

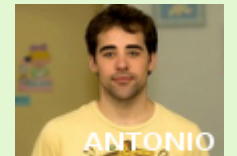
1.- Descripción de la tarea



Caso práctico

Juan recomienda a **Antonio** utilizar AJAX para darle un toque de modernidad a sus aplicaciones. Además, le comenta que existe una librería que le va a facilitar enormemente el programar con AJAX y va a conseguir dar buenos efectos y mayor dinamismo.

A **Antonio** le atrae mucho el tema y se pone manos a la hora con esta nueva sección.



¿Qué te pedimos que hagas?

El tiempo con Weather Api

En esta tarea vamos a obtener información sobre el tiempo actual, previsto y contaminación, además de usar la nueva API de geodificación.

¿Cómo lo vamos a implementar? Pues en un solo archivo html y con un sólo js. No es necesario utilizar clases.

Nos vamos a tener que registrar en WeatherAPI: <https://www.weatherapi.com/>

Lo habitual en el mundo actual es utilizar alguna API ya sea JSON, XML o incluso CSV. Esta API soporta JSON y XML.

Esta API REST te puede devolver datos como la temperatura, el viento, la humedad, previsión de lluvias, contaminación, alertas, etc... Y es gratuita con hasta 5 millones de consultas al mes.

¿Qué tenéis que hacer?

Antes de nada en el punto 2.- Información de Interés tienes una explicación de como registrarte, acceder a los datos, y más cuestiones necesarias y obligatorias para la tarea.

Se va a realizar una pequeña app donde tengas dos botones y un `input text`.

El primer botón es un botón va a mostrar la previsión del tiempo actual.

El segundo botón la previsión para los diez próximos días.

También vamos a usar la API de información de cities que tiene la API GeoDB Cities: <http://geodb-cities-api.wirefreethought.com/>

Veamos que se pide exactamente:

- ✓ Resumen de la **predicción para los próximos 3 días**, a las 14:00 y a las 5:00. Han actualizado las condiciones de la API y la demo no dura un mes, así que no se piden 9 días.
- ✓ El día de hoy debe quedar más destacado respecto a los otros días. Debe realizarse con otra consulta diferente. Debe estar limitada al día de hoy. No me vale que hagas una sola consulta para las dos acciones (tiempo actual y previsión).
- ✓ Para el día el actual (No se hace una consulta a la previsión, si no al día actual como se explican en el punto 2):
 - ➡ Temperatura **actual**.

- Viento y dirección. Si no hay viento este dato no es visible.
 - Estado del tiempo e Icono con la previsión. O sea que debe aparecer el típico icono que llueve, o está soleado o ventoso, etc.
 - Llueve o no con su tipo de precipitación. Si no hay lluvia este dato no es visible.
 - Calidad del Aire. Muestre varios tipos de partículas. Puede elegir a parte de las obligatorias: co y no2.
- ✔ Para el resto de los días:
 - Temperaturas.
 - Iconos.
 - Si hay alerta o no.
 - Puesta y salida del sol.
- ✔ Si es una ciudad o pueblo lo que se solicita se va a mostrar:
 - Población.
 - País. En español.
 - Región. En español.
- ✔ Vamos a usar la **longitud** y **latitud** que nos ofrece cada consulta para **mostrar en un mapa** donde se encuentra el lugar que hemos escrito. Para ello nada más sencillo que usar openStreetMaps. Este mapa estará dentro de un *iframe*. *Tienes dos botones para el tiempo, pero siempre muestras el mismo mapa.*
- ✔ Si pulsamos un tercer botón vamos a usar la longitud y latitud que Que nos ofrece el objeto ***geolocation***. Para ello nada más sencillo que usar openStreetMaps. Este mapa estará dentro de un iframe. Es bastante fácil.
- ✔ Nota extra (no puedes superar el 10 pero si compensar algo): usar leaflet en vez de openstreetmap directamente y que cuando se pulse una posición en el mapa se muestre la previsión del día de hoy.

Observaciones:

- **Las consultas se deben realizar en unidades métricas.**
- **jQuery es necesario para la práctica.** Sobre todo para la visualización. **Una de las consultas será con API fetch**, la otra con \$.ajax de jQuery o Axios, pero no aseguro que este último vaya a estar nombrado en el examen.
- Al utilizar mapas vas a tener que usar **async await**. Sin usarlo no se actualizará algunas veces el contenido.
- Para obtener el valor de un input del formulario con id="localidad" se puede utilizar, por ejemplo, \$("#localidad").val() que te devuelve el valor actual (no el predefinido) de ese campo.
- Si creas una función para gestionar un evento en jQuery puede llevar un parámetro evento (es opcional) pero muy interesante para abortar la propagación del evento. Podrías usar por ejemplo:

```
$("#button").click(function (evt) {
  //// Código
  evt.preventDefault();
});
```

- Para cambiar el css con jQuery podéis utilizar:

```
$('#elemento').css('propiedad-css', 'valor a asignar');
```

2.- Información de interés.

Recursos necesarios y recomendaciones

- ✓ Ordenador personal.
- ✓ Diferentes navegadores web para ejecutar y probar su funcionamiento y compatibilidad cross-browser.
- ✓ Cuando subáis procurar no superéis el tamaño de la entrega. Por favor. Escalar las imágenes de si es necesario.
- ✓ Editor web para teclear el código de la aplicación.
- ✓ Iconos de condiciones climáticas. <https://openweathermap.org/weather-conditions> Puedes coger de ahí los enlaces a cada imagenm.
- ✓ API geodb-cities. <http://geodb-cities-api.wirefreethought.com/>

Y además os ayudo un poco para que entendáis las estructuras que se necesitan.

Aquí quiero ayudaros a comprender un poco más como funciona esta API Rest que es fácil. Cómo está en inglés os recomiendo un diccionario y que os leáis esta página antes de entrar en las de documentación: <https://www.weatherapi.com/docs/> Además también tenéis un simulador: <https://www.weatherapi.com/api-explorer.aspx> de funcionamiento de la API.

Si el inglés es un problema, puedes usar deepl <https://www.deepl.com/translator> es un traductor de inglés bastante bueno.

Lo primero que tenéis que hacer es registraros aquí: <https://www.weatherapi.com/signup.aspx>

Al registrarte tendrás podrás crear una APP, y dentro tendrás una key o clave con al que podrás realizar consultas. Es un número único y personal que deberéis meter en las consultas. Lo puedes encontrar ahí <https://www.weatherapi.com/signup.aspx> en la parte superior a la derecha del menú lateral izquierdo.

Asegúrate de usar

No creo que tengas problemas con el límite de consultas ya que dicen que tenemos 5 millones. Lo que no tenemos es acceso a las APIs de pago.

A todas las consultas posteriores debéis añadir **key=tuclave** , siendo tuclave el número que os dan al registraros y que podéis ver en vuestro perfil.

```
const TUCLAVE=000000000000; // En vez de los ceros pon la clave que te han dado al registrarte. // Además es un ejemplo perfecto de uso de const adecuado.
```

En el input de entrada vamos a tener una cadena de texto. Si no contiene ninguna "," coma, pues entonces vamos a suponer que la localización está en España (habría que añadir ,es).

Veamos por ejemplo del tiempo actual en Granada. He hecho una petición de varios días.

✓ <https://api.weatherapi.com/v1/current.json?key=TUCLAVE&q=Granada ES&aqi=yes>

Normalmente basta con poner es Granada sin el espacio ES. Pero por si acaso tenlo en cuenta.

```
{
  "location": {
    "name": "Granada",
    "region": "Andalucia",
    "country": "Spain",
    "lat": 37.1833,
    "lon": -3.6,
    "tz_id": "Europe/Madrid",
    "localtime_epoch": 1743674157,
```

```

    "localtime": "2025-04-03 11:55"
  },
  "current": {
    "last_updated_epoch": 1743673500,
    "last_updated": "2025-04-03 11:45",
    "temp_c": 13.1,
    "temp_f": 55.6,
    "is_day": 1,
    "condition": {
      "text": "Partly cloudy",
      "icon": "//cdn.weatherapi.com/weather/64x64/day/116.png",
      "code": 1003
    },
    "wind_mph": 2.9,
    "wind_kph": 4.7,
    "wind_degree": 133,
    "wind_dir": "SE",
    "pressure_mb": 1011.0,
    "pressure_in": 29.85,
    "precip_mm": 0.0,
    "precip_in": 0.0,
    "humidity": 94,
    "cloud": 75,
    "feelslike_c": 13.4,
    "feelslike_f": 56.0,
    "windchill_c": 15.9,
    "windchill_f": 60.6,
    "heatindex_c": 15.9,
    "heatindex_f": 60.6,
    "dewpoint_c": 6.1,
    "dewpoint_f": 43.0,
    "vis_km": 9.0,
    "vis_miles": 5.0,
    "uv": 1.3,
    "gust_mph": 3.4,
    "gust_kph": 5.4,
    "air_quality": {
      "co": 233.1,
      "no2": 18.5,
      "o3": 59.0,
      "so2": 1.295,
      "pm2_5": 8.695,
      "pm10": 11.285,
      "us-epa-index": 1,
      "gb-defra-index": 1
    }
  }
}

```

La primera posición del array resultante es la que vamos a usar en todos los casos. Si por casualidad no sale en primera posición, no os preocupéis. Indicáis usando country y state cual es el lugar. Recordad que al poner *"localización,es"* nos aseguramos que sea en España. Como ejercicio para repasar para el examen podéis intentar recorrer el array entero y mostrar la información completa y todos los lugares que se llaman como esa localización.

Para finalizar un pequeño ejemplo de uso. Supongamos que datos_devueltos es el archivo JSON leído desde el servido y #cambia lo que deseamos cambiar:

```
$("#cambia").append(datos_devueltos.location.country +" nos devuelve Spain " + datos_devueltos.current.humidity +" Nos devuelve la humedad.<br>"); // Te devuelve el nombre y presión.
```

Recordar que no vale usar la consulta de previsión de forecast para el día de hoy. Tenéis que hacer dos consultas, una para el tiempo actual y otra para el tiempo futuro.

✓ <https://api.weatherapi.com/v1/forecast.json?key=TUCLAVE&q=Granada ES&days=1&aqi=yes&alerts=yes>

```
{
  "location": {
    "name": "Granada",
    "region": "Andalucia",
    "country": "Spain",
    "lat": 37.1833,
    "lon": -3.6,
    "tz_id": "Europe/Madrid",
    "localtime_epoch": 1743674500,
    "localtime": "2025-04-03 12:01"
  },
  "current": {
    "last_updated_epoch": 1743674400,
    "last_updated": "2025-04-03 12:00",
    "temp_c": 12.1,
    "temp_f": 53.8,
    "is_day": 1,
    "condition": {
      "text": "Overcast",
      "icon": "//cdn.weatherapi.com/weather/64x64/day/122.png",
      "code": 1009
    },
    "wind_mph": 2.2,
    "wind_kph": 3.6,
    "wind_degree": 205,
    "wind_dir": "SSW",
    "pressure_mb": 1011.0,
    "pressure_in": 29.85,
    "precip_mm": 0.0,
    "precip_in": 0.0,
    "humidity": 100,
    "cloud": 100,
    "feelslike_c": 12.6,
    "feelslike_f": 54.6,
    "windchill_c": 17.3,
    "windchill_f": 63.1,
    "heatindex_c": 17.3,
    "heatindex_f": 63.1,
    "dewpoint_c": 6.4,
    "dewpoint_f": 43.5,
    "vis_km": 10.0,
    "vis_miles": 6.0,
    "uv": 1.7,
    "gust_mph": 2.2,
    "gust_kph": 3.6,
    "air_quality": {
      "co": 223.85,
```

```
    "no2": 14.43,
    "o3": 65.0,
    "so2": 1.48,
    "pm2_5": 7.585,
    "pm10": 10.36,
    "us-epa-index": 1,
    "gb-defra-index": 1
  }
},
"forecast": {
  "forecastday": [
    {
      "date": "2025-04-03",
      "date_epoch": 1743638400,
      "day": {
        "maxtemp_c": 15.7,
        "maxtemp_f": 60.2,
        "mintemp_c": 7.8,
        "mintemp_f": 46.0,
        "avgtemp_c": 11.5,
        "avgtemp_f": 52.6,
        "maxwind_mph": 5.6,
        "maxwind_kph": 9.0,
        "totalprecip_mm": 1.01,
        "totalprecip_in": 0.04,
        "totalsnow_cm": 0.0,
        "avgvis_km": 9.5,
        "avgvis_miles": 5.0,
        "avghumidity": 78,
        "daily_will_it_rain": 1,
        "daily_chance_of_rain": 88,
        "daily_will_it_snow": 0,
        "daily_chance_of_snow": 0,
        "condition": {
          "text": "Patchy rain nearby",
          "icon": "//cdn.weatherapi.com/weather/64x64/day/176.png",
          "code": 1063
        },
        "uv": 0.7,
        "air_quality": {
          "co": 226.14400000000006,
          "no2": 10.737400000000001,
          "o3": 65.08,
          "so2": 1.3098,
          "pm2_5": 9.9308,
          "pm10": 17.967200000000002,
          "us-epa-index": 1,
          "gb-defra-index": 1
        }
      },
      "astro": {
        "sunrise": "07:57 AM",
        "sunset": "08:39 PM",
        "moonrise": "11:15 AM",
        "moonset": "02:02 AM",
        "moon_phase": "Waxing Crescent",
        "moon_illumination": 27,
        "is_moon_up": 1,
```

```

        "is_sun_up": 0
    },
    "hour": [
        {
            "time_epoch": 1743631200,
            "time": "2025-04-03 00:00",
            "temp_c": 12.4,
            "temp_f": 54.4,
            "is_day": 0,
            "condition": {
                "text": "Partly Cloudy ",
                "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png",
                "code": 1003
            }
        }, .....
    ]
}

```

Para la API GeoDB Cities no es necesario que nos registremos.

El tipo de consulta será GET, como en el anterior caso.

La dirección desde la que vamos a partir de la siguiente dirección.

http://geodb-free-service.wirefreethought.com/v1/geo/places

Quedando así la consulta de la ciudad de Granada.

<http://geodb-free-service.wirefreethought.com/v1/geo/places?limit=5&offset=0&types=CITY&namePrefix=Granada&languageCode=es>

Tenemos que tener en cuenta que esta API está saturada muchas veces y tienes que tener en cuenta los posibles errores.

```

{
  "data": [
    {
      "id": 3229065,
      "wikiDataId": "Q2101383",
      "type": "CITY",
      "name": "Granada",
      "country": "Colombia",
      "countryCode": "CO",
      "region": "Meta",
      "regionCode": "MET",
      "regionWdId": "Q238629",
      "latitude": 3.54722222,
      "longitude": -73.70861111,
      "population": 56065,
      "distance": null,
      "placeType": "CITY"
    },
    {
      "id": 27978,
      "wikiDataId": "Q8810",
      "type": "CITY",
      "name": "Granada",
      "country": "España",
      "countryCode": "ES",

```

```
    "region": "Andalucía",
    "regionCode": "AN",
    "regionWdId": "Q5783",
    "latitude": 37.17805555,
    "longitude": -3.60083333,
    "population": 230595,
    "distance": null,
    "placeType": "CITY"
  },
  {
    "id": 3557462,
    "wikiDataId": "Q31916723",
    "type": "CITY",
    "name": "Ciudad de Granada",
    "country": "España",
    "countryCode": "ES",
    "region": "Andalucía",
    "regionCode": "AN",
    "regionWdId": "Q5783",
    "latitude": 37.18817,
    "longitude": -3.60667,
    "population": 234325,
    "distance": null,
    "placeType": "CITY"
  },
  {
    "id": 83019,
    "wikiDataId": "Q205648",
    "type": "CITY",
    "name": "Granada",
    "country": "Nicaragua",
    "countryCode": "NI",
    "region": "Departamento de Granada",
    "regionCode": "GR",
    "regionWdId": "Q258405",
    "latitude": 11.93035,
    "longitude": -85.953591666,
    "population": 105171,
    "distance": null,
    "placeType": "CITY"
  },
  {
    "id": 83019,
    "wikiDataId": "Q205648",
    "type": "CITY",
    "name": "Granada",
    "country": "Nicaragua",
    "countryCode": "NI",
    "region": "Departamento de Granada",
    "regionCode": "GR",
    "regionWdId": "Q258405",
    "latitude": 11.93472222,
    "longitude": -85.95888888,
    "population": 105171,
    "distance": null,
    "placeType": "CITY"
  }
],
```



```
"metadata":{  
  "currentOffset":0,  
  "totalCount":5  
}  
}
```

Si os fijáis podéis saber el total de países al final dentro de metadata. O podéis comprobar la longitud con length, o usar un bucle for in o for of y dentro ya podéis acceder a los datos. Os recomiendo el for in para que practiquéis.

Ahora bien, si no habéis puesto el país deberéis asegurar que el lugar que buscamos es ES comparando con "countryCode":"ES" . Como el país es la parte final de la cadena de entrada después de una "," coma, deberéis sacar esa parte y compararlo. De esa forma habréis encontrado la ciudad con la población solicitada.

Finalmente sobre el tiempo voy a mostraros un ejemplo de como se vería el mapa de la consulta de Granada aproximadamente (aunque puede que no se vea debido a restricciones de exeLearning).

[Ver el mapa más grande](#)

Si no lo ves bien abre en enlace y a la derecha tendrás un botón de compartir donde te genera un iframe. Con esa base debes pensar como adaptarlo a los datos.

La otra opción del mapa es hacerlo programando como se indica en el punto 1.



Indicaciones de entrega

Una vez realizada la tarea, el envío se realizará a través de la plataforma. Comprime la carpeta con el proyecto en un fichero .zip y nómbralo siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre_DWEC_Tarea06