React Conectado: ***My Weather App***

Fechas de entrega: 30 de Noviembre de 2021

Contexto

Como hemos visto, ReactJS es una librería que nos ayuda a crear interfaces de usuario de una manera más ágil y robusta.

Para que ReactJS se nutra de información, necesitamos de un servidor en internet que nos los proporcione.

Todas las aplicaciones web SPA se basan en el siguiente flujo de pasos, desde el punto de vista del usuario:

1. Navego a la ruta donde se encuentra la información que quiero ver.
2. La página solicita mediante `***fetch***` la información a un backend
3. Una vez obtiene la información renderiza en componentes la página para mostrar en el HTML la información que desea ver.

Como todavía no sabemos crear backends, nos vamos a nutrir de uno gratuito que nos proporciona la información sobre meteorológica de una ciudad (tiempo, viento, etc) de los próximos 7 días.

El backend en concreto que vamos a utilizar para crear una aplicación del tiempo es:

https://openweathermap.org/

Este API nos proporciona información de muchos tipos:

- Tiempo actual de una ciudad
- Previsión para los próximos días
- Radiación solar
- Alertas del tiempo mundiales
- Alertas en carreteras
- Histórico del tiempo
- etc.

Dentro de todas las posibilidades, la mayoría son de pago, pero tenemos una que se puede utilizar de manera gratuita. Pricing - OpenWeatherMap

Principalmente vamos a usar dos endpoints:

- Current weather data - OpenWeatherMap
- One Call API: weather data for any geographical coordinate - OpenWeatherMap

Objetivos

Afianzar los conceptos para usar componentes de función y hooks.

Afianzar la creación de aplicaciones ReactJS que obtienen sus datos de un backend.

Reforzar el aprendizaje para el uso de contexto de ReactJS a través de sus hooks.

Aprender la diferencia entre componentes "***Stateless vs Statefull***".

Dominar el uso de fechas en JavaScript.

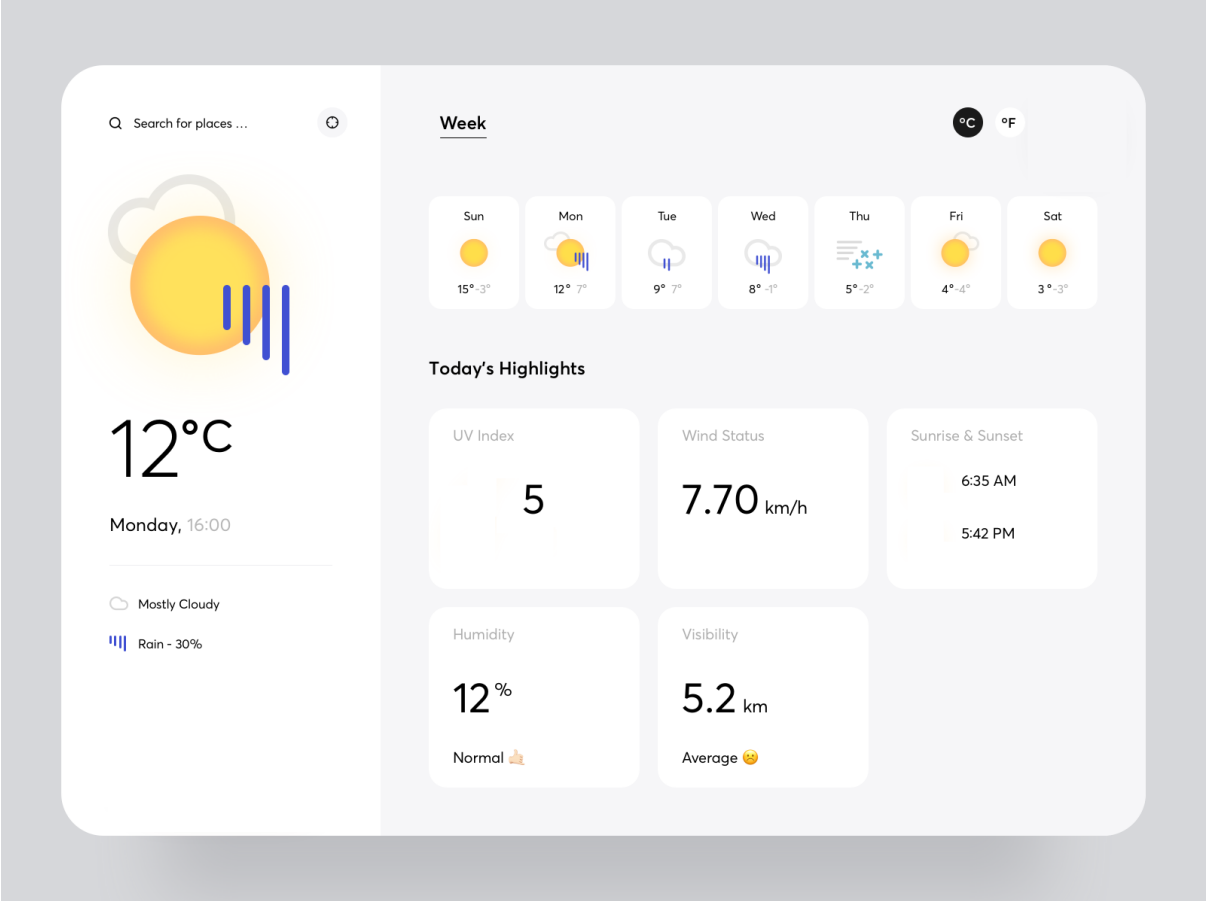
Requisitos

Para realizar esta práctica se han de tener conocimientos de:

- HTML
- CSS
- JS
- ReactJS:
    - Creación de una aplicación
    - Creación de componentes de clase y de función
    - Consumir un componente dentro de otro componente
    - Pasar propiedades de un componente padre a un componente hijo
    - Utilizar el estado de un componente de reactjs
    - Pasar información de un componente hijo a un padre
    - Ciclo de vida de un componente ReactJS
    - Uso de Hooks (useState, useEffect, useContext y useRef)

Diseños (ORIENTATIVOS)

La aplicación tendrá una sola página que deberá visualizarse como la siguiente imagen:

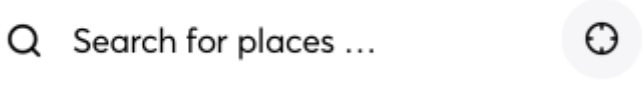


La página muestra una card principal con toda la información del tiempo de los próximos 7 días y la información meteorológica principal del día actual.

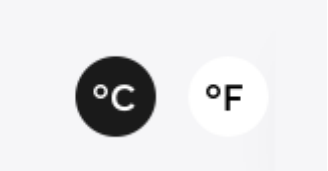
La card principal está dividida en dos partes:

- La información actual
- la información de el resto de días de la semana + datos extra del día actual

Por otro lado tenemos un buscador, que nos permite buscar la ciudad de la cual queremos obtener la información del tiempo o seleccionar la geolocalización actual:



Por último la aplicación dispone de un selector que nos hace cambiar la web de Celsius a Fahrenheit, según tengamos seleccionado uno u otro.



Fuente

- Vamos a utilizar Roboto - Google Fonts

Colores

- Color de fondo de la página: `#D6D7DB`
- Color de fondo del lado izquierdo de la card general + las mini-cards: `#FFFFFF`
- Color de fondo del lado derecho de la card general: `#F7F6F9`
- Color de letra oscura y unidades seleccionadas (ºC): `#1A1A1A`
- Color de letra clara: `#CECECE`

Imágenes

- La imágenes que se utilizan, nos la va a proporcionar el API. ver API Images

Entendiendo el API

El API es un backend que tiene una parte gratuita que nos proporciona información sobre la previsión del tiempo de una ciudad y además nos proporciona las imágenes de cada una de las situaciones meteorológicas.

El API es muy extensa, pero nosotros utilizaremos los endpoints (url's) gratuitos necesarias para mostrar la aplicación. En concreto necesitaremos:

- One Call API: weather data for any geographical coordinate - OpenWeatherMap
- Current weather data - OpenWeatherMap (Para recuperar las coordenadas de una ciudad por su nombre)

Generando un API KEY

La mayoría de librerías externas y backends "públicos" necesitan de la creación de una cuenta para poder utilizarse.

Una vez creado el usuario en la página, debemos generar una clave que nos identifique como usuarios y que utilizaremos para realizar las peticiones HTTP.

A esta clave se la conoce como API KEY.

Para generar un API KEY en nuestro backend del tiempo debemos seguir su documentación.
How to start to work with Openweather API - OpenWeatherMap

**** Nota: Nunca publiquéis en un repositorio público vuestra API KEY. Cuidado al hacer push al servidor siempre eliminar ese valor**

Request

Para obtener la información necesaria en nuestra aplicación necesitamos realizar llamadas HTTP al API de tipo GET.

Cada API tiene documentada la forma en la que hay que llamarla. Lo primero que necesitamos es entender cómo podemos llamar al API y saber que API's vamos a necesitar. En este caso serían las siguientes:

- One Call API: weather data for any geographical coordinate - OpenWeatherMap
- Current weather data - OpenWeatherMap (Para recuperar las coordenadas de una ciudad por su nombre)

Un ejemplo del primer API es:

```
https://api.openweathermap.org/data/2.5/onecall?lat={lat}&lon={lon}&exclude={part}&appid={API key}
```

Donde:

- lat: la latitud del lugar donde queremos obtener el tiempo
- lon: la longitud del lugar donde queremos obtener el tiempo
- exclude: lo que queremos excluir de la respuesta. En nuestro caso solo necesitamos la información de los días y el current, por lo que podemos excluir lo demás
- appid: el API Key

Response

Una vez que realizamos la petición, si todo ha ido bien, obtendremos una respuesta con la información solicitada.

En la documentación tenemos una descripción de lo que significa cada campo. Por ejemplo:

```
current.feels_like : es la información sobre la sensación térmica
```

Cada entidad de la respuesta viene con su fecha asociada en un campo ***"dt"***, que es el timestamp de esa fecha.

Para poder tener en JS el objeto Date de ese dt, se necesita crear la fecha de la siguiente manera:

const date = new Date(dt * 1000);

Imágenes

El propio backend tiene un banco de imágenes para las diferentes situaciones meteorológicas.

Cuando solicitamos la información, nos viene un objeto del cual podremos sacar el identificador del icono en el backend. Este atributo suele llamarse:

`icon, ej: current.weather.icon`

En esta guía podéis entender la forma de generar la URL para la obtención de los iconos para cada situación https://openweathermap.org/weather-conditions

Os recomiendo que siempre busquéis como mínimo las imágenes con densidad de pixel 2x. EJ:

https://openweathermap.org/img/wn/10d@2x.png

o también podéis coger la de 4x:

https://openweathermap.org/img/wn/10d@4x.png

Funcionalidades

Aplicación Base

- Cuando la aplicación cargue deberá obtener la información del tiempo de la geolocalización del usuario.
- La aplicación debe visualizar la información meteorológica del día actual en el panel de la izquierda
    - Imagen meteorológico actual
    - Temperatura actual
    - Día de la semana y hora de la información meteorológica
    - Descripción
    - Probabilidad de lluvia
- La aplicación debe mostrar el listado con la información meteorológica diaria de los siguientes 7 días:
    - Imagen con el tiempo
    - Temperatura Máxima
    - Temperatura Mínima
- La aplicación debe mostrar un panel con la información relevante del día actual:
    - Índice de Radiación ultravioleta
    - Velocidad del viento
    - Hora de amanecer y puesta del sol

- Humedad
- Visibilidad

Buscando por ciudades

- Tenemos un buscador en el panel izquierdo que nos debe permitir introducir el nombre de una ciudad.
- Cuando se pulse intro, se realizará la búsqueda para la ciudad introducida y se mostrará la información de la nueva ciudad
- Si el usuario pulsa en el botón de localización, la aplicación buscará la información meteorológica de las coordenadas actuales del usuario.

Cambiando las unidades

- En la parte superior derecha, el usuario podrá seleccionar las unidades de medida en las cuales quiere recibir la información.
- La aplicación actualizará su vista para que el usuario visualice la información meteorológica en esas unidades de medida.

Entrega

La entrega se realizará el 30 de Noviembre y se expondrá a los compañeros en clase.

Se puede mostrar la aplicación siendo ejecutada en local.

En caso de querer desplegarla en algún servicio de hosting (Heroku) se debe tener en cuenta