Guía Parte 2: Las primeras funcionalidades de tu mapa

Preparación

Para trabajar sobre la tercer parte de la consigna de este proyecto, necesitás abrir los siguientes archivos que se encuentran en la carpeta `js` :

1. `geocodificador.js`
2. `lugares.js`
3. `marcadores.js`

Recordá que los cambios que hagas sobre este proyecto se van a ver abriendo en tu navegador el archivo `index.html`.

Paso 1: Obtener las Coordenadas

Ahora ya podemos visualizar un mapa. La primera funcionalidad que vas a agregar es la de mostrar marcadores cuando se selecciona una dirección en el mapa. Para esto, primero es necesario que tu aplicación traduzca una dirección a una serie de coordenadas. Sin las coordenadas, el programa no entenderá dónde tiene que mostrar el marcador.

Entonces, vamos a empezar por programar una función en el archivo `geocodificador.js` que, dada una dirección, obtenga las coordenadas de esta. Para eso deberás completar la función `usaDireccion(geocodificador, direccion, funcionAllamar)` que usará el geocodificador para averiguar las coordenadas de `dirección` y llamará a `funcionAllamar`.

Ojo: Por como está implementada la función `funcionALlamar` *siempre se llama con los siguientes parámetros:*

1. *Dirección: la dirección pasada por parámetro.*

2. *Coordenada: la ubicación de tipo google.maps.LatLng*

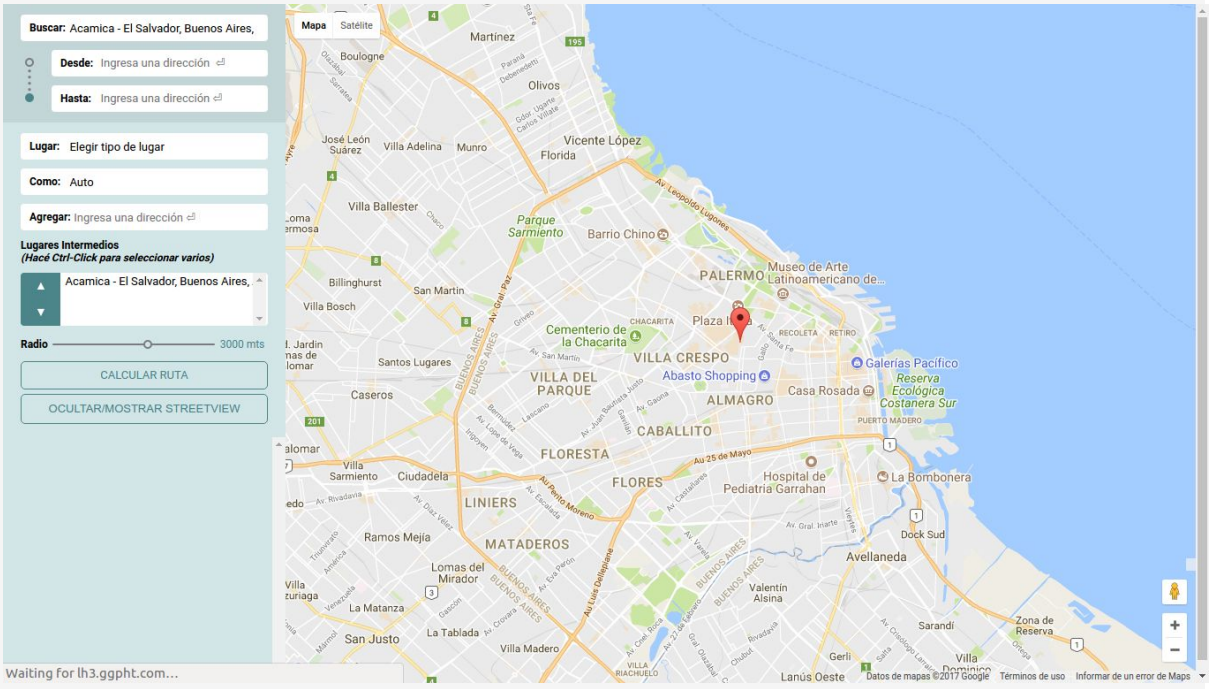
> Pista: Recordá que el geocodificador da como respuestas un arreglo de resultados, cada uno con su ubicación. Para ver como es la respuesta del geocodificador podés usar la consola.

Paso 2: Mostrar un marcador

Para mostrar el marcador hay que completar la función `mostrarMiMarcador(ubicacion)` ubicado en el archivo `marcador.js`, que cree un elemento con un título y una animación .

> Pista: Para crear un marcador vas a necesitar usar un constructor de la API. Revisá la documentación y encontrá la función necesaria.

Una vez implementada `mostrarMiMarcador(ubicacion)`, deberías ver el mapa así:



Paso 3: Buscar lugares cercanos

Vamos a empezar por agregar la funcionalidad de buscar lugares de algún tipo en especial, alrededor de la ubicación de un marcador. Los tipos de lugares son una categoría de cada establecimiento que los mismos locales definen en Google Maps, pudiendo ser restaurantes, estaciones de servicio, cines, etc. La idea es que nuestro mapa filtre solo los lugares del tipo que el usuario quiere encontrar, cerca del marcador que definió previamente.

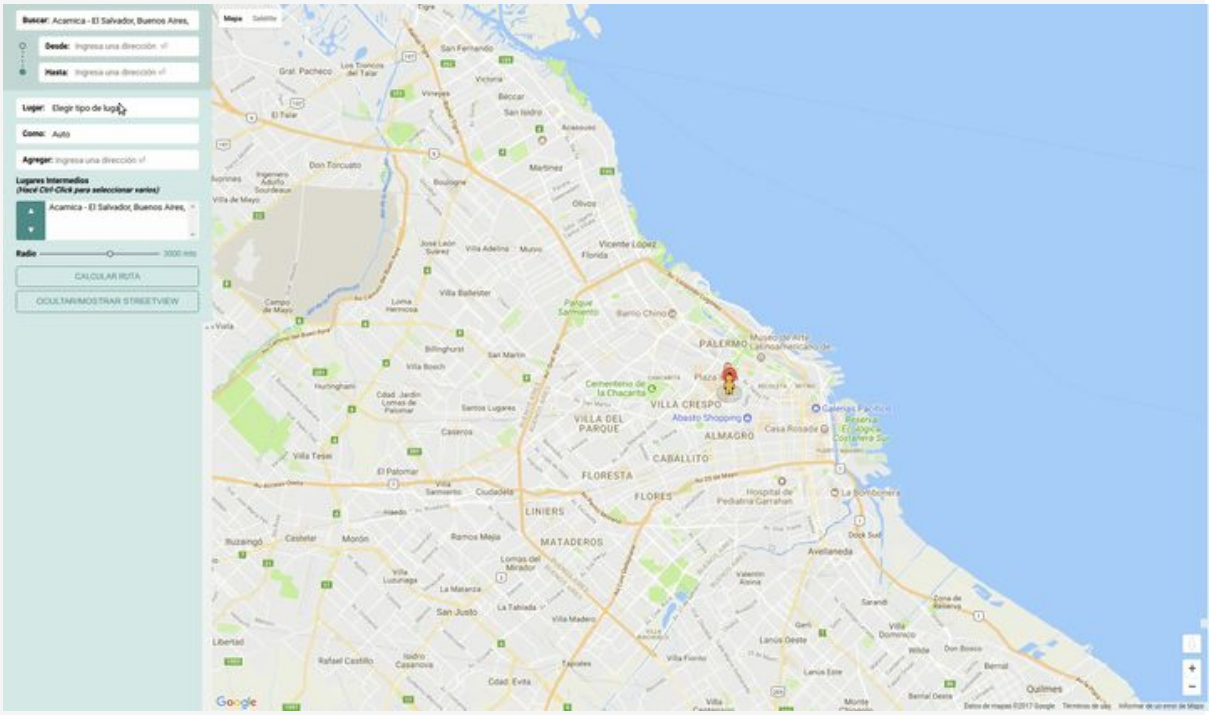
Completá la función `buscarCerca(posicion)`, de modo que realice una búsqueda de los lugares cercanos a una ubicación dada. La función deberá mostrar solo los elementos con el tipo de lugar y el radio especificado por el usuario.

Luego, utilizá la función `marcarLugares` del `marcadorModulo` para marcar los lugares obtenidos.

`buscarCerca` es utilizada desde el archivo `marcador.js`, en particular en la función `marcar`.

> Pista: Necesitás obtener el valor de los elementos `tipoDeLugar` y `radio`.

Ahora ya podés buscar lugares cerca. La página debería verse así:



Paso 4: Crear la función para autocompletar dirección

Ahora vamos a incorporar la función para autocompletar las direcciones con los datos que tenemos disponibles en la API. Para esto, vas a tener que implementar la función `autocompletar()`, ubicada en el archivo en `lugares.js`. Esta se ejecutará cada vez que el usuario empiece a escribir cualquier dirección y hará que el campo de texto se autocomplete.

Para limitar el campo de búsqueda y evitar que la aplicación haga sugerencias innecesarias, vamos a establecer un límite territorial dentro del cual va a buscar para autocompletar. Estos límites los podés crear haciendo un círculo (de Google Maps) con un cierto radio. Recomendamos que este sea 20000 metros para buscar sobre toda una ciudad, aunque lo podés modificar a tu gusto.

Nota: Este círculo tiene que crearse sin que el usuario final lo vea, por lo que tendría que estar oculto.

Pista: Ver en la documentación la clase Autocomplete de google.maps.places.

Una vez realizado los pasos, deberías poder buscar lugares o calles más fácilmente: