José Antonio López Ávila

TFG 2º asir  IES LOS MANANTIALES

Weather web app

Contenido

[1 estudio del problema y análisis del sistema 2](#_Toc128342881)

[1.1 Introducción 2](#_Toc128342882)

[1.2 Finalidad 2](#_Toc128342883)

[1.3 Objetivos 2](#_Toc128342884)

[2 modelado de la solución 3](#_Toc128342885)

[2.1 recursos humanos 3](#_Toc128342886)

[2.2 recursos hardware 3](#_Toc128342887)

[2.3 recursos software 3](#_Toc128342888)

[3 planificación 3](#_Toc128342889)

[4 fase de pruebas 4](#_Toc128342890)

[5 conclusiones finales 4](#_Toc128342891)

[5.1 Grado de cumplimiento de los objetivos fijados 4](#_Toc128342892)

[5.2 Propuesta de modificaciones o ampliaciones futuras 4](#_Toc128342893)

[6 documentación del sistema desarrollado 4](#_Toc128342894)

[6.1 Manual de instalación 4](#_Toc128342895)

[6.2 Manual de uso 4](#_Toc128342896)

[7 bibliografía 4](#_Toc128342897)

[7.1 Cursos realizados: 4](#_Toc128342898)

[7.2 tecnologías usadas 5](#_Toc128342899)

[7.3 Inspiración 5](#_Toc128342900)

[7.4 dudas puntuales 5](#_Toc128342901)

# 1 estudio del problema y análisis del sistema

## 1.1 Introducción

Es una app web del clima que permite ver y comparar estadísticas del tiempo de diferentes localidades.

## 1.2 Finalidad

Hacer uso de peticiones a API REST mediante JavaScript controlando los posibles errores del protocolo HTTP.

Hacer uso de la API de geolocalización controlando posibles errores

Hacer uso de la API de web storage controlando posibles errores

Hacer la APP responsive

Manipular el DOM desde JavaScript

Hacer cambios a la app desde JavaScript haciendo uso de clases CSS

Trabajar una metodología CSS: BEM

Diseñar la interfaz de usuario desde FIGMA

Seguridad con las APIs Keys

## 1.3 Objetivos

Haciendo uso de la API de geolocalización nos mostrara los datos de nuestra localización actual si el usuario lo permite.

Haciendo uso de la API de almacenamiento web tendremos un registro de las ciudades que hemos visualizado con alguna información de la misma. Este registro se podrá eliminar mediante un botón.

Mediante un botón podremos cambiar entre grados Celsius a Fahrenheit y viceversa.

Mostrar la información del tiempo junto a la fecha y hora de la consulta.

Cambiar el fondo de la página en función de la hora del día en la que se encuentre la localización mostrada mediante funciones del objeto date de JavaScript.

La app se debe de mostrar bien independientemente del dispositivo mediante mediaqueries en CSS.

Mostrar la información del clima de forma clara y amigable para el usuario

Mostrar la información de una ciudad que introduzca el usuario

Mostrar un popup personalizado cuando haya errores: (cuando el usuario introduzca mal los datos, que no de permiso de geolocalización, que haya algún error por parte de la API etc)

# 2 modelado de la solución

## 2.1 recursos humanos

2 personas: una se encargaría del diseño y otra de la programación e implementación del diseño o una sola persona que se encargue de ambas cosas.

## 2.2 recursos hardware

Un hosting para tener operativa la app, DNS, HTTPS.

Un equipo para programar y diseñar la app.

Varios dispositivos para comprobar que la app se muestre correctamente: Tablet, ordenadores, móvil

## 2.3 recursos software

Un editor de código

Servidor web local para hacer las pruebas

Un navegador web moderno

Prepros o extensiones para el editor de código que comprima el código CSS y añada los prefijos CSS para cada navegador según sea necesario.

Git para controlar las versiones del código que escribamos

La API a la que hacer las peticiones del clima openweathermap

# 3 planificación

Mediante el uso de herramientas de productividad **toggl track** y **notion** me fui organizando las semanas para hacer todas las necesidades del proyecto.

# 4 fase de pruebas

Hice varias pruebas desde el inspector de elementos del navegador con la vista de desde otros dispositivos.

Hice pruebas reales desde mi teléfono, tablet y ordenadores personales

Hice pruebas desactivando la geolocalización, forzando a no tenerla disponible

Hice pruebas con los diferentes códigos de estado del protocolo HTTP forzando a que aparecieran errores.

Hice pruebas haciendo uso de navegadores antiguos que no dispongan de web storage

# 5 conclusiones finales

## 5.1 Grado de cumplimiento de los objetivos fijados

## 5.2 Propuesta de modificaciones o ampliaciones futuras

# 6 documentación del sistema desarrollado

## 6.1 Manual de instalación

## 6.2 Manual de uso

# 7 bibliografía

## 7.1 Cursos realizados:

[HTML](https://www.youtube.com/playlist?list=PLROIqh_5RZeB92ME1GFyeqDVOa-gL0Ybd)

[CSS](https://www.youtube.com/playlist?list=PLROIqh_5RZeDbvISffzihyxzqJBt_z3-Z)

[JavaScript](https://www.youtube.com/playlist?list=PLROIqh_5RZeBAnmi0rqLkyZIAVmT5lZxG)

[FIGMA](https://www.youtube.com/playlist?list=PLvq-jIkSeTUbpfewvbKLhHctdkgadAy-K)

## 7.2 tecnologías usadas

[Documentación API openweathermap](https://openweathermap.org/current)

[mdn: objeto date](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Date)

[mdn: API de geolocalización](https://developer.mozilla.org/es/docs/Web/API/Geolocation_API)

[Ejemplo con geolocalización](https://youtu.be/mONXtKKaV_w)

[mdn: API localStorage](https://developer.mozilla.org/es/docs/Web/API/Window/localStorage)

[Ejemplo con localstorage y sessionstorage](https://youtu.be/ASQQUSFtr8g)

[Funcionamiento de las promesas, async/await](https://www.youtube.com/watch?v=rKK1q7nFt7M)

[mdn: función fetch](https://developer.mozilla.org/es/docs/Web/API/fetch)

[Metodologías CSS y explicación de BEM](https://youtu.be/6LUz0MnNCiA)

[Iconos SVG animados](https://www.amcharts.com/free-animated-svg-weather-icons/)

## 7.3 Inspiración

[Comunidad de FIGMA: Interfaces Apps clima](https://www.figma.com/community/search?resource_type=mixed&sort_by=relevancy&query=weatherapp&editor_type=all)

## 7.4 dudas puntuales

[mdn](https://developer.mozilla.org/es/docs/Web/JavaScript/)

[JavaScript info: libro con mucha información](https://es.javascript.info/)

[stack overflow](https://stackoverflow.com/)