TAREA OPCIONAL 2

INTRODUCCION

Usaremos dos servicios externos. El primer servicio será un servicio de geolocalización que, dada una ip, obtenga los datos de esa ip. Para nuestro proyecto hemos elegido la API **ipstack** (https://ipstack.com/documentation). Este servicio nos proporcionará, entre otros, los datos de longitud y latitud, los cuales usaremos para mostrar la ubicación del cliente en **Google Maps** (https://developers.google.com/maps/documentation/javascript/adding-a-google-map#maps_add_map-javascript).

Además, el enunciado dice que, cito textualmente *"Se valorará positivamente el uso del formato **JSON** como intercambio de datos. También se valorarán diferentes formas de acceso al servicio, por ejemplo, desde un **script del cliente** y/o desde una aplicación **PHP**."*. Le haremos caso y haremos que la geolocalización se obtenga mediante **PHP** devolviendo un **json** y mostrar la ubicación en Maps se realizará mediante un **script del cliente**.

OBTENER GEOLOCALIZACIÓN (IPSTACK): GETLOCATION.PHP

La API que hemos elegido, tiene dos modos de llamada. La primera es dando una dirección ip añadida en la url (http://api.ipstack.com/132.12.154.125?access_key=MI_KEY) o haciendo que obtenga la ip de donde se está haciendo la llamada (http://api.ipstack.com/check?access_key=MI_KEY). Usaremos la primera forma para obtener la ip del cliente y la segunda forma para obtener la ip del servidor.

Es diferente si se está trabajando en localhost o en la nube, ya que, en local, no se genera la variable global php $_SERVER['HTTP_X_REAL_IP']; que es de donde sacaremos la ip del cliente.

Preparamos las variables $ip, para el cliente y $ip2 para el servidor, desde *servicesConfig.php*:

```php
$estado = 0; //1 para la nube
if ($estado ==0){
    $verifypass='http://localhost/LABS/Proyecto1SW/php/VerifyPassWS.php';
    $getQuestion='http://localhost/LABS/Proyecto1SW/php/GetQuestionWS.php';
    $ip = 'check';
    $ip2 = 'check';
}else{
    $verifypass='https://swmikel-iturria.000webhostapp.com/Proyecto1SW/php/VerifyPassWS.php';
    $getQuestion='https://swmikel-iturria.000webhostapp.com/Proyecto1SW/php/GetQuestionWS.php';
    $ip = $_SERVER['HTTP_X_REAL_IP'];
    $ip2 = 'check';
```

Para hacer la llamada, la API nos indica que tendremos que hacerlo mediante **CURL**. Creamos ahora la conexión que haremos para obtener la localización del cliente. Para ello, le pasamos la ip del cliente en el parámetro $ip que ya tenemos inicializada en servicesConfig.php:

```php
$access_key = '8249e3ea791e655e29fb052c47a228e8';
include "servicesConfig.php";

//PARA LOS DATOS DEL CLIENTE
// Initialize CURL:
$ch = curl_init('http://api.ipstack.com/'.$ip.'?access_key='.$access_key.'');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

Ejecutamos el CURL con curl_exec y guardamos el **JSON** de respuesta en una variable a la que llamaremos $json y después cerramos la conexión.

```
// Store the data:
$json = curl_exec($ch);
curl_close($ch);
```

Una vez tenemos la respuesta, decodificamos la respuesta de JSON que nos ha mandado el servicio web y lo guardamos en $api_result:

```
// Decode JSON response:
$api_result = json_decode($json, true);
```

Esto sería lo que devuelve el JSON:


```
{"ip":"80.26.201.84","type":"ipv4","continent_code":"EU","continent_name":"Europe","country_code":"ES",
"country_name":"Spain","region_code":"PV","region_name":"Basque Country","city":"Donostia \/ San
Sebasti\u00e1n","zip":"20001","latitude":43.32276916503906,"longitude":-1.9728599786758423,"location":
{"geoname_id":3110044,"capital":"Madrid","languages":
[{"code":"es","name":"Spanish","native":"Espa\u00f1ol"},
{"code":"eu","name":"Basque","native":"Euskara"},{"code":"ca","name":"Catalan","native":"Catal\u00e0"},
{"code":"gl","name":"Galician","native":"Galego"},
{"code":"oc","name":"Occitan","native":"Occitan"}],"country_flag":"http:\/\/assets.ipstack.com\/flags\/
es.svg","country_flag_emoji":"\ud83c\uddea\ud83c\uddf8","country_flag_emoji_unicode":"U+1F1EA
U+1F1F8","calling_code":"34","is_eu":true}}
```


De ahí nos interesaré quedarnos con “ip”, “country_name”, “region_name”, “city”, “zip”, “latitude” y “longitude”, el resto, aunque son interesantes (por ejemplo, capital del país, idiomas que se hablan…) no los usaremos.

Una vez está decodificado, podremos leer los resultados, con lo que leemos los resultados que nos interesan y los guardamos en variables para facilitar después la impresión:

```
$ip =$api_result['ip'];
$pais =$api_result['country_name'];
$region =$api_result['region_name'];
$ciudad =$api_result['city'];
$cp=$api_result['zip'];
$latitud=$api_result['latitude'];
$longitud=$api_result['longitude'];
```

Importante ver como hemos guardado la latitud y la longitud, ya que luego los usaremos para mostrar la ubicación en Maps.

Hacemos exactamente lo mismo para obtener los datos del servidor (de hecho, hemos hecho copiar-pegar y añadido a todas las variables un 2, recordemos que $ip2 ya está definida en *servicesConfig.php*).

Ahora, preparamos los datos para su impresión. Sera imprimido en una tabla básica:

```
echo "<table border=1>";
echo "<tr><th></th><th>Cliente</th><th>Servidor</th></tr>";
echo "<tr><td>Dirección ip:</td>";
echo "<td>$ip</td>";
echo "<td>$ip2</td></tr>";
echo "<tr><td>Pais:</td>";
echo "<td>$pais</td>";
echo "<td>$pais2</td></tr>";
echo "<tr><td>Region:</td>";
echo "<td>$region</td>";
echo "<td>$region2</td></tr>";
echo "<tr><td>Ciudad:</td>";
echo "<td>$ciudad</td>";
echo "<td>$ciudad2</td></tr>";
echo "<tr><td>Codigo Postal:</td>";
echo "<td>$cp</td>";
echo "<td>$cp2</td></tr>";
echo "</table>";
```

Ya está definido el cliente para obtener la localización por ip. Procedemos ahora con la segunda parte.

CREDITS.PHP

Tal y como decía el enunciado, se valorará hacer las llamadas de dos formas, uno mediante php y la otra mediante un script en el cliente. Como la parte de la geolocalización la hicimos mediante **php**, mostrar la ubicación en **Maps** la haremos con un **script en el cliente**.

Primero, añadiremos el script de la API que vamos a usar:

```
<script
  src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBvzzWoLs2-LJiFqqsXRnKp7t1eFpeCw5E&callback=initMap&libraries=&v=weekly"
  defer
></script>
```

Una vez tenemos la API, creamos el estilo que tendrá el mapa, en nuestro caso, no queremos que sea muy grande asique fijamos el alto a 400 pixeles, lo centramos poniendo margin:auto, hacemos que el ancho sea el 60% de la página y le añadimos un borde azul para seguir con el estilo de la página:

```
<style type="text/css">
  #map {
    height: 400px;
    margin: auto;
    width: 60%;
    border-style: solid;
    border-color: blue;
  }
</style>
```

Antes de proceder a crear el mapa, incluimos el php de la geolocalización. Recuerda que los valores de la **latitud** y la **longitud** están en variables de ese fichero con lo que hacemos el include:

```
<div style="margin:auto; width:60%;">
<?php include "getLocation.php";?>
</div>
```

Ahora, procedemos a crear la función de JavaScript para inicializar el mapa a la que llamaremos initMap. Crearemos una constante “pos” a la que le añadiremos los valores de las variables de latitud y longitud que habíamos obtenido en *getLocation.php*. Después, usando las funciones de la API google.maps.Map y google.maps.Marker, creamos el mapa y el marker (el icono rojo que señala un punto en el mapa), al que le daremos como posición la variable que hemos creado con la longitud y latitud del cliente.

```
<script>
  function initMap() {
    const pos = { lat: <?php echo "$latitud";?>, lng: <?php echo "$longitud";?> };
    const map = new google.maps.Map(document.getElementById("map"), {
      zoom: 4,
      center: pos,
    });
    const marker = new google.maps.Marker({
      position: pos,
      map: map,
    });
  }
</script>
```

Creamos el div donde será mostrado el mapa:

```
<div id="map"></div>
```

Procedemos a probarlo.

PRUEBAS

En localhost:

Tus datos de conexión son:

	Cliente	Servidor
Dirección ip:	80.26.201.84	80.26.201.84
Pais:	Spain	Spain
Region:	Basque Country	Basque Country
Ciudad:	Donostia / San Sebastián	Donostia / San Sebastián
Codigo Postal:	20001	20001

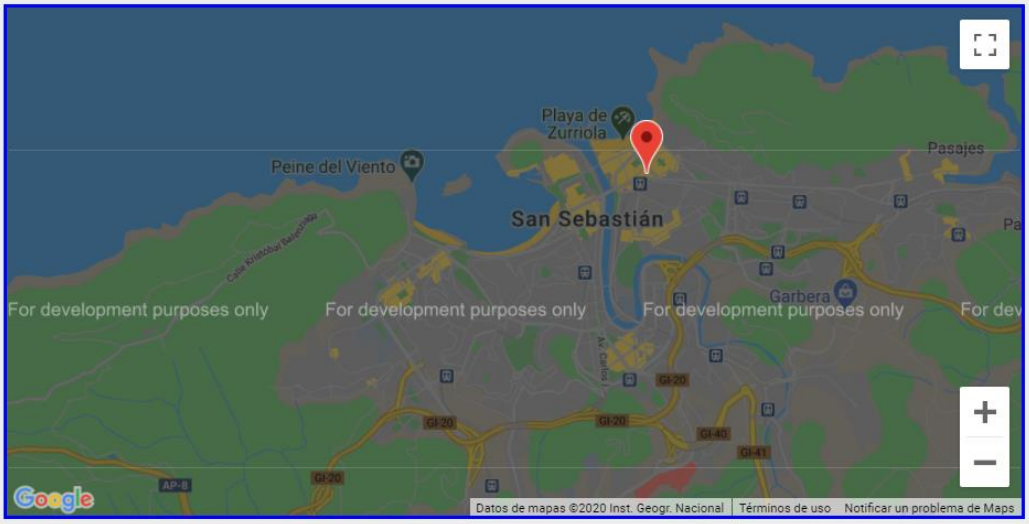
La dirección del cliente y del servidor es la misma, ya que trabajamos en local. Comprobamos la dirección IP desde la que se ha hecho la llamada desde una página que muestra las ip (https://www.cual-es-mi-ip.net/):

Tu dirección IP es 80.26.201.84 Geolocalizar IP

Proveedor de Internet	Pais	Proxy
Telefonica de Espana	Spain	no

Efectivamente, la dirección IP es la correcta. Miramos si el mapa apunta bien en donde nos encontramos:

Te encuentras en:



Apunta en San Sebastián correctamente.

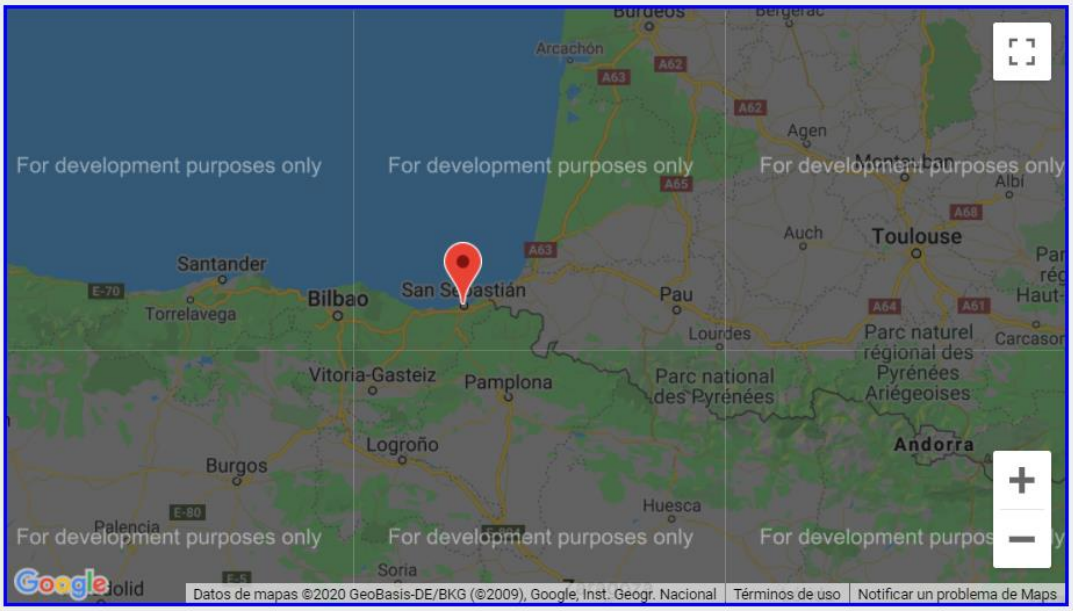
En webhost:

Tus datos de conexión son:

	Cliente	Servidor
Dirección ip:	80.26.201.84	2a02:4780:bad:1:fced:1ff:fe01:854
Pais:	Spain	Lithuania
Region:	Basque Country	Vilnius
Ciudad:	Donostia / San Sebastián	Vilnius
Codigo Postal:	20001	01100

Como se puede observar, ahora la ip del servidor y la del cliente es distinta. La ip del cliente es la misma que teníamos cuando hemos probado en localhost (y recordemos que es la misma ip que nos mostraba la web https://www.cual-es-mi-ip.net/). Además, el mapa funciona y se muestra correctamente y el puntero apunta exactamente a la posición de la IP:

Te encuentras en:



Se puede decir que todo funciona correctamente.