una animación suave de transición.

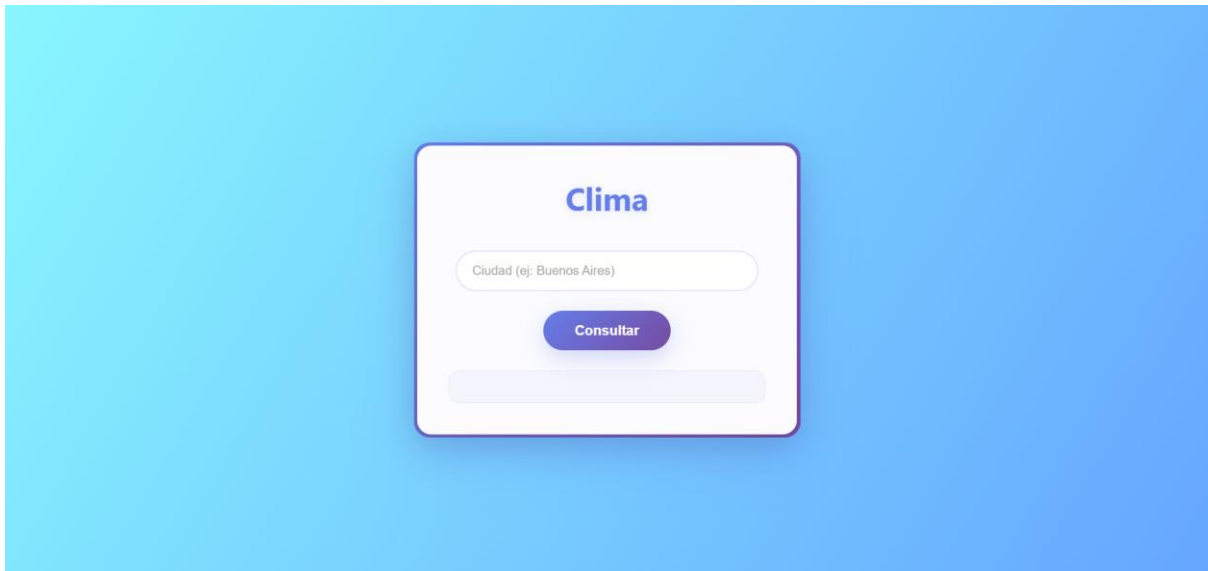
Paso 3: Búsqueda del clima

En la interfaz principal, el usuario ve:

- Un título "Clima"
- Un campo de texto con el placeholder "Ciudad (ej: Buenos Aires)"

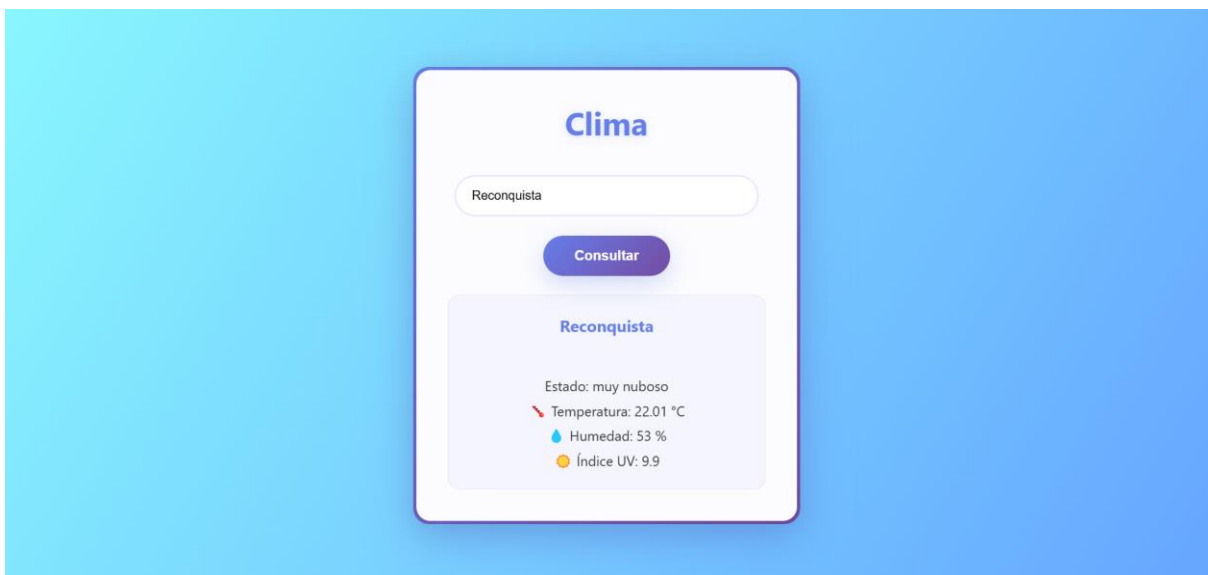
- Un botón "Consultar"

El usuario puede buscar de dos formas escribir el nombre de la ciudad y hacer clic en el botón o escribir el nombre y presionar Enter



Paso 4: Visualización de resultados

Una vez realizada la búsqueda, la aplicación muestra, los datos se presentan de forma clara con emojis para identificar cada tipo de información rápidamente.

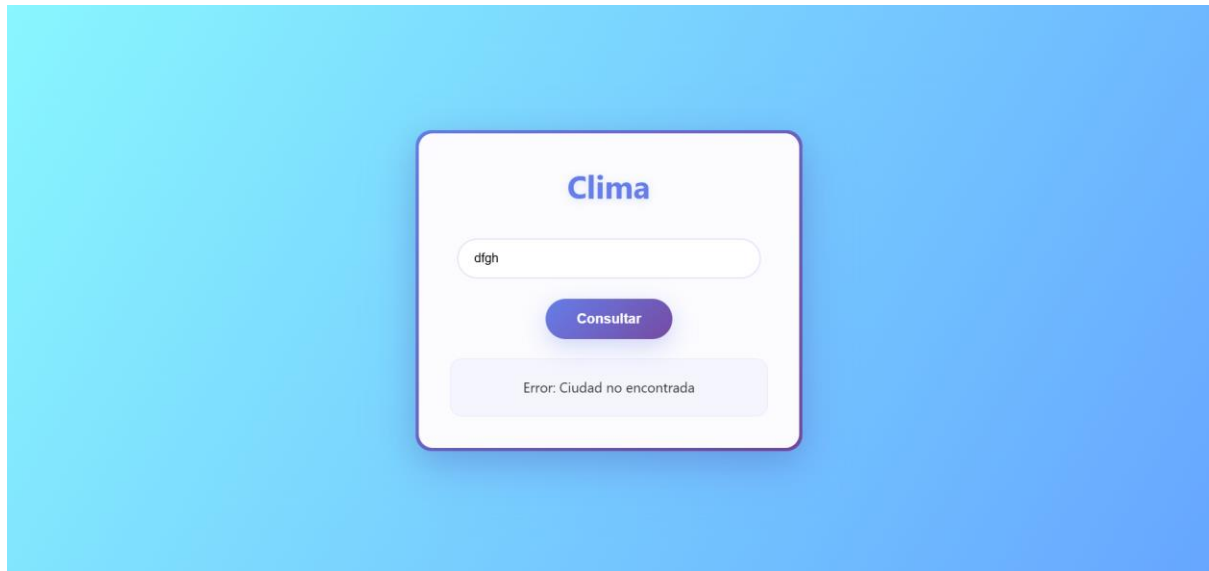


Paso 5: Manejo de errores

Si el usuario escribe una ciudad que no existe o hay un problema de conexión, aparece el mensaje:

"Error: Ciudad no encontrada"

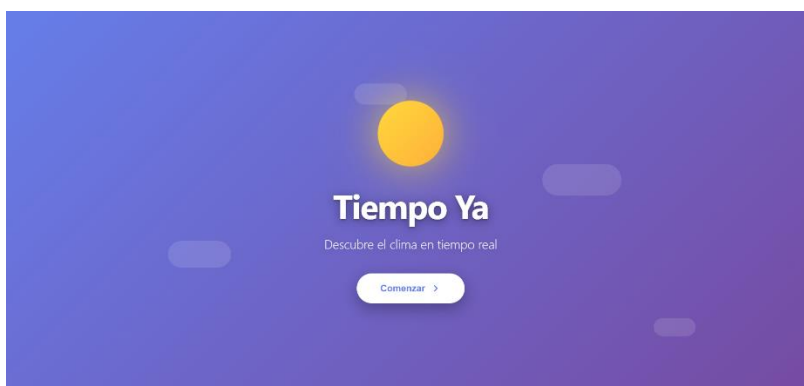
Esto permite que el usuario sepa que debe corregir el nombre de la ciudad e intentar nuevamente.



Adaptación responsive

La aplicación se adapta automáticamente al tamaño de pantalla:

- En computadoras: Interfaz centrada con diseño amplio
- En tablets: Elementos proporcionales y bien distribuidos
- En celulares: Diseño compacto optimizado para pantallas pequeñas



Detalle técnico

Lenguajes y tecnologías utilizadas

HTML: Utilizado para la estructura del documento y todos los elementos de la página, incluyendo formularios para la entrada de datos del usuario y la integración con scripts y hojas de estilo. Nos permitió organizar el contenido de forma semántica y clara.

CSS: Empleado para el diseño visual con gradientes modernos que le dan profundidad a la interfaz, animaciones para las nubes, el sol y las transiciones entre pantallas, y media queries para lograr un diseño responsive. También implementamos efectos hover en botones e inputs que responden a la interacción del usuario, además de sombras y bordes redondeados que dan un aspecto moderno y profesional.

JavaScript: Usado para toda la funcionalidad de la aplicación mediante funciones asíncronas con async/await, realizando llamadas a la API usando Fetch para obtener los datos meteorológicos. También nos permitió manipular el DOM para mostrar los resultados dinámicamente, manejar eventos como clicks y presionar Enter, y validar la entrada del usuario antes de realizar las búsquedas.

API utilizado

OpenWeatherMap API

Utilizamos dos endpoints de esta API:

- Weather Data: Para obtener temperatura, humedad y estado del cielo
(URL: <https://api.openweathermap.org/data/2.5/weather>)
- UV Index: Para obtener el índice de radiación UV :
(URL: <https://api.openweathermap.org/data/2.5/uvi>)

Esta API nos permite consultar información de más de 200,000 ciudades en todo el mundo y proporciona datos actualizados cada 10 minutos.

Mantuvimos una estructura simple con tres archivos principales, separando claramente la estructura (HTML), el diseño (CSS) y la funcionalidad (JavaScript).

Características técnicas destacadas

Animaciones CSS:

- Nubes flotantes con movimiento horizontal continuo (20 segundos por ciclo)
- Sol con efecto de pulsación que simula brillo
- Transiciones suaves entre pantallas
- Efectos hover en botones que elevan el elemento

Funcionalidad JavaScript:

- Validación de campo vacío antes de buscar
- Codificación correcta de nombres de ciudades con encodeURIComponent
- Manejo de errores con try-catch
- Dos llamadas API secuenciales (primera para clima, segunda para UV usando las coordenadas obtenidas)

Diseño responsive:

- Media query para pantallas menores a 768px
- Fuentes que se reducen proporcionalmente
- Elementos que se reajustan para pantallas pequeñas

Herramientas de desarrollo

- Visual Studio Code: Editor de código
- Git/GitHub: Control de versiones y hosting (GitHub Pages)
- GitHub Pages: Hosting gratuito para la aplicación web

URL del proyecto: <https://sofia59.github.io/clima/>

Conclusiones

El desarrollo de Tiempo Ya nos permitió aplicar los conocimientos adquiridos en la materia de Gestión de Desarrollo de Software, trabajando de forma colaborativa y organizada mediante metodología Scrum.

Logros alcanzados

El desarrollo de Tiempo Ya nos permitió aplicar los conocimientos adquiridos en la materia, logrando crear una aplicación funcional que cumple con todos los objetivos propuestos. Conseguimos implementar una interfaz moderna y atractiva que es fácil de usar, demostrando que es posible combinar estética con funcionalidad sin complicar la experiencia del usuario.

Uno de los mayores logros fue implementar exitosamente la conexión con una API externa (OpenWeatherMap), lo cual nos permitió obtener datos meteorológicos reales y actualizados de cualquier ciudad del mundo. El diseño responsive que desarrollamos permite usar la aplicación desde cualquier dispositivo, adaptándose perfectamente a computadoras, tablets y celulares. Además, las animaciones y transiciones que incorporamos mejoran significativamente la experiencia del usuario, haciendo que la aplicación se sienta fluida, profesional y agradable de usar.

Aprendizajes del proyecto

Durante el desarrollo aprendimos:

Durante el desarrollo del proyecto adquirimos varios aprendizajes importantes. Aprendimos a trabajar colaborativamente dividiendo responsabilidades por área técnica (HTML, CSS y JavaScript), lo que nos permitió avanzar de forma más eficiente y organizada. Descubrimos la importancia de probar cada componente antes de integrarlo al proyecto completo, ya que esto nos ayudó a detectar errores tempranamente y evitar problemas mayores.

También aprendimos cómo consumir APIs externas con JavaScript moderno usando `async/await` y `fetch`, lo cual fue fundamental para obtener los datos meteorológicos en tiempo real. Adquirimos técnicas de diseño CSS para crear interfaces modernas con animaciones, gradientes y efectos visuales que mejoran la experiencia del usuario.

Otro aprendizaje clave fue resolver problemas técnicos investigando documentación oficial, consultando recursos en línea y trabajando en equipo para encontrar soluciones entre todos. Y no podemos dejar de mencionar que la extensión de GitHub nos salvó, facilitando enormemente el control de versiones, la colaboración en el código y el hosting gratuito del proyecto en GitHub Pages.

Posibles mejoras futuras

Si continuáramos desarrollando el proyecto, existen varias funcionalidades que podrían mejorar la experiencia del usuario. Una mejora importante sería implementar un pronóstico extendido de 7 días, permitiendo a los usuarios planificar actividades con anticipación al mostrar temperaturas máximas, mínimas y probabilidad de lluvia para cada día de la semana.

La geolocalización automática del usuario detectaría la ubicación actual mediante GPS y mostraría el clima local sin necesidad de escribir la ciudad, haciendo la aplicación más conveniente para consultas rápidas. El historial de ciudades buscadas recordaría las últimas búsquedas, permitiendo acceder rápidamente a ubicaciones frecuentes sin tener que escribirlas nuevamente.

Incorporar gráficos de temperatura y humedad haría la información más visual mediante gráficos de barras o líneas que muestren las variaciones a lo largo del día. Finalmente, un modo oscuro/claro permitiría al usuario elegir el tema visual que prefiera, siendo el modo oscuro especialmente útil para consultas nocturnas ya que reduce el brillo de la pantalla.

Referencias

OpenWeatherMap API: <https://openweathermap.org/api>

CSS-Tricks: <https://css-tricks.com>

GitHub Pages Documentation: <https://pages.github.com>