**Lab - Google Maps en Angular**

Angular 12 trajo consigo muchas novedades y características que hacen más productivo al Desarrollador que usa este Framework para crear sus proyectos, hay 2 componentes que trajo esta versión de Angular, uno es el componente YouTube y el otro componente es Google Maps y en este laboratorio se mostrara como usar e implementar un mapa de Google con este componente.

**Uso de GoogleMapsModule (@angular/ google-maps)**

GoogleMapsModule exporta tres componentes que podemos utilizar:

* GoogleMap: es la envoltura de Google Maps, disponible a través del selector google-map
* MapMarker: se utiliza para añadir marcadores en el mapa, disponible a través del selector map-marker
* MapInfoWindow: la ventana de información de un marcador, disponible a través del selector map-info-window

**Propiedades de entrada**

Se puede personalizar el estilo del mapa por defecto utilizando las propiedades @Input(). Las propiedades más utilizadas se añaden como propiedades @Input, podemos establecer el tamaño del mapa, establecer el centro y establecer el nivel de zoom.

|  |  |
| --- | --- |
| **Property** | **Descripcion** |
| height | Sets the initial height |
| width | Sets the initial width |
| center | Sets the initial center |
| zoom | Sets the initial zoom |
| options | [Sets the options](https://developers.google.com/maps/documentation/javascript/reference/map#MapOptions) |

**Eventos de GoogleMaps**

|  |  |  |
| --- | --- | --- |
| **operty** | **JavaScript API Method** | **Descripcion** |
| boundsChanged | bounds\_changed | Este evento se dispara cuando los límites de la ventana gráfica han cambiado |
| centerChanged | center\_changed | Este evento se dispara cuando la propiedad del centro del mapa cambia |
| mapClick | click | Este evento se dispara cuando el usuario hace clic en el mapa |
| mapDblclick | dblclick | Este evento se dispara cuando el usuario hace doble clic en el mapa. |
| mapDrag | drag | Este evento se dispara repetidamente mientras el usuario arrastra el mapa |
| mapDragend | dragend | Este evento se dispara cuando el usuario deja de arrastrar el mapa |
| mapDragstart | dragstart | Este evento se dispara cuando el usuario comienza a arrastrar el mapa |
| headingChanged | heading\_changed | Este evento se dispara cuando la propiedad del encabezado del mapa cambia |
| idle | idle | Este evento se dispara cuando el mapa queda inactivo después de hacer paneo o zoom |
| maptypeidChanged | maptypeid\_changed | Este evento se dispara cuando la propiedad mapTypeId cambia |
| mapMousemove | mousemove | Este evento se dispara cuando el ratón del usuario se mueve sobre el contenedor del mapa |
| mapMouseout | mouseout | Este evento se dispara cuando el ratón del usuario sale del contenedor del mapa |
| mapMouseover | mouseover | Este evento se dispara cuando el ratón del usuario entra en el contenedor del mapa |
| projectionChanged | projection\_changed | Este evento se dispara cuando la proyección ha cambiado |
| mapRightclick | rightclick | Este evento se dispara cuando el evento DOM contextmenu se dispara en el contenedor del mapa |
| tilesloaded | tilesloaded | Este evento se dispara cuando los mosaicos visibles han terminado de cargarse |
| tiltChanged | tilt\_changed | Este evento se dispara cuando la propiedad de inclinación del mapa cambia |
| zoomChanged | zoom\_changed | Este evento se dispara cuando la propiedad de zoom del mapa cambia |

**Implementación**

Lo primero que haremos es crear un nuevo proyecto con Angular, ejecutando el siguiente comando:

|  |
| --- |
| ng new appGoogleMaps  ...  ... (Continua la instalación)  ...    # Ingreso al directorio del proyecto creado  cd appGoogleMaps |

Ahora se instala el componente oficial Google Maps de Angular, ejecutando el siguiente comando:

|  |
| --- |
| npm install @angular/google-maps --save    + @angular/google-maps@9.2.3  updated 1 package and audited 16282 packages in 16.872s |
|  |

**Get Maps API Key**

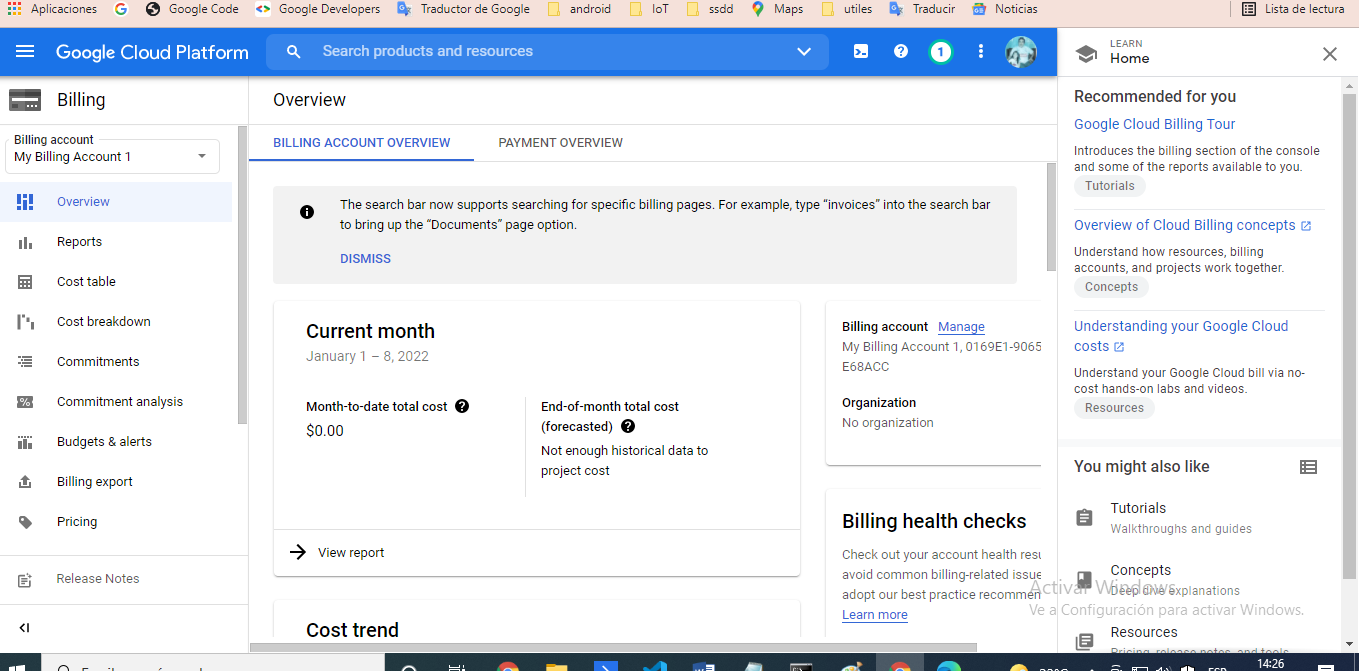
Ahora, se debe configurar el plugin de google maps en la aplicación angular. Así que, en primer lugar, debe obtenerse la clave de la API de mapas, y a continuación se observara como obtener dicha clave.

**Uso de las claves API**

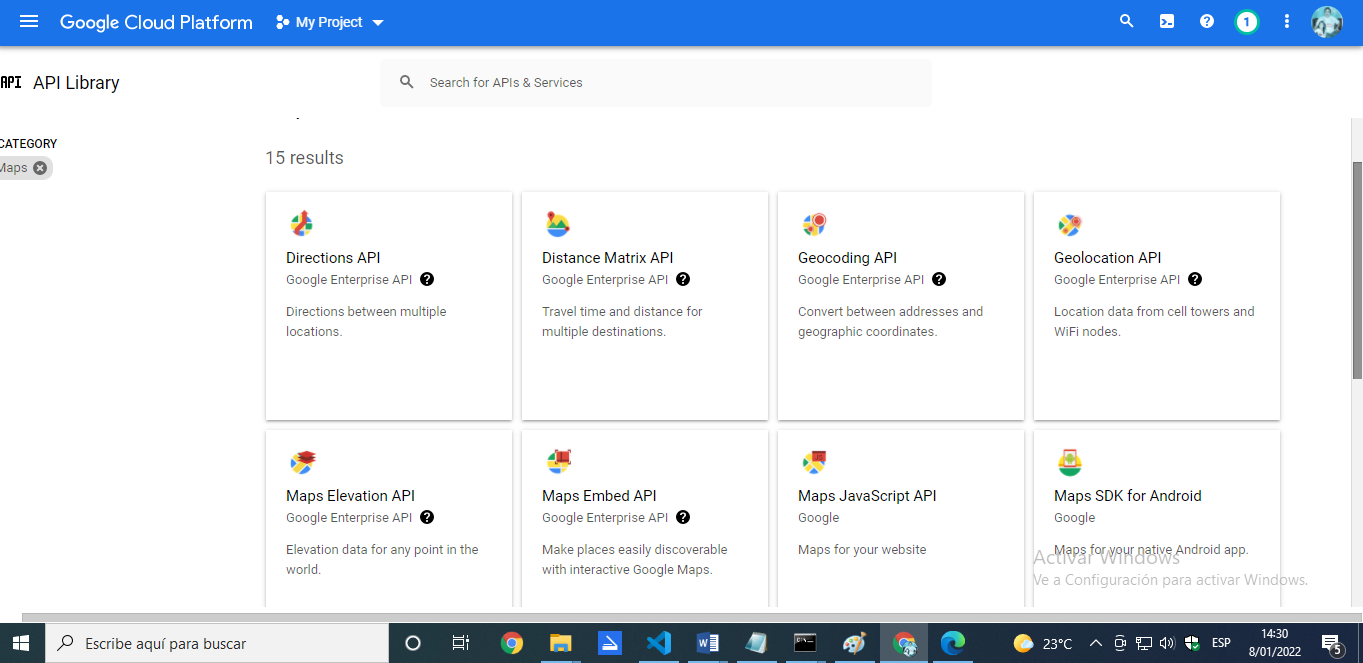
Los productos de Google Maps Platform están protegidos contra el uso no autorizado restringiendo las llamadas a la API a aquellas personas que proporcionan las credenciales de autenticación adecuadas. Estas credenciales adoptan la forma de una clave de API, una cadena alfanumérica única que asocia tu cuenta de facturación de Google con tu proyecto y con la API o el SDK específicos.

**Antes de empezar**

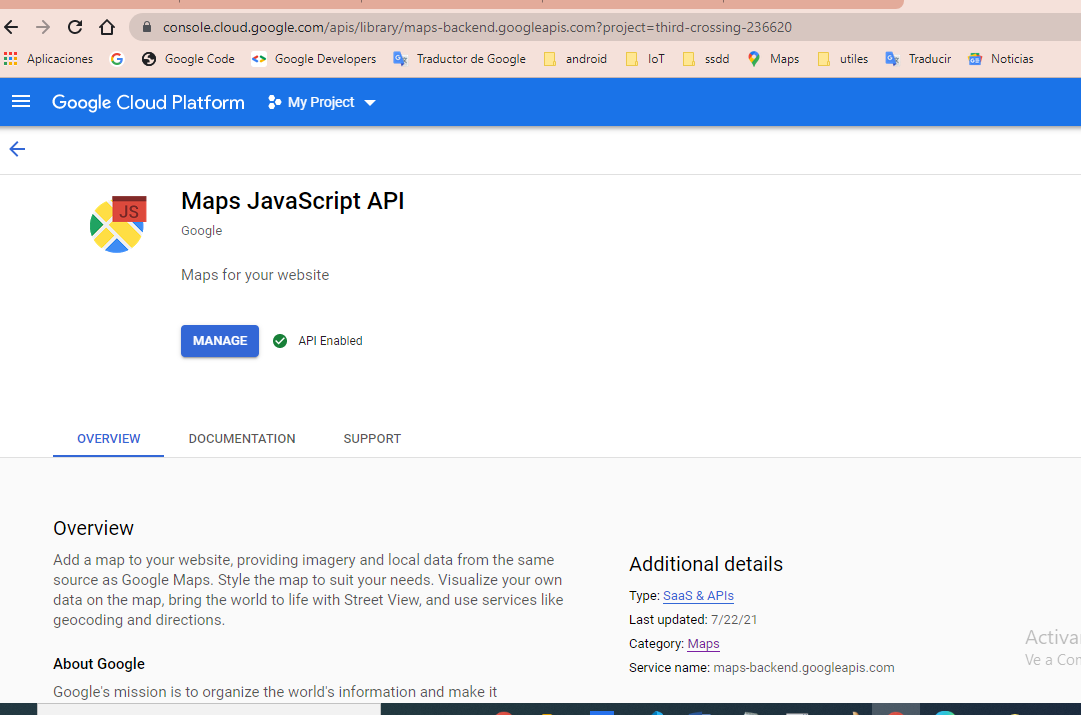
Antes de empezar a utilizar Maps JavaScript API, necesitas un proyecto con una cuenta de facturación y Maps JavaScript API habilitado.



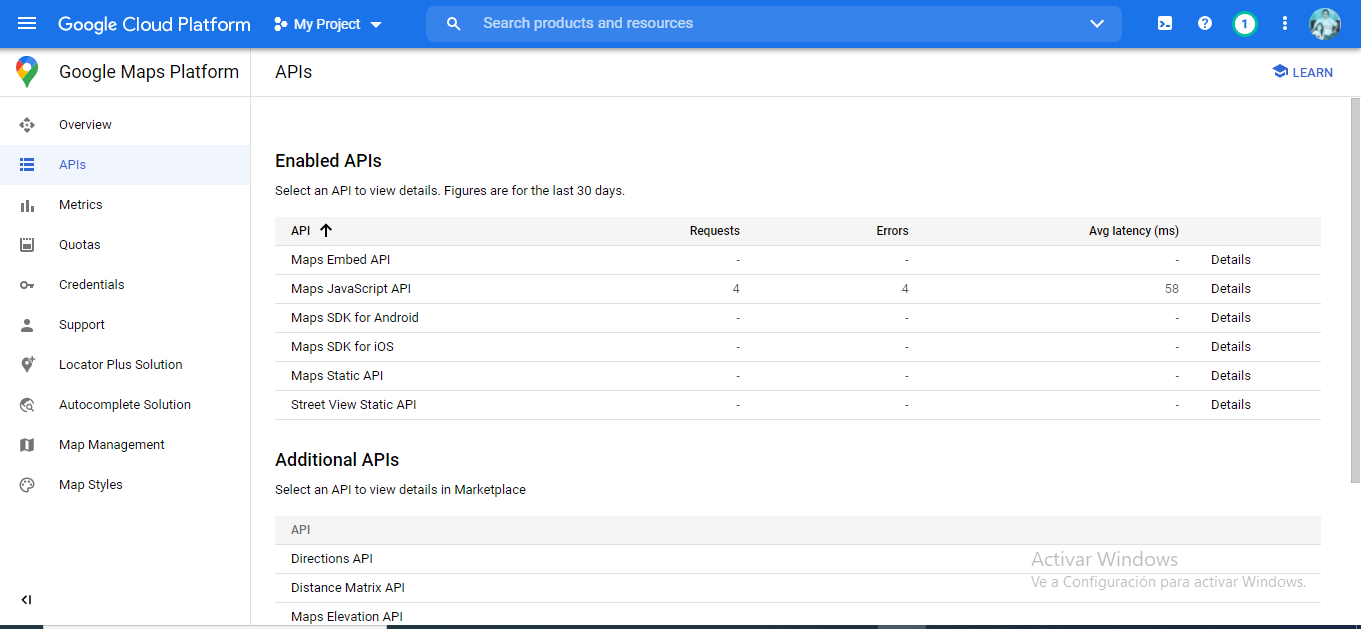
Luego en el menú de la izquierda escoger APIs y de allí Maps



Al presionar Maps Javascript API, se muestra la siguiente pantalla:



Se presiona Manage y se muestra un menu a la izquierda, por lo que se escoge la opcion de credenciales



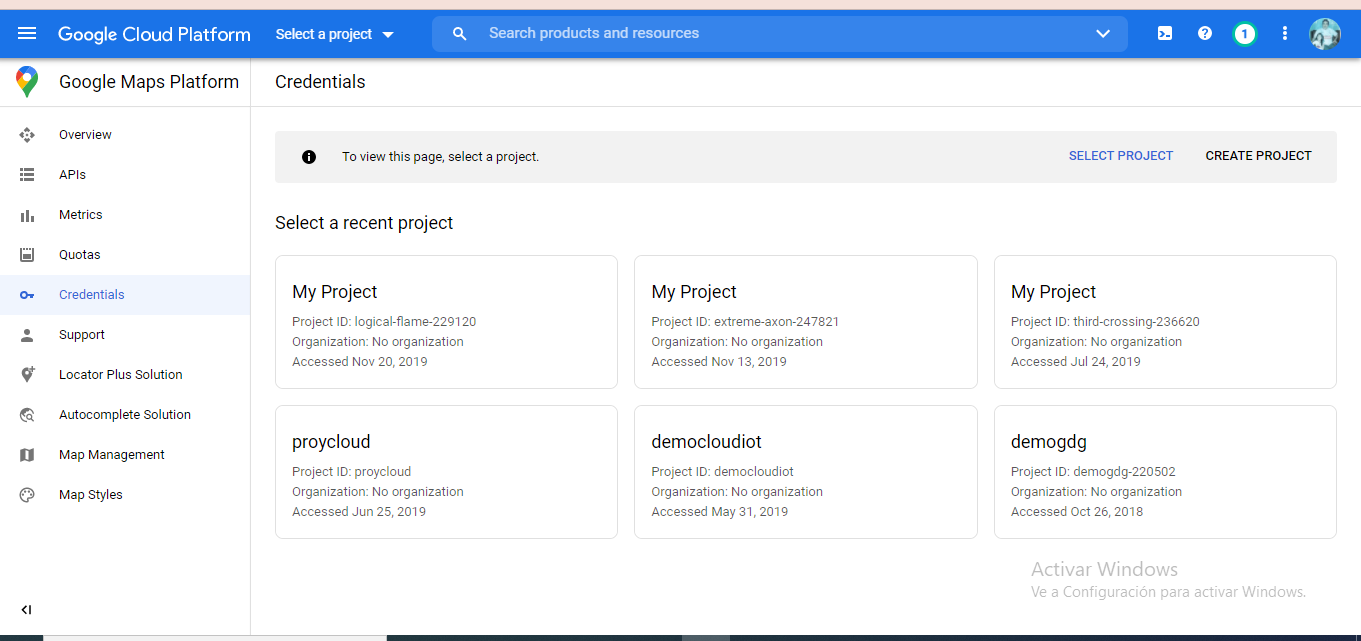
**Creación de API Key**

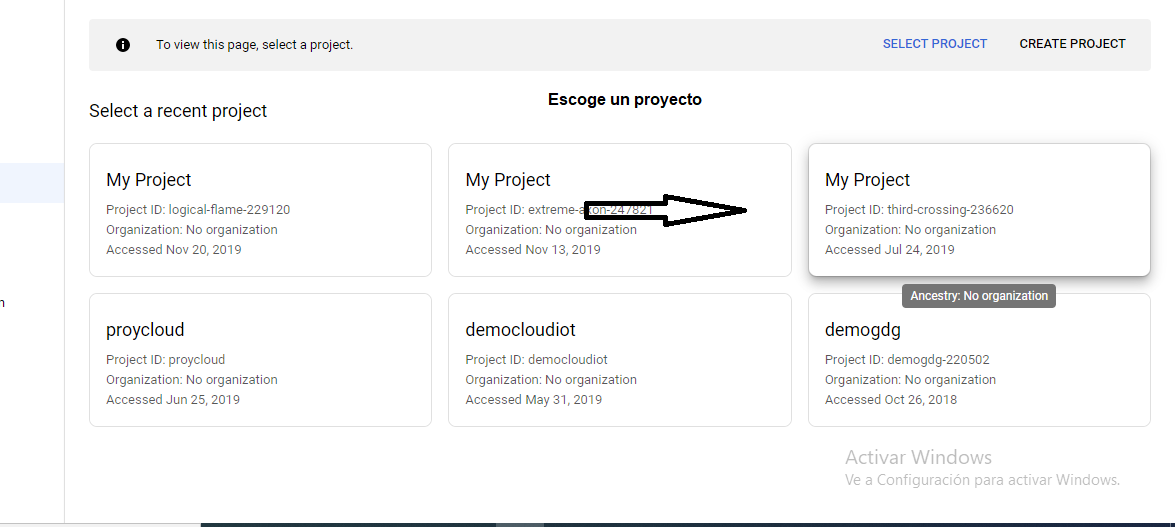
La clave de API es un identificador único que autentifica las solicitudes asociadas a su proyecto a efectos de uso y facturación. Debe tener al menos una clave de API asociada a su proyecto.

Para crear una clave de API:

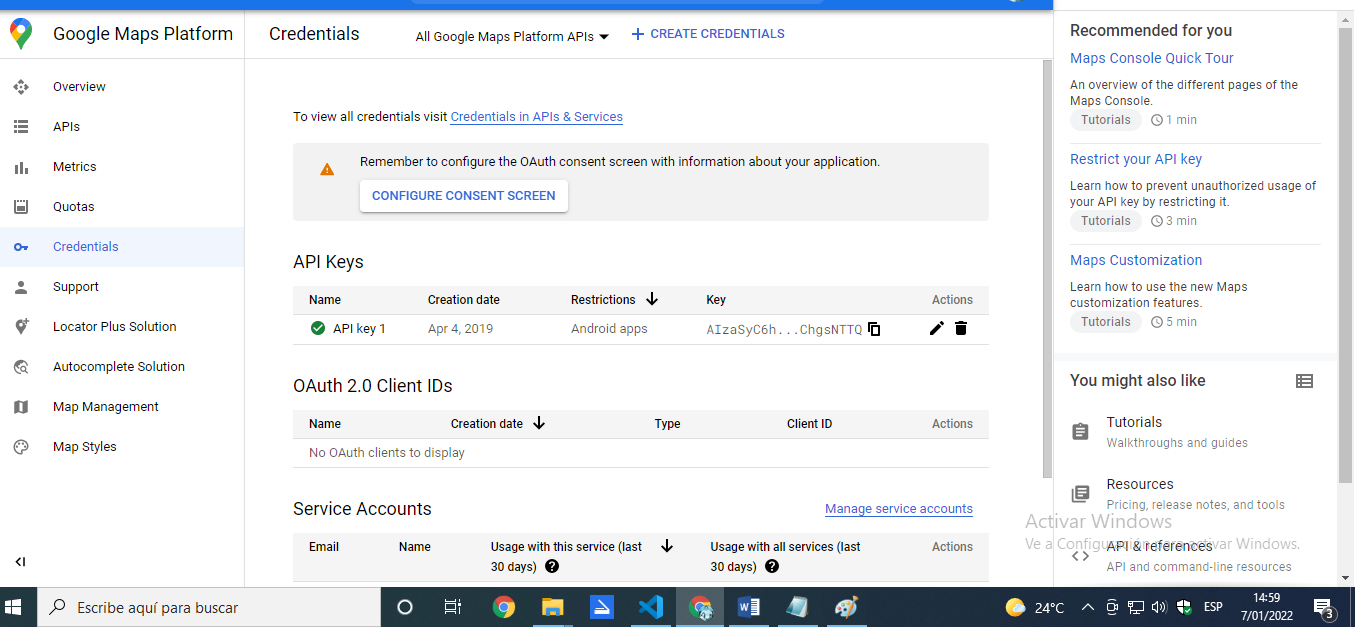
**Consola**

1. Ir a la página Google Maps Platform > Credenciales.
   1. Ir a la página de credenciales

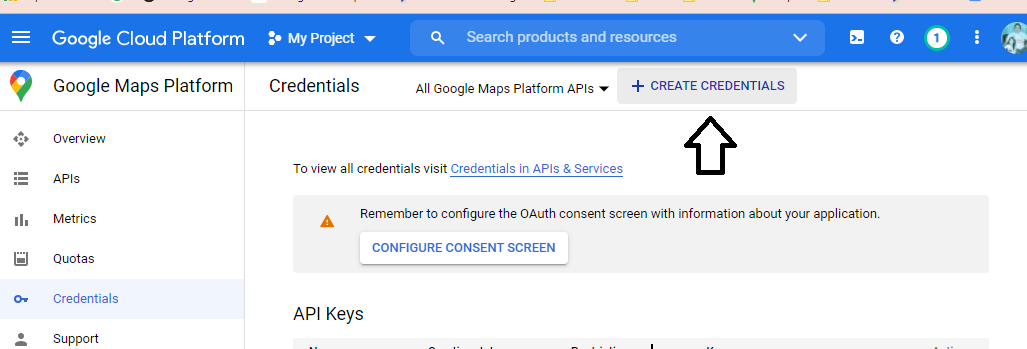




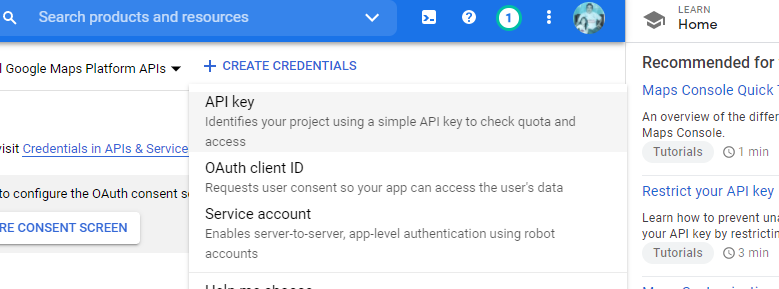
Si no se cuenta con algún proyecto, se debe crear uno.

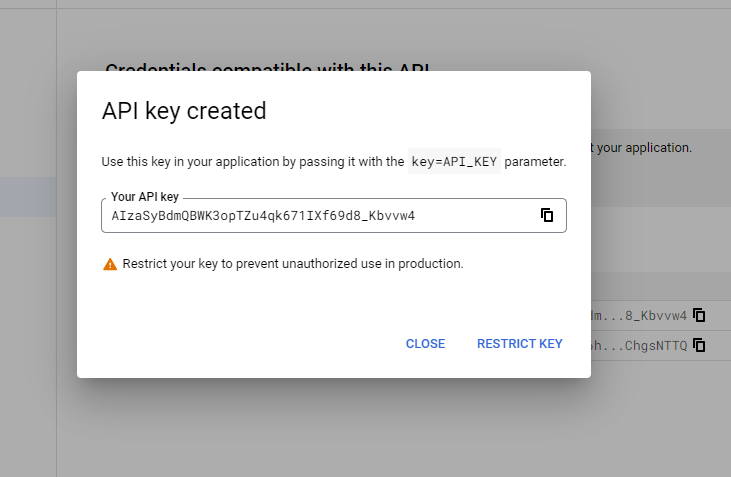


Presionar CREATE CRDENTIALS

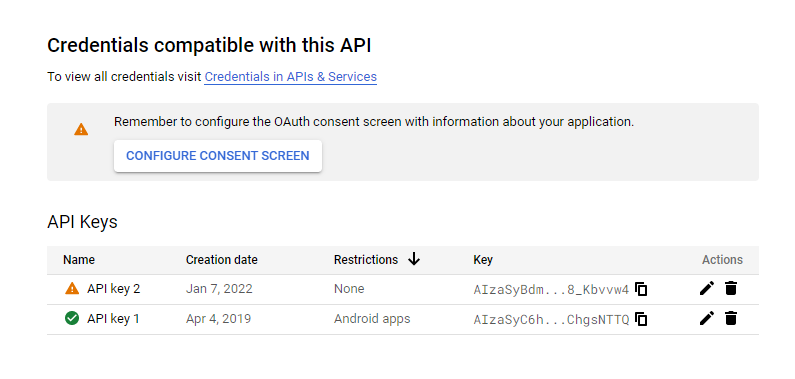


Y luego escoger API KEY





Se presiona CLOSE



La nueva clave de la API aparece en la página de credenciales bajo claves de la API.

(Recuerde restringir la clave de API antes de utilizarla en producción).

Después de crear el nuevo proyecto, se ha creado un conjunto de directorios y archivos, y a continuacion se abre el archivo **app.module.ts**, este archivo se encuentra en **appGoogleMaps** > **src** > **app** > **app.module.ts**

|  |
| --- |
|  |

Dentro del archivo **app.module.ts** agrego lo siguiente (En este archivo se invoca a Google Maps de Angular)

|  |
| --- |
| import { BrowserModule } from '@angular/platform-browser';  import { NgModule } from '@angular/core';    // Google Maps de Angular  import {GoogleMapsModule} from '@angular/google-maps';    import { AppComponent } from './app.component';    @NgModule({    declarations: [      AppComponent    ],    imports: [      BrowserModule,      GoogleMapsModule,    ],    providers: [],    bootstrap: [AppComponent]  })  export class AppModule { } |

Al añadir el componente Google Maps a una plantilla ya se puede ver y utilizar el mapa de Google. El mapa se comportará como un mapa por defecto con la funcionalidad por defecto, por ejemplo, se puede acercar y alejar la imagen, y arrastrar en el mapa.

Paso seguido se abre el archivo **app.component.ts** que se encuentra en **appGoogleMaps** > **src** > **app** > **app.component.ts**

|  |
| --- |
|  |

Dentro del archivo **app.component.ts** debo colocar los métodos tal cual estan en la API de Google Maps (JavaScript) para que pueda funcionar nuestro mapa, añadiendose lo siguiente.

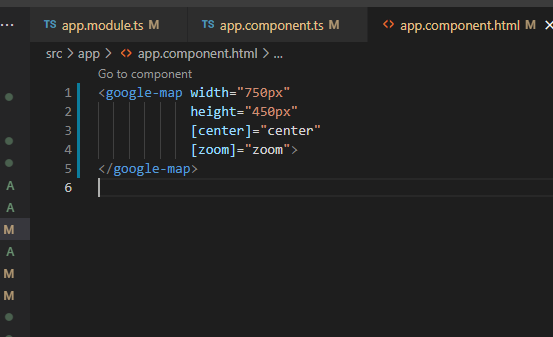
|  |
| --- |
| import { Component, ViewChild } from '@angular/core';  import {MapInfoWindow, MapMarker} from '@angular/google-maps';    @Component({    selector: 'app-root',    templateUrl: './app.component.html',    styleUrls: ['./app.component.css']  })  export class AppComponent {      title = 'Como usar el Componente Google Maps de Angular';      // Configuración de Google Maps    center = {lat: -11.9865699, lng: -77.0679477};  zoom = 15;  display?: google.maps.LatLngLiteral;  } |

Por último se abre el archivo **app.component.html** que se encuentra en **appGoogleMaps** > **src** > **app** > **app.component.html**

|  |
| --- |
|  |

En el archivo **app.component.html** el cual es el archivo HTML del proyecto, se agrega lo siguiente:

|  |
| --- |
| <google-map width="750px"              height="450px"              [center]="center"              [zoom]="zoom">  </google-map> |



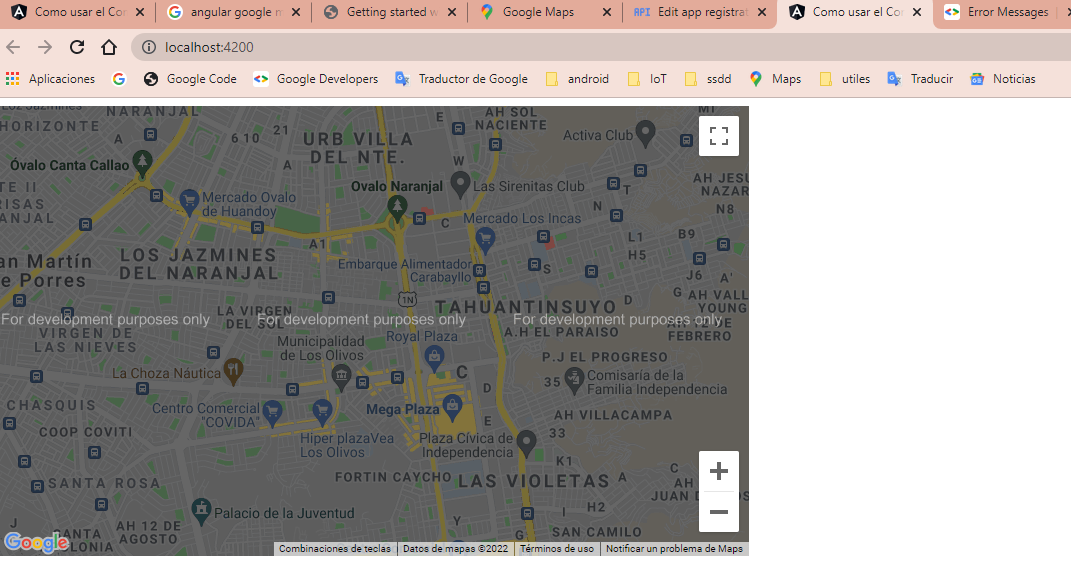
Por último debo colocar mi API KEY de Google Maps, para esto abrimos el archivo **index.html** que se encuentra en **appGoogleMaps** > **src** > **index.html**

|  |
| --- |
| <!doctype html>  <html lang="en">  <head>    <meta charset="utf-8">    <title>Como usar el Componente Google Maps de Angular </title>    <base href="/">    <meta name="viewport" content="width=device-width, initial-scale=1">    <link rel="icon" type="image/x-icon" href="favicon.ico">  </head>  <body>      <app-root></app-root>      <!-- Mi API KEY de Google Maps -->    <script src="https://maps.googleapis.com/maps/api/js?key=ACA-COLOCA-TU-API-KEY">    </script>    </body>  </html> |

Ahora se ejecuta el servicio de Angular ejecutando el siguiente comando:

|  |
| --- |
| ng serve -o |

Y se obtiene el mapa de Google (Se está usando la plantilla HTML por defecto que me brinda Angular al crear un proyecto)



**Mejoras del proyecto inicial**

Se agregarán algunas mejoras al proyecto inicial para mostrar otras características de la librería Google-maps.

**app.components.ts**

1. se agregar la interface OnInit en esta clase, de la siguiente manera:

import { Component, OnInit } from '@angular/core';

1. de esta manera, se redefine la clase de la siguiente manera:

export class AppComponent implements OnInit{

1. esto permite la declaración del método ngOnInit con el código siguiente:

ngOnInit(): void {

navigator.geolocation.getCurrentPosition((position) => {

this.center = {

lat: position.coords.latitude,

lng: position.coords.longitude,

}

})

}

1. Se redefine las variables de entrada de esta clase, de la siguiente manera:

// Configuración de Google Maps

center: google.maps.LatLngLiteral;

zoom = 15;

options: google.maps.MapOptions = {

mapTypeId: 'hybrid',

zoomControl: false,

scrollwheel: false,

disableDoubleClickZoom: true,

maxZoom: 15,

minZoom: 8,

}

1. Debe agregarse los métodos de control de Zoom, de la siguiente manera:

zoomIn() {

if (this.zoom < this.options.maxZoom)

this.zoom++

}

zoomOut() {

if (this.zoom > this.options.minZoom)

this.zoom--

}

De esta forma el código completo, será el siguiente:

import { Component, OnInit } from '@angular/core';

import {MapInfoWindow, MapMarker} from '@angular/google-maps';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit{

  title = 'Como usar el Componente Google Maps de Angular';

  ngOnInit(): void {

   navigator.geolocation.getCurrentPosition((position) => {

      this.center = {

        lat: position.coords.latitude,

        lng: position.coords.longitude,

      }

    })

  }

  // Configuración de Google Maps

  center: google.maps.LatLngLiteral;

  zoom = 15;

  options: google.maps.MapOptions = {

    mapTypeId: 'hybrid',

    zoomControl: false,

    scrollwheel: false,

    disableDoubleClickZoom: true,

    maxZoom: 15,

    minZoom: 8,

  }

  zoomIn() {

    if (this.zoom < this.options.maxZoom)

         this.zoom++

  }

  zoomOut() {

    if (this.zoom > this.options.minZoom)

         this.zoom--

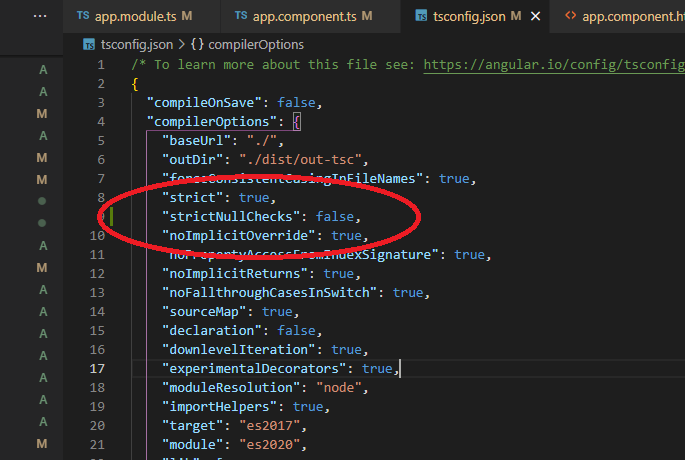
  }

}

**Nota importante:**

El código desarrollado en zoomIn y zoomOut genera un mensaje de error debido a que las variables minZoom y maxZoom pueden ser de tipo null, según el compilador de Typescript. Para evitar este error (de tipo falso positivo) , debe agregarse en el archivo tsconfig.ts la línea siguiente:

"strictlyNullChecks": false

****

**app.component.html**

Con el nuevo código se redefine la parte visual de este componente, de la siguiente manera:

<h2>Google Maps en Angular 12</h2>

<google-map width="800px"

height="500px"

[center]="center"

[zoom]="zoom"

[options]="options">

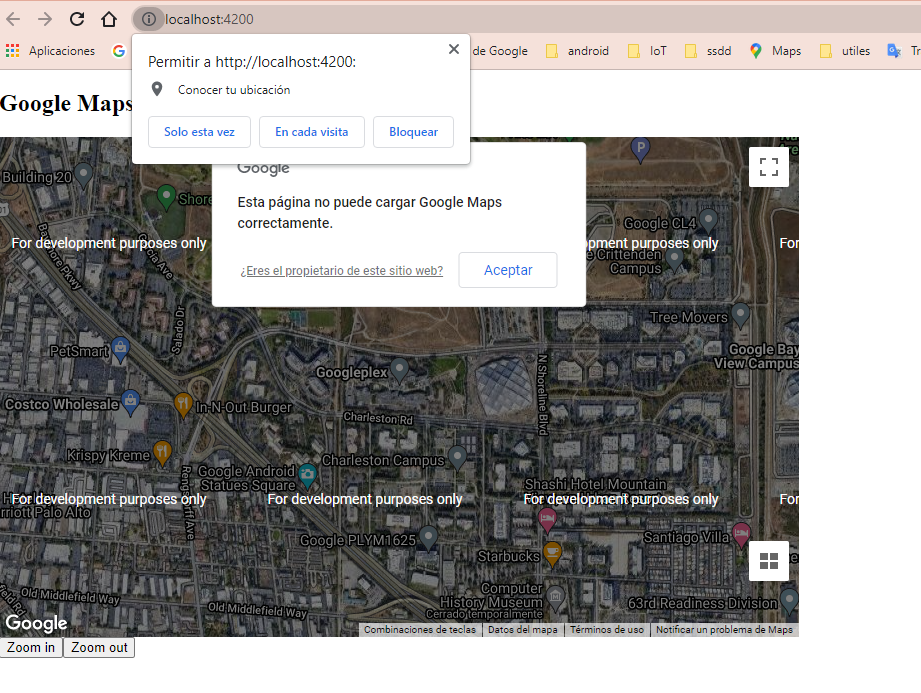
</google-map>

<!-- Use custom zoom buttons -->

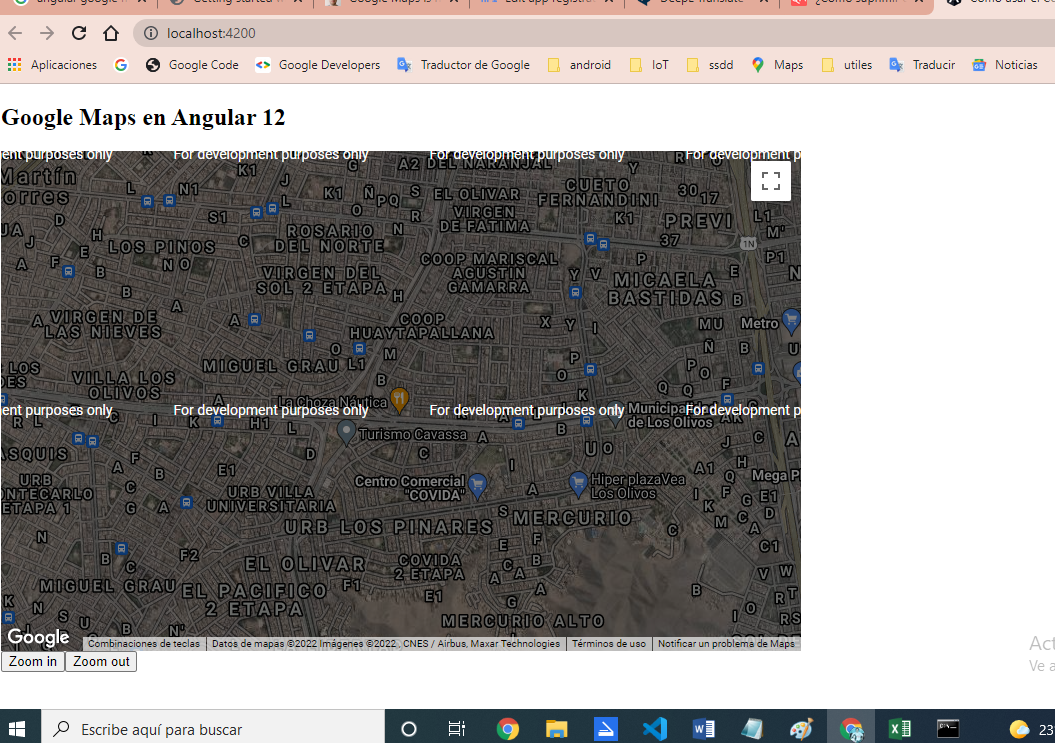
<button (click)="zoomIn()">Zoom in</button>

<button (click)="zoomOut()">Zoom out</button>

De esta forma, al ejecutarse el código completo obtenemos la siguiente salida:



Detectada la localización, se muestra la posición en el mapa

****

**Uso de eventos de GoogleMap**

Como se ha visto, en la primera parte de este laboratorio, se ha mostrado los eventos que esta librería ofrece. En este caso se aplicará el evento *click*.

En *app.component.html* se modifica la etiqueta google-map de la siguiente manera:

<google-map width="800px"

            height="500px"

            [center]="center"

            [zoom]="zoom"

            [options]="options"

            (mapClick)="click($event)">

</google-map>

A continuación, en *app.component.ts* se agregara el siguiente método:

click(event: google.maps.MapMouseEvent) {

    console.log(event)

  }

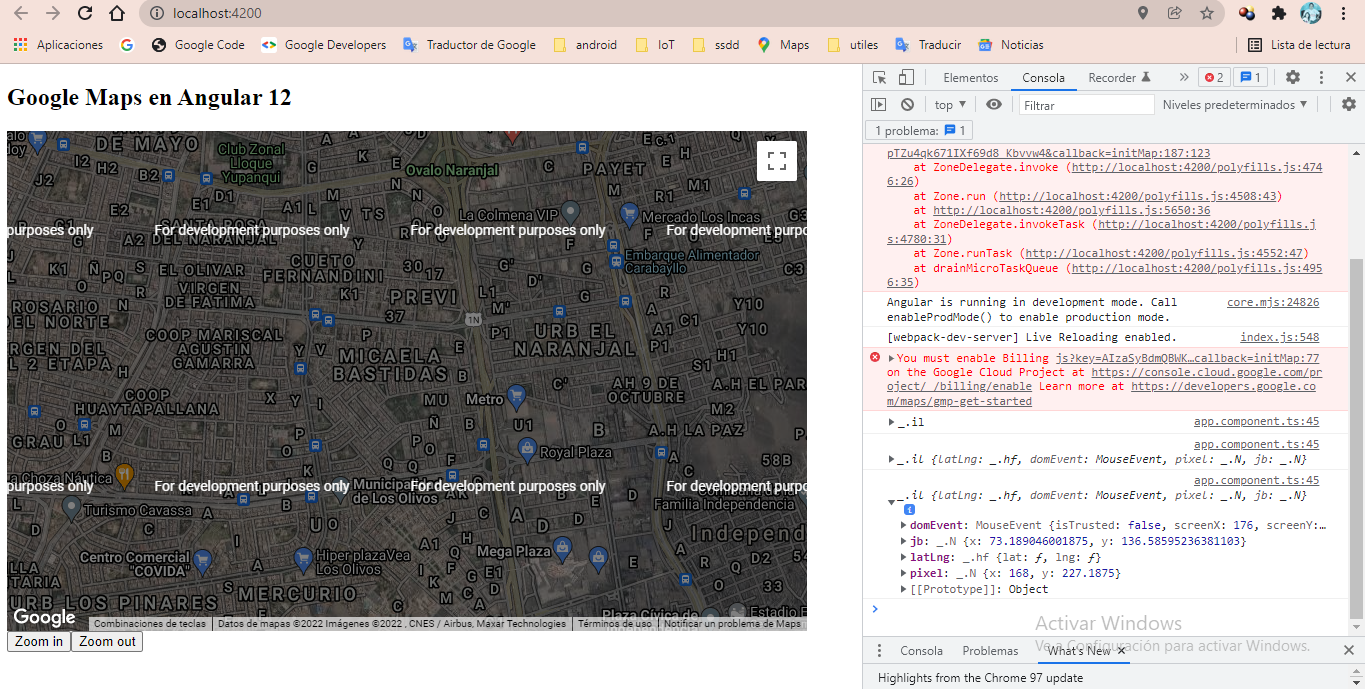
Sin embargo, aparecerá un error en MouseEvent, que aparece como indefinido. Esto se corregirá con la instalación de una librería (hoy deprecated) denominada @types/googlemaps por lo que se instalara de la siguiente manera:

npm i @types/googlemaps@latest

para la actualización en la última versión de Angular 12, se instalará de la siguiente forma:

npm i @types/google.maps@latest´

cambiando por ello, el nombre del evento a MapMouseEvent



Angular para que funcione sin generar problemas con Angular 12.