





**API**  
**PUEBLITOS DE**


**ESPAÑA**



## Api Elegida



**GeoDB Cities API Documentation (wirefreethought)**  
Get global city, region, and country data. Filter and display results in multiple languages.

 RapidAPI

<https://rapidapi.com/wirefreethought/api/geodb-cities/>

# LLamada a la API



- Declaro 2 variables que utilizaré a lo largo del programa.
- En el HTML tenemos una caja de búsqueda con un evento que se dispara al cambiar el contenido.
- Recogemos el valor de búsqueda y lo usamos para petición a la API.
- Parseamos el resultado en formato JSON y lo procesamos

2

```
let datosResultado;  
let indice;  
  
function sugerencias() {  
  
    var busqueda = document.getElementById("busqueda").value;  
  
    //llamar API.  
  
    //Codigo generado por la página https://rapidapi.com/wirefreethought/api/geodb-cities/  
  
    const data = null;  
  
    const xhttp = new XMLHttpRequest();  
    xhttp.withCredentials = true;  
  
    // listener para ejecutar código cuando se completa la llamada a la API  
    xhttp.addEventListener("readystatechange", function () {  
        if (this.readyState === 4 && this.status === 200) {  
            //procesar la respuesta de la API  
  
            // el resultado esta en la propiedad "responseText"  
            var resultadoJSON = JSON.parse(xhttp.responseText);  
            procesarResultado(resultadoJSON);  
        }  
    });  
  
    xhttp.open("GET", "https://wft-geo-db.p.rapidapi.com/v1/geo/cities?countryIds=ES&namePrefix=" + busqueda);  
    xhttp.setRequestHeader("X-RapidAPI-Key", "fb87d7a3b9mshdc50827050564a7p168cfbjsne0473be5d1d7");  
    xhttp.setRequestHeader("X-RapidAPI-Host", "wft-geo-db.p.rapidapi.com");  
  
    xhttp.send(data);  
  
}
```

# Función para procesar el resultado

- Como parámetro de entrada, le pasamos el resultado del JSON

- Recogemos el elemento <ul> a través de su id, que inicialmente está oculto, y por css lo mostramos en la página.

- Del JSON de entrada, extraemos "data" que contiene un array con los datos que necesito

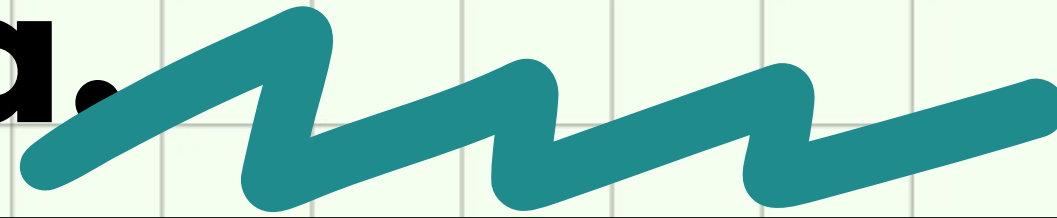
- Recorro el array "data" (datosResultado), y por cada "dato" genero un elemento de lista con el "name" de dato (que contiene el nombre del pueblo)

- Además, a cada item de lista le añado el evento "onclick", y aprovecho el contador del bucle para que sea su parámetro de entrada (y así diferenciar que fila es la que se pulsa)

```
function procesarResultado(resultadoJSON) {  
  // variable para el contenido html de la lista de sugerencias (ul)  
  let lihtml = ""  
  
  const listaSugerencias = document.getElementById("listaSugerencia");  
  listaSugerencias.style.display="block";  
  
  // en el campo data del JSON estan los datos de la lista  
  datosResultado = resultadoJSON["data"];  
  
  // recorrer para generar los items de la lista  
  for (let i = 0; i < datosResultado.length; i++) {  
    const dato = datosResultado[i];  
  
    // para diferenciar el elemento de listas que se pulsa, al onclick le pasamos el indice (i) como parametro  
    lihtml = lihtml + "<li onclick='opcionLista(" + i + ")'>" + dato["name"] + "</li>"  
  }  
  
  // insertar el html de la lista  
  listaSugerencias.innerHTML = lihtml;  
}
```

- Por último, la variable donde he ido concatenando los items será el contenido de la lista, para que se muestren los resultados en la página.

# Funcion escoger de la lista.



```
function opcionLista(opcion) {  
  indice = opcion;  
  
  // obtenemos la opcion de la lista que ha seleccion el usuario haciendo click  
  const dato = datosResultado[indice];  
  
  // rellenamos la caja de busqueda con el nombre del pueblo y ocultamos la lista de sugerencias  
  document.getElementById("busqueda").value = dato["name"];  
  document.getElementById("listaSugerencia").style.display = "none";  
}
```

● El parámetro de entrada (opcion), como he explicado antes, se debe corresponder con el índice en mi array de datos (el contador en el bucle)

2 ● A partir del "indice" obtenemos el dato correspondiente (objeto JSON con toda la información del pueblo/cuidad)

● Extraemos el nombre ("name") de dato JSON y lo escribimos en la caja de búsqueda

● Como ya se ha seleccionado un dato, ocultamos la lista de sugerencias

# Función buscar pueblos.

De la petición a la API, tenemos guardado el array con los resultados en una variable.

De la selección de sugerencias, tenemos guardado el índice de la lista seleccionado por el usuario en otra variable.

Luego, el dato seleccionado será el que esté en la posición que indica el "índice"

Cogemos el dato en dicha posición (un objeto JSON), y de él extraemos la información que necesitamos

- Para mostrar en pantalla, necesitamos la provincia y la población.

- Para mostrar en Google maps, necesitamos la latitud y la longitud (para pasárselo al método).

Como ya tenemos todo lo que necesitamos, mostramos el <div> que tiene los resultados

```
let map;

function buscarPueblo() {
  const dato = datosResultado[indice];

  const latitud = dato["latitude"];
  const longitud = dato["longitude"];

  const provincia = dato["region"];
  const poblacion = dato["population"];

  document.getElementById("provincia").innerHTML=provincia;
  document.getElementById("poblacion").innerHTML=poblacion;

  initMap(latitud, longitud);

  document.getElementById("resultado").style.display = "block";
}
```



# Api utilizada para crear el mapa



# Funcion mapa.



```
function initMap(latitud, longitud) {  
  map = new google.maps.Map(document.getElementById("map"), {  
    center: { lat: latitud, lng: longitud },  
    zoom: 9,  
  });  
}
```

Según el ejemplo que proporciona la API, para pintar un mapa hacen falta su posición global, es decir, latitud y longitud

Entonces, como van a depender del pueblo en sí, lo que hacemos es pasar dichos datos como parametros (desde fuera) al método encargado de pintar el mapa



# Resultado.



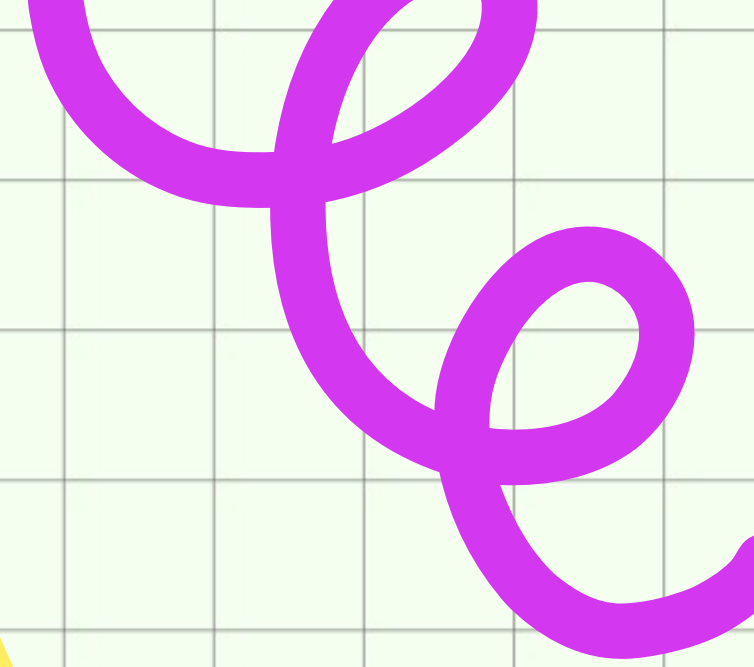
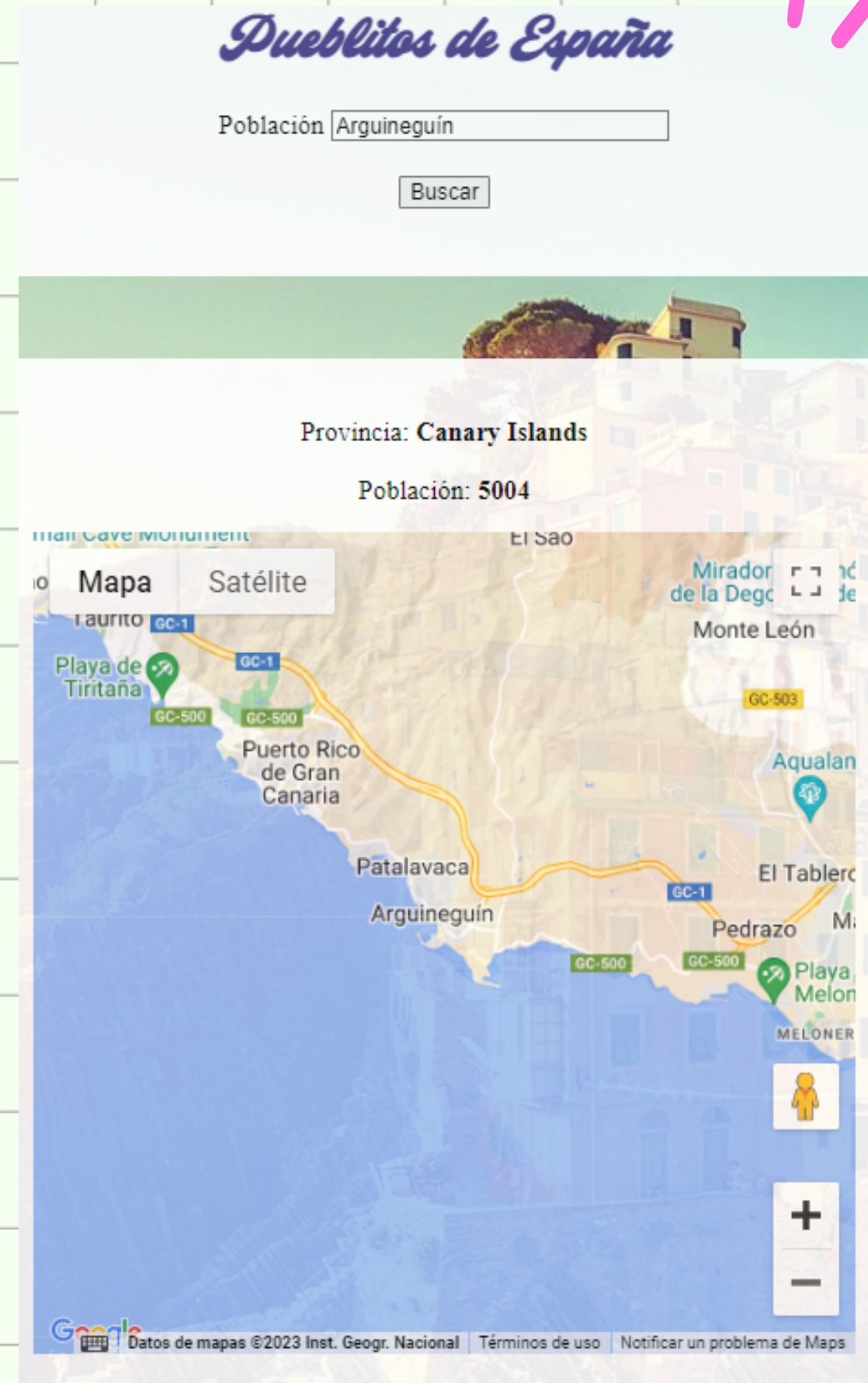
*Dueblitos de España*

Población

- Argual
- Arguedas
- Arguineguín
- Arguis
- Arguisuelas

Buscar

# Resultado



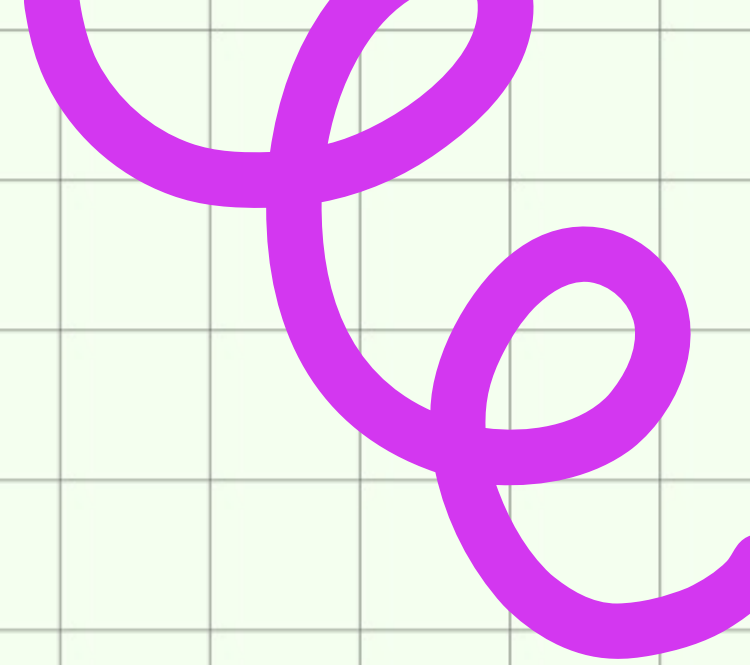


**PUEBLITOS DE ESPAÑA**

**JQUERY**



# JQuery



**1** Lo primero que hice fue hacer una copia del mismo proyecto (ficheros html y js).

**2** Reescribimos el código JS, pero usando JQuery (algunos ejemplos):

```
var busqueda = $("#busqueda").val();
```

```
$("#provincia").html(provincia);  
$("#poblacion").html(poblacion);
```

```
$("#resultado").css("display", "block");
```



# Llamada a la api usando JQuery

```
function sugerencias() {  
  
    var busqueda = $("#busqueda").val();  
  
    //llamar API.  
  
    //Codigo generado por la página https://rapidapi.com/wirefreethought/api/geodb-cities/  
    const settings = {  
        "async": true,  
        "crossDomain": true,  
        "url": "https://wft-geo-db.p.rapidapi.com/v1/geo/cities?countryIds=ES&namePrefix=" + busqueda ,  
        "method": "GET",  
        "headers": {  
            "X-RapidAPI-Key": "fb87d7a3b9mshdc50827050564a7p168cfbjsne0473be5d1d7",  
            "X-RapidAPI-Host": "wft-geo-db.p.rapidapi.com"  
        }  
    };  
  
    $.ajax(settings).done(function (response) {  
        // el resultado esta en "response" y ya en formato JSON  
        procesarResultado(response);  
    });  
}
```