Tarea Opcional 2

Contenido

Enunciado

Utilizando servicios web de geolocalización o los que consideréis oportunos, modificar la página de créditos para que muestre información de la ubicación del cliente, la ciudad desde la que accede, ubicación del servidor, y cualquier otra información que se considere relevante. 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.

Modificaciones en Credits.php

Las únicas modificaciones que se han hecho en este fichero han sido, añadir las etiquetas *div* en las que introduciremos los resultados, y la adición de los scripts para la correcta ejecución.

```
5       <script src="../js/jquery-3.4.1.min.js"></script>
6       <script src="https://maps.googleapis.com/maps/api/js"></script>
7       <script src="../js/LocateUser.js"></script>
```

```
29      <h2>Tu ubicación, sientete vigilado:</h2>
30      <a href="http://www.geoplugin.com/geolocation/" target="_new">IP Geolocation</a> by <a href="http://www.geoplugin.com/" target="_new">geoPlugin</a>
31    <div id="map_canvas" style="width:50%; height:100%; margin-right:auto; margin-left:auto"></div>
32      <div id="location-info"></div>
33      <div id="weather-info"></div>
```

Como vemos, añadido el api de Google Maps, la cual nos permite añadir un mapa con un ping en la ubicación del cliente, de manera que sea más cómodo para el usuario.

Ficheros Creados

ClientGetUserLocation.php

Este primer fichero se encarga de llamar al servicio de geolocalización. Para este apartado hemos utilizado el servicio GeoPlugin, que nos permite, a través de la IP del usuario, conocer su ubicación actual. Los datos sobre la ubicación del cliente se reciben en formato JSON, que se tratan desde el fichero LocateUser.js. La implementación es la siguiente:

```
1   <?php
2
3       $json = file_get_contents('http://www.geoplugin.net/json.gp?ip=**.***.***.**');
4       //$json = file_get_contents('http://www.geoplugin.net/json.gp?ip='.$_SERVER['HTTP_X_REAL_IP']);
5
6       echo $json;
7   ?>
```

En caso de que estemos trabajando en Local, la línea de código que deberemos utilizar es la 3, en Web Host utilizaremos la 4. Esto se debe a que la variable *HTTP_X_REAL_IP*, si estamos trabajando en local, contiene la IP de Local Host, es decir, 127.0.0.1, por lo que no es admitida por el Servicio Web.

ClientLocationWeather.php

Este fichero es similar al anterior, se encarga de llamar al servicio que obtiene el tiempo en la localización del usuario y devuelve la información, en formato JSON, que se trata a través del fichero LocateUser.js. Para ello, se ha utilizado la api proporcionada por Open Weather Map. La implementación es la siguiente:

```php
<?php
    $apikey = "******************************";
    $json = file_get_contents("http://api.openweathermap.org/data/2.5/forecast?lat={$_GET['lat']}&lon={$_GET['lon']}&APPID={$apikey}&units=metric");
    echo $json;

?>
```

Como vemos, es necesaria una apikey, la cual es una versión gratuita, pero suficiente para el trabajo que queremos hacer. Como vemos en la imagen, se le pasan como parámetros la latitud y la longitud del usuario, así como, el tipo de unidades métricas para que la temperatura resultante, se nos devuelva en grados centígrados y no grados Kelvin.

LocateUser.js

De este documento se extraen tres funcionalidades:

Inicialización del mapa

```javascript
lat=0.0;
lng=0.0;
function initialize() {
    var latlng = new google.maps.LatLng(lat, lng);
    var myOptions = {
        zoom: 2,
        center: latlng,
        mapTypeId: google.maps.MapTypeId.SATELLITE
    };
    var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);

    var marker = new google.maps.Marker({
        position: map.getCenter(),
        map: map,
        title: 'Haz click aquí para hacer Zoom'
    });

    google.maps.event.addListener(marker, 'click', function() {
        if (map.getZoom() == 2)
            map.setZoom(20);
        else
            map.setZoom(2);
    });

    google.maps.event.addListener(map, 'click', function(e) {
        alert('Latitud: ' + e.latLng.lat() + '\nLongitud: ' + e.latLng.lng());
    });
}
```

Gracias a la documentación que ofrece Google sobre esta api podemos definir un simple mapa con un marcador en las coordenadas que nosotros fijemos, en este caso, dos variables, que se definen al inicio del documento y que se modifica su valor cuando nosotros llamamos al servicio Web de GeoPlugin.

Tratamiento JSON GeoPlugin

```
$(document).ready(function(){
    $.ajax({
        type: 'GET',
        url: '../php/ClientGetUserLocation.php',
        dataType:'json',
        async: false,
        cache: false,
        success: function(data){
            lat = data.geoplugin_latitude;
            lng = data.geoplugin_longitude;
            $("#location-info").html("<table border = \1\" style=\"margin-
            right:auto; margin-left:auto\"><tr><th>Ciudad: </th>
            <td>"+data.geoplugin_city+"</td></tr><tr><th>Region: </th>
            <td>"+data.geoplugin_regionName+"</td></tr><tr><th>Pais: </th>
            <td>"+data.geoplugin_countryName+"</td></tr></table>");

        }
    });
```

Como se ve en la imagen, a través de una petición Ajax, se solicita el fichero ClientGetUserLocation.php, y se recogen los datos del JSON obtenido. De estos datos, se han seleccionado, únicamente, la ciudad, la región y el país del cliente que se está conectando, y se han colocado en una tabla dentro del fichero Credits.php.

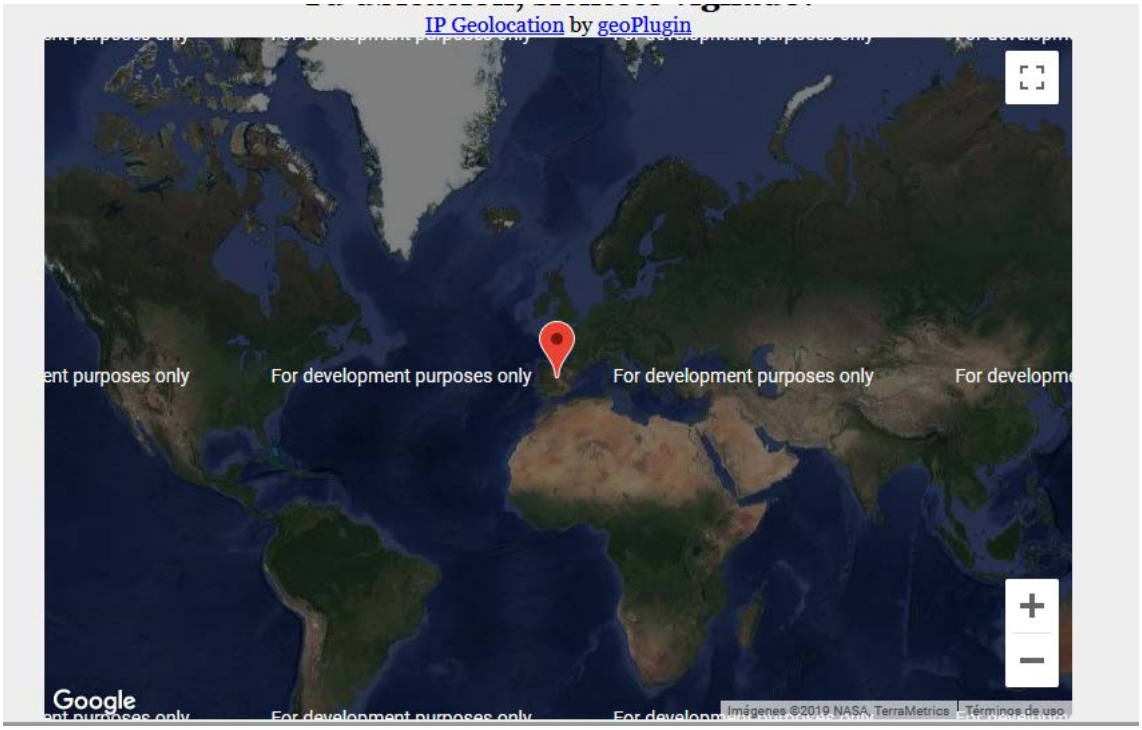
Tratamiento JSON Open Weather

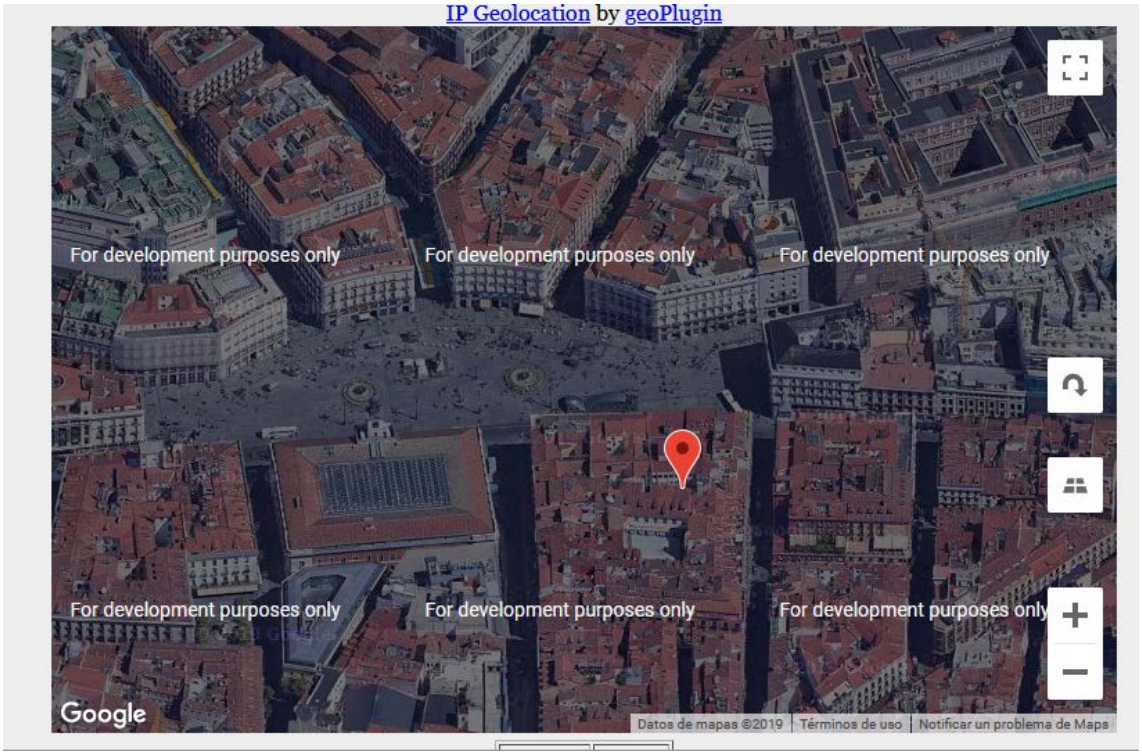
```
    $.ajax({
       type: 'GET',
       url: '../php/ClientLocationWeather.php?lat='+lat+'&lon='+lng,
       dataType:'json',
       async: false,
       cache: false,
       success: function(data){
           $("#weather-info").html("<table border = \1\" style=\"margin-right:auto;
           margin-left:auto\"><tr><th>Estado del cielo: </th>
           <td>"+data.list[0].weather[0].main+"</td></tr><tr><th>Descripcion: </th>
           <td>"+data.list[0].weather[0].description+"</td></tr><tr>
           <th>Temperatura: </th><td>"+data.list[0].main.temp+"°C</td></tr>
           </table>");

       }
    });
    initialize();
});
```

Como vemos, a través de una petición Ajax, se solicita el fichero ClientLocationWeather.php seguido de los parámetros *lat* y *lon* obtenidos gracias a la petición Ajax anterior. Cuando recibimos los datos, los introducimos en una tabla que los muestra al usuario. Los datos escogidos para esta prueba, pudiendo haber seleccionado más, han sido, el estado del cielo y una descripción del mismo, y la temperatura actual en la ubicación.

Por último, al estar ya definidas las variables se llama al método *initialize()*.

Ejecución en Web Host





Debido a la compañía telefónica desde la que se están realizando las pruebas, la ubicación actual es Madrid, ciudad en la que está ubicada la compañía.

En cuanto a los datos de las tablas:

Ciudad:	Madrid
Region:	Madrid
Pais:	Spain

Estado del cielo:	Clouds
Descripcion:	overcast clouds
Temperatura:	9.9ºC