

UNIVERSIDAD INTERAMERICANA DE PANAMÁ

FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO

MATERIA: PROGRAMACION GRAFICA

Proyecto final

Docente: Maryon Torres

Estudiantes:

Dylan Pérez 4-818-171

Alexis López 8-858-1068

2021

**Lecciones aprendidas.**

Esta Aplicación se basa en tener la información del clima del día actual y los días venideros, utilizando la información provista de un openweather por medio de un API. Esta API requiere la longitud y latitud para obtener los datos climáticos de dicho sitio.

Para conseguir la longitud y latitud se utiliza la información GPS del dispositivo y se extrae esta información. Para ser mas entendible se incluye una imagen descriptiva sobre el clima actual.

Se utilizo React para realización de esta aplicación.

React es una librería completa, adecuada en muchos tipos de proyectos distintos. Nos permite un desarrollo ágil, ordenado y con una arquitectura mantenible, focalizada en componentes y que nos ofrece un gran rendimiento.

Haciendo este producto utilizamos el contexto hook y state de React, Los Hooks son funciones que te permiten enganchar el estado de React y el ciclo de vida desde componentes de función. Los hooks no funcionan dentro de las clases — te permiten usar React sin clases.

Utilizamos los hooks para actualizar el contenido de la pagina una vez ya se cargado la información del clima.

Aprendimos a llamar un API y Traer información de ella, para esto se utilizo el API de openweather, que tiene una base de datos para el clima de una gran cantidad de lugares.

Esta aplicación toma la información de tu ubicación y te el informe del clima basado en tu área. Incluso para los días de siguientes de la semana.

Se utilizaron 3 componentes dentro de la aplicación, aprendimos a como incrustar múltiples componentes dentro de un pagina.

**Parte del Código**

App.js

import './App.css';

import React, { useEffect, useState } from "react";

import { Dimmer, Loader } from 'semantic-ui-react';

import Weather from './components/weather';

import Forecast from './components/forecast';

import Imagen from './components/imagen'

export default **function** App() {

**const** [lat, setLat] = useState([]);

**const** [long, setLong] = useState([]);

**const** [weatherData, setWeatherData] = useState([]);

**const** [forecast, setForecast] = useState([]);

**const** [error, setError] = useState(null);

  useEffect(() **=>** {

*//navigator.geolocation.getCurrentPosition(function(position) {*

        setLat(8.987255);

*//console.log(position.coords.latitude);*

        setLong(-79.516017);

*//console.log(position.coords.longitude);*

*// });*

      getWeather(8.987255, -79.516017)

      .then(weather **=>** {

        setWeatherData(weather);

        setError(null);

      })

      .catch(err **=>** {

        setError(err.message);

      });

      getForecast(8.987255, -79.516017)

        .then(data **=>** {

          setForecast(data);

          setError(null);

        })

        .catch(err **=>** {

          setError(err.message);

        });

  }, [lat,long,error])

**function** handleResponse(response) {

    if (response.ok) {

      return response.json();

    } else {

      throw new Error("Please Enable your Location in your browser!");

    }

  }

**function** getWeather(lat, long) {

    return fetch(

      `${process.env.REACT\_APP\_API\_URL}/weather?lat=${lat}&lon=${long}&units=metric&appid=${process.env.REACT\_APP\_API\_KEY}`

    )

      .then(res **=>** handleResponse(res))

      .then(weather **=>** {

        if (Object.entries(weather).length) {

**const** mappedData = mapDataToWeatherInterface(weather);

          console.log(mappedData);

          return mappedData;

        }

      });

  }

**function** getForecast(lat, long) {

    return fetch(

      `${process.env.REACT\_APP\_API\_URL}/forecast?lat=${lat}&lon=${long}&units=metric&appid=${process.env.REACT\_APP\_API\_KEY}`

    )

      .then(res **=>** handleResponse(res))

      .then(forecastData **=>** {

        if (Object.entries(forecastData).length) {

          return forecastData.list

            .filter(forecast **=>** forecast.dt\_txt.match(/09:00:00/))

            .map(mapDataToWeatherInterface);

        }

      });

  }

**function** mapDataToWeatherInterface(data) {

    console.log(data);

**const** mapped = {

      date: data.dt \* 1000, *// convert from seconds to milliseconds*

      description: data.weather[0].main,

      temperature: Math.round(data.main.temp),

      main: data.weather[0].main,

      wt:data

    };

*// Add extra properties for the five day forecast: dt\_txt, icon, min, max*

    if (data.dt\_txt) {

      mapped.dt\_txt = data.dt\_txt;

    }

    return mapped;

  }

  return (

    <div className="App">

      <div className="barra">

        <p className="header">TIEMPONOTI</p>

      </div>

      {(typeof weatherData.main != 'undefined') ? (

        <div className="Contenedor">

            <div className="Clima">

              <Weather weatherData={weatherData.wt}/>

              <Forecast forecast={forecast} weatherData={weatherData}/>

            </div>

            <div className="Imagen">

              <Imagen ima={weatherData.wt} />

          </div>

      </div>

      ): (

        <div>

          <Dimmer active>

            <Loader>Cargando Informacion..</Loader>

          </Dimmer>

        </div>

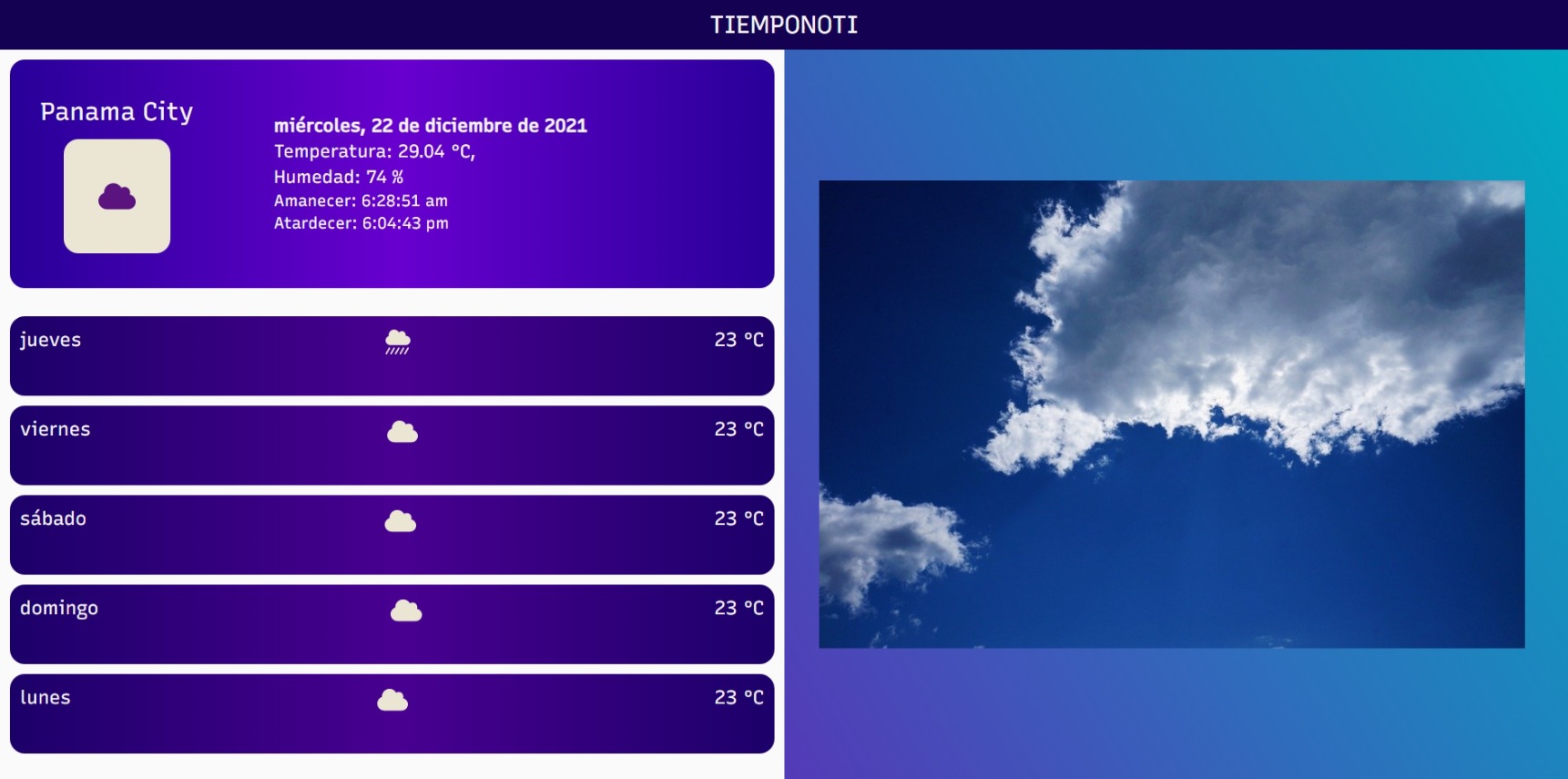
      )}

    </div>

  );

}

**IMAGEN**

****