

**Universidad Politécnica Salesiana**

INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

# APLICACIÓN DEL CLIMA



*Informe 03*

Autor:  
Ricardo Romo

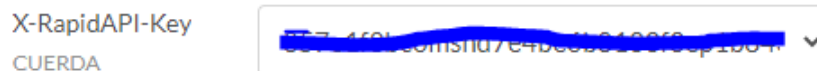
June 16, 2020

## PROBLEMA

Se desea crear una aplicación que reciba los parámetros de -c (ciudad a la cual se desea sacar la información), -o (opciones p=presión, h=humedad, su demanda no es necesaria), y la cual muestre por consola, la ciudad, la temperatura de esta, si se envía el parámetro -o p se mostrara la presión y si es -o h la humedad, para realizar este proceso deberá conectarse a una API. Antes de comenzar lo primero que debemos de hacer es crearnos una cuenta dentro de **Rapid API** <https://rapidapi.com/>



Luego iniciamos sesión y nos dirigimos a la API de **Open Weather Map** <https://rapidapi.com/community/api/open-weather-map>, esta API nos entrega en tiempo real la temperatura, presión, humedad, entre otros datos sobre una ciudad que se elija, este mismo api nos entrega un identificador el cual nos servirá para hacer peticiones a esta API.



Y ya con esto podemos comenzar

1. Comenzamos iniciando npm con los comandos **npm init**
2. Luego instalamos nuestras dependencias que en este caso usaremos las siguientes:
  - (a) **npm install axios save**
  - (b) **npm install yargs save**

Una vez ya instaladas nuestras dependencias proseguimos a crea una carpeta llamada controlador y dentro de esta un archivo llamado **clima.js**. Este **clima.js** va a ser nuestro controlador, es decir va a realizar los pedidos a la API.

#### **clima.js**

1. Primero llamamos a nuestro modulo Axios con el comando require
2. Creamos la función asyn getClima() el cual recibiremos el nombre de la ciudad, y la opción de los datos que se necesita siendo p para presión y h para humedad.
3. Utilizamos el comando **encodeURIComponent** para codificar la ciudad en modo URL para que la petición get pueda procesarla
4. Llamamos a nuestro modulo axios.get() para realizar una petición tipo get enviando como parámetros nuestra **key** proporcionada por la API en la página **appid** y la ciudad que deseamos consultar la información **q**
5. Una vez obtenidos los resultados guardamos en un objeto y retornamos dependiendo la respuesta que necesitemos
6. Por último exportamos esta función con **module.exports**

```
1  const axios = require('axios');
2  const getClima = async(city, option) => {
3      const cityURL = encodeURIComponent(city)
4      const resp = await axios.get(`https://
      api.openweathermap.org/data/2.5/weather?
      appid=94162bb23f3c7370cf2eff66e4d17814&q=${cityURL}&
      units=metric`);
5      switch (option) {
6          case 'p':
7              return ({ temperatura: resp.data.main.temp,
                          presion: resp.data.main.pressure })
8          case 'h':
9              return ({ temperatura: resp.data.main.temp,
                          humedad: resp.data.main.humidity })
10         default:
11             return ({ temperatura: resp.data.main.temp,
                          presion: resp.data.main.pressure, humedad:
                          resp.data.main.humidity })
12     }
13 }
14
15 module.exports = {
16     getClima
17 }
```

Ahora crearemos en la carpeta raíz nuestro main que lo nombraremos como **app.js**

1. Comenzamos llamando a nuestro modulo **yargs** con el comando `require`, seguido por el comando **.options** en el cual se enviaran todos los datos a ser ingresados por el usuario , en este caso sería **city,option**, teniendo cada uno su propia configuración, al final, enviamos `.argv` para recibir estos argumentos.
2. Llamaremos también a nuestro controlador ya exportado y lo nombraremos como **weather**
3. Crearemos una función asyn llamada `get_information()`, el cual recibirá los parámetros de la ciudad.
4. Creamos la variable `values` la cual recibirá de nuestro controlador **clima.js** los datos que se pidió.
5. al final llamamos a la función `get_information()`, enviamos la ciudad, e imprimimos la respuesta dentro de una tabla.

```
1  const argv = require('yargs').options({
2    city: {
3      alias: 'c',
4      description: "Name if the city",
5      demand: true
6    },
7    option: {
8      alias: 'o',
9      description: "presion o humedad",
10   }
11  }).argv;
12
13
14  const weather = require('./controlador/clima')
15  const get_information = async(city) => {
16    try {
17      const values = await weather.getClima(argv.city,
18        argv.option);
19      return values
20    } catch (error) {
21      return `la ciudad ${city} no se encuentra`;
22    }
23  }
24
25  get_information(argv.city).then(resp => {
26    console.log(`DATOS DE LA CIUDAD: ${argv.city}`);
27    console.table(resp);
28  }).catch(console.log)
```

---

## EJECUCIÓN 1

```
$ node app.js -c QUITO
DATOS DE LA CIUDAD: QUITO
```

| (index)     | Values |
|-------------|--------|
| temperatura | 14     |
| presion     | 1028   |
| humedad     | 87     |

## EJECUCIÓN 2

```
$ node app.js -c QUITO -o p
DATOS DE LA CIUDAD: QUITO
```

| (index)     | Values |
|-------------|--------|
| temperatura | 14     |
| presion     | 1028   |

## EJECUCIÓN 3

```
$ node app.js -c QUITO -o p
DATOS DE LA CIUDAD: QUITO
```

| (index)     | Values |
|-------------|--------|
| temperatura | 14     |
| presion     | 1028   |

## CONCLUSIÓN

- Las conexiones a APIS es un recurso necesario para los servicios web, debido a que estas dos se complementan, es decir la una obtiene información, y la otra procesa la información realizando las respectivas operaciones que desee el usuario.

## CÓDIGO

<https://github.com/rromom/clases-web/tree/master/codigos>