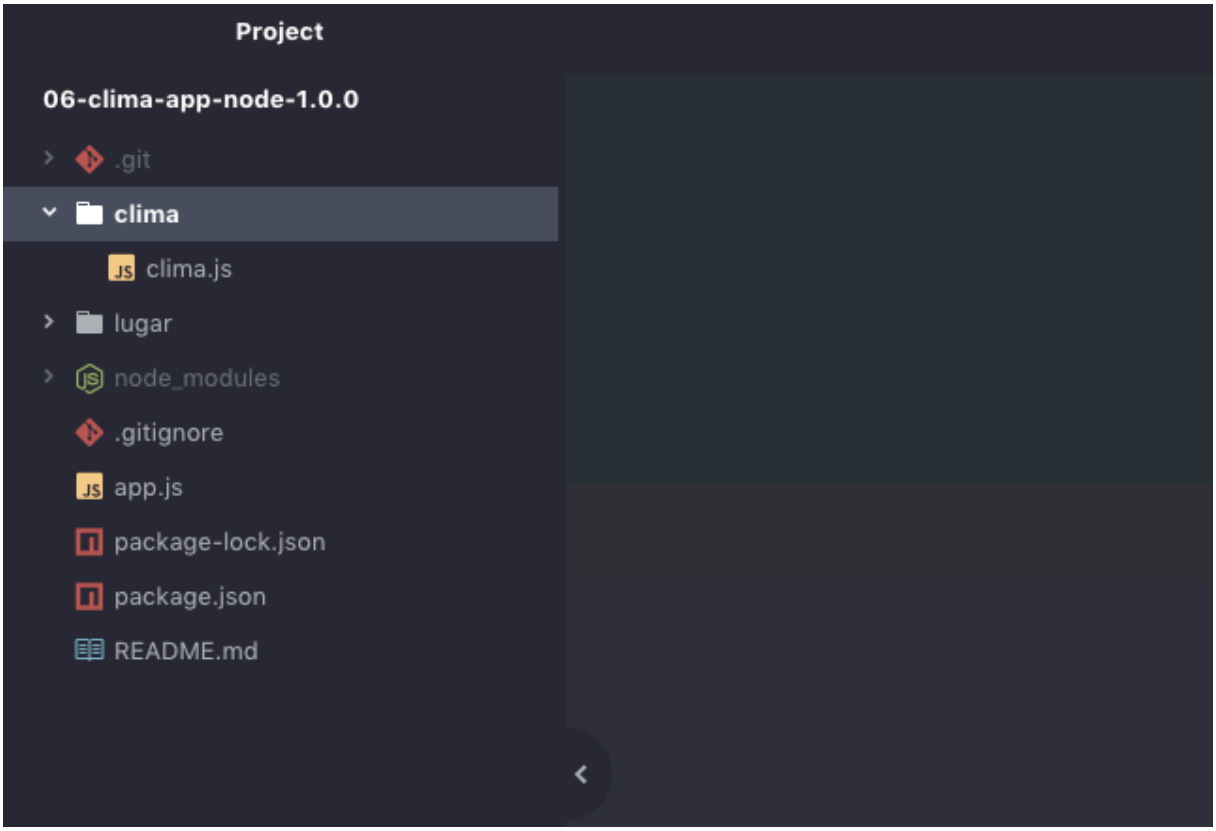
Licenciatura en Sistemas de Información

Programación Avanzada

API-Rest con NodeJS – Persistencia

Ejercicio

En base del modelo de la clase, consumo de Apis con axios, implemente un consumo de datos de la pagina de https://openweathermap.org/api , el modelo deberá tomar los datos del clima de la ciudad que demos ingreso por consola implementando también Yargs.



b) optimice el código para que solo en app.js sean llamadas las funciones del clima y lugar, en base al modelo propuesto utilizando async-away.

app.js

```
const lugar = require('./lugar/lugar');
const clima = require('./clima/clima');

const argv = require('yargs').options({
    direccion: {
        alias: 'd',
        desc: 'Dirección de la ciudad para obtener el clima',
        demand: true
    }
}).argv;


let getInfo = async(direccion) => {

    try {

        let coors = await lugar.getLugarLatLng(direccion);
        let temp = await clima.getClima(coors.lat, coors.lng);
```

Para modularizar nuestro código generamos 2 archivos, “lugar”, “clima”.

Lugar:

lugar.js

```
const axios = require('axios');


const getLugarLatLng = async(direccion) => {

    let encodedUrl = encodeURI(direccion);

    let resp = await axios.get(`https://maps.googleapis.com/maps/api/geocode/json?address=${ encodedUrl }&key=AIzaSyDzbQ_553v-n8QNs2aafN9QaZbBy`

    if (resp.data.status === 'ZERO_RESULTS') {
        throw new Error(`No hay resultados para la ciudad ${ direccion }`);
    }

    let location = resp.data.results[0];
    let coors = location.geometry.location;

    return {
        direccion: location.formatted_address,
        lat: coors.lat,
        lng: coors.lng
    }

}

module.exports = {
    getLugarLatLng
}
```

Clima:

clima.js

```
const axios = require('axios');


const getClima = async(lat, lng) => {

    let resp = await axios.get(`http://api.openweathermap.org/data/2.5/weather?lat=${ lat }&lon=${ lng }&units=metric&appid=f369635965b00ad16ce

    return resp.data.main.temp;
}


module.exports = {
    getClima
}
```

En app.js, solo quedara el siguiente código:

Project

```
06-clima-app-node-1.0.0
  .git
  clima
    clima.js
  lugar
    lugar.js
  node_modules
  .gitignore
  app.js
  package-lock.json
  package.json
  README.md
```

app.js

```
const lugar = require('./lugar/lugar');
const clima = require('./clima/clima');

const argv = require('yargs').options({
    direccion: {
        alias: 'd',
        desc: 'Dirección de la ciudad para obtener el clima',
        demand: true
    }
}).argv;


let getInfo = async(direccion) => {

    try {

        let coors = await lugar.getLugarLatLng(direccion);
        let temp = await clima.getClima(coors.lat, coors.lng);

        return `El clima en ${ coors.direccion } es de ${ temp }`;

    } catch (e) {
        return `No se pudo determinar el clima en ${ direccion }`;
    }

}


getInfo(argv.direccion)
    .then(mensaje => console.log(mensaje))
    .catch(e => console.log(e));
```

2-) Del modelo de nuestra Api-Rest, aplique persistencia de los datos en un BBDD. Los datos que se quieren guardar son: localidad, temperatura, presión y humedad.

3-) Suba el repositorio a GitHub