# Proyecto MapModule

SISTEMA DE MEDICION DE AREAS

EQUIPO #3:
ENCINAS VERDUZCO OSCAR IGNACIO
SODARI SAMANIEGO IVAN ALONSO
SOTO PALAFOX GERMAN

FECHA: 2/DIC/2019

# Índice

## Introducción

Realizar mediciones de terrenos nunca había sido tan fácil como con las herramientas que te proporcionan aplicaciones como google maps, y llevar estas funcionalidades a casos más específicos pueden llegar a ser muy útiles para empresas que requieran estas mediciones para realizar cualquier actividad, como hacer cotizaciones exactas de forma rápida y eficaz, para ofrecerle un mejor servicio al cliente. Esto es lo que busca este proyecto, el cual será explicado de manera más detallada a lo largo de este documento.

## Objetivo

Se busca crear una herramienta para poder medir el perímetro y área de los terrenos externos de una casa, entiéndase como: patio frontal, patio trasero, lateral derecho y lateral izquierdo. Pudiendo guardar el área obtenida de cada terreno en pies cuadrados, colocar etiquetas o modificar dichas medidas, con el objetivo de que empresas interesadas puedan utilizar esta información para sus diferentes preferencias. Es importante considerar que las medidas ya obtenidas podrán ser consultadas cuantas veces sea necesario.

## Planteamiento del problema

A la hora de requerir un servicio, ya sea de jardinería o de algún tipo de construcción, el costo de dicho trabajo será en base al área a trabajar, El calcular esta suele requerir ir a la propiedad y hacer mediciones antes de dar una cotización. Por lo que se requiere de una herramienta que permita tomar mediciones sin estar presencialmente en el lugar donde se pretende laborar, también guardar dichas mediciones para futuras llamadas de clientes recurrentes.

## Alcance del proyecto

La solución a la problemática planteada consiste en utilizar las API’s que proporciona Google Maps orientada a JavaScript implementándolas en una página web para identificar las direcciones deseadas, y poder mediante una herramienta de medición obtener los valores de los perímetros requeridos de dicho terreno, guardándose en la base de datos, para su posterior consulta y/o utilización.
Las tecnologías a utilizar se dividirán en Frontend y Backend. Para el Frontend se usará HTML 5, CSS con su framework bootstrap, JavaScript para implementaciones de las API’s, y también se implementará la librería de JavaScript ReactJs. Para el Backend se

utilizará el framework ASP.NET Core con su lenguaje C#, encargado de realizar los algoritmos que se requieran, para conectar con la base de datos SQL Server, se usará entity framework. Para tener un orden en el código, se implementará la arquitectura de software MVC (Modelo, Vista, Controlador).
La medición consistirá en trazar un polígono sobre el terreno deseado mostrado por la API de Google Maps, y utilizar la magnitud del zoom momentáneo que tenga el mapa, para calcular el área a escala en pies cuadrados de la figura trazada, utilizando un algoritmo que se encargue de procesar toda la información introducida.
Contará con una sección donde se podrá consultar todas las medidas ya registradas, para que las diversas empresas que requieran esa información puedan utilizarla según sean sus conveniencias.
También se implementará un control de usuarios mediante un login que se encargue de que haya un flujo controlado en la página web.

## Justificación

Cuando un cliente requiere de algún servicio, la pregunta más importante que se debe responder es ¿Cuál es el Precio? Y como servidores idealmente debemos dar una respuesta precisa, sin embargo, esto requeriría o bien, mandar un experto al área y que cotice, o bien, dar estimados basados en experiencias previas, dando oportunidad a una aproximación errónea. Es ahí donde entra el sistema. Este nos permitirá dar al cliente un presupuesto con un margen de error mínimo, otorgandonos así, un cálculo preciso y mostrando un ahorro, pues no se requerirá que se encuentre físicamente ahí un experto, en lo que respecta al primer contacto con el cliente.

### Desarrollo del proyecto

El desarrollo del proyecto se centra principalmente en dos etapas:
1. Desarrollo de una página web con la librería de javascript React.Js
React Js es una librería de javascript desarrollada por facebook para construir interfaces de usuario mediante componentes. Nos permite un desarrollo ágil, ordenado y con una arquitectura mantenible, usando componentes y que nos ofrece un gran rendimiento. Es actualmente una de las librerías más utilizadas para el desarrollo de interfaces y más demandadas dentro del mundo de la programación de front-end.
El proyecto desarrollado cuenta con un index de donde se renderizan los 9 componentes con los que cuenta el sistema, los cuales son: App, AddSurfaceArea, Area, LeftPanel, SearchBar, AreaList, EditSurfaceArea, PanelContent y PanelHeader.
Además, cuenta con una carpeta llamada util, donde podemos encontrar:

- AddressList: la cual guarda una lista de direcciones a las cuales puede consultar el sistema
- Map: en donde se guarda todo lo relevante de la API de google maps para poder interactuar con los usuarios.

2. Desarrollo de una APIRest con la tecnología de ASP.NET Core
ASP.NET Core fue desarrollado por microsoft, es un framework nuevo de código abierto y multiplataforma para la creación de aplicaciones modernas conectadas a Internet, como aplicaciones web y APIs Web. Dentro de sus ventajas encontramos que se pueden desarrollar y ejecutar aplicaciones ASP.NET Core en Windows, Mac y Linux.
Para este proyecto desarrollamos una API Rest, la cual consiste en realizar la función de web service, la cual se encarga de recibir las peticiones http que realizan los usuarios dentro de la página realizada en react y procesarlas hacia la base de datos, regresando los valores que el usuario está esperando.
Se crearon cuatro carpetas las cuales son: Data, Models, Controllers y DTOs

- Data: en esta sección se establece el contexto hacia la base de datos, manejando las propiedades necesarias para su conexión exitosa.
- Models: aquí se encuentran representadas las dos tablas de la base de datos, location y área, con sus atributos correspondientes.
- Controllers: en estas clases se encuentran los dos controladores de las dos tablas que contiene la base de datos, área y location, aquí es donde se establecen todos los métodos de get, post, put y delete.
- DTOs: un data transfer object consiste en realizar como una copia de una tabla para regresar esta en cualquier petición y evitar redundancia entre llaves primarias y foráneas.

Además de configurar el Startup y el appsettings.json para su funcionamiento correcto.

## Posibles Riesgos

Cuando se trata de aplicaciones web hay varios riesgos. Los más importantes son:
Disponibilidad o Uptime del sistema: Siendo que será alojado en azure, el servicio de alojamiento nos garantiza un 99.995% de uptime, lo que quiere decir que nos garantiza que el sistema esté disponible esa cantidad de tiempo. Pero abra un .005% en el que se encuentre abajo debido al proveedor.

- Seguridad: Al estar en internet, la aplicación es susceptible a ser atacada por hackers. Entre los ataques más comunes y los que afectarían a una aplicación de este tipo son:
    1. DoS y DDos(Denial of Service): Afecta la disponibilidad del sistema.

2. Ataque de diccionario: Ya que pueden conseguir contraseñas de un usuario y/o administrador, y esto puede afectar la integridad del sistema.
3. SQL injection: Pone en peligro la integridad de la base de datos
4. XSS: Este tipo de ataque consiste en subir scripts de JavaScript en el servidor

- Disponibilidad de google maps: Google nos garantiza contar con una disponibilidad de tiempo de al menos 99.9% de google maps, lo cual deja un margen de un máximo de .1% de tiempo en el que probablemente no esté disponible en su totalidad.

## Evaluación económica del proyecto

Al ser una aplicación web, entre los costos se encuentran el costo de desarrollo, la adquisición del dominio y finalmente, el costo de alojamiento de la aplicación en la nube.
Para específicos:
Equipo de desarrollo: Al ser desarrollado por estudiantes de 7mo semestre, y al estar conformado por 3 integrantes el equipo y con un tiempo de desarrollo de 2 meses, el cálculo fue el siguiente:
número de integrantes * mensualidad a programador web junior * porcentaje de la carrera completado (7/9) *tiempo de desarrollo(2 meses)
Dominio Web: El mínimo para comprar un dominio es por un tiempo de un año. En GoDaddy(Tienda de dominios). El costo por un año es de 159.99 y 339.99 la renovación cada año.
Alojamiento: El alojamiento por mes para el paquete S1 es de 1338.45 pesos.
Total: Lo anterior nos da un total de cincuenta y cinco mil novecientos cuarenta y cuatro pesos con cuarenta y cinco centavos

*Tabla 1. Evaluación económica del proyecto*

| Cantidad | Unidad | Concepto | Precio |
|---|---|---|---|
| 2 | Mes | Equipo de desarrollo(3 programadores web junior) | 54446.00 |
| 1 | Año | Dominio Web(nombre del sitio) | 159.99 |
| 1 | Mes | Alojamiento en la nube(Azure) del Sistema. Paquete S1 | 1338.45 |
| Total | | | MXN 55944.44 |

## Resultados esperados

Como sistema/proyecto en sí se espera contar con un sistema con una disponibilidad mayor al 99.9%. Donde, al ingresar, el usuario este será capaz de realizar, modificar y mostrar mediciones en una dirección ingresada por el usuario, esto bajo la premisa que esta se encuentre registrada en la base de datos.
Como solución para la empresa, se espera que el sistema sea de agrado a la empresa en cuanto a diseño y experiencia de usuario, de forma que sea intuitiva y tácita la forma de aprender a usarlo, y que sea más veloz que el módulo de medición que ya poseen.

## Impacto del proyecto

Se pretende que el sistema funja como herramienta para la empresa a la hora de realizar mediciones, y al ser más rápidas que hacerlo físicamente, presentar un ahorro en cuanto a contratación de personal que las realice y también en cuanto al tiempo, pues es más rápido realizar estas mediante el uso de software.

## Anexos

### Peticiones por segundo que resuelven Java, ASP.NET Core y Node.js

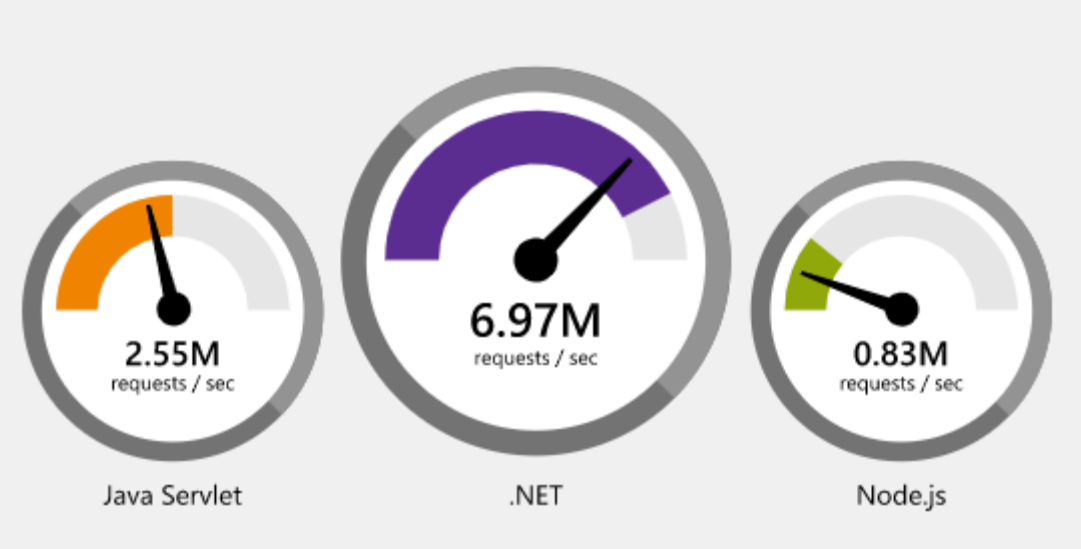



*Figura 1. Peticiones resueltas por segundo*

## Otros puntos

## Presentación del equipo y sus roles en el proyecto

- **Oscar Encinas:** codificación de servicios rest y backend, creación de la base de datos, Alojamiento de la aplicación de la API Rest en la nube.
- **Ivan Sodari**: diseño de la arquitectura, diseño de la interfaz de usuario, diseño de la base de datos, Creación de componentes, Implementación de la API de Google Maps, Creación de Polígonos, Alojamiento de la interfaz de usuario en la nube.
- **German Soto:** Implementación del algoritmo de medición, diseño de la documentación.

## Descripción de componentes

### Backend:

Encontramos la estructura básica de un API Rest, haciendo la función de un web service, resolviendo las peticiones realizadas por los usuarios que utilicen el sistema, las peticiones se dividen en 4:

1. GET: se encarga de traer información de la base de datos, ya sea todos los datos en general, o un registro en específico, dependiendo de la url ingresada por el usuario.
2. POST: es la encargada de guardar registros dentro de la base de datos.
3. PUT: se encarga de actualizar valores dentro de la base datos.
4. DELETE: es la encargada de eliminar un registro específico de la base de datos.

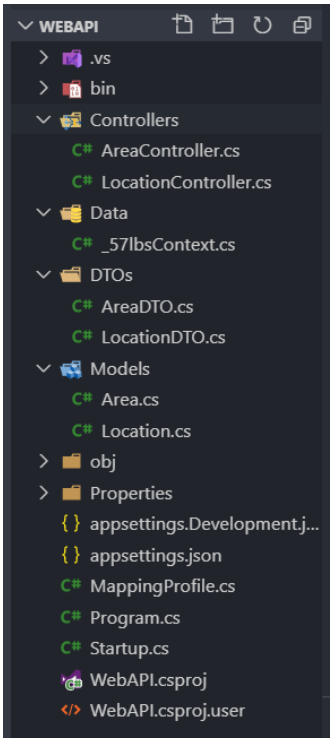



*Figura 2. Estructura del código de la API REST*

La manera de llamar a las peticiones anteriores es agregando al final de la URL la siguiente extensión: URL/api/{nombre controlador}/{id/address}.

Estas peticiones anteriormente mencionadas se encuentran dentro de los controladores existentes dentro de la API Rest, los cuales interactúan con los modelos, los cuales representan las tablas existentes en la base de datos.

Para poder resolver las peticiones hechas a la tabla location, incluyendo la tabla foránea área, se necesita hacer un mapeo de la tabla, por lo que utilizamos un data transfer object (DTO) que es un contenedor de datos para mover datos entre capas buscando así evitar redundancia y logrando de esta manera el funcionamiento correcto del sistema.

**Base de datos:**

Dentro de la base de datos AddressList, podemos encontrar dos tablas, las cuales son: Location y Área. Location contiene *address* como su llave primaria y *coordinates*, que es un campo que contiene la latitud y longitud de una dirección. Las coordenadas están guardadas en formato JSON string, para así convertirlas fácilmente a un objeto JSON que se puede utilizar en el sistema.

Área cuenta con un id único, *id_area*, una llave foránea, *address*, hacia la dirección de la tabla Location, el nombre asignada al área, el total del área en pies cuadrados, los puntos que conforman el polígono, el color con que se verá el área, y el tipo de área que es, ya sea frontal, trasera o lateral. El campo points, igual que el campo coordenadas de la tabla Location, esta guardado en formato JSON string. A continuación, se muestra el diagrama entidad-relación de la base de datos AddressList.

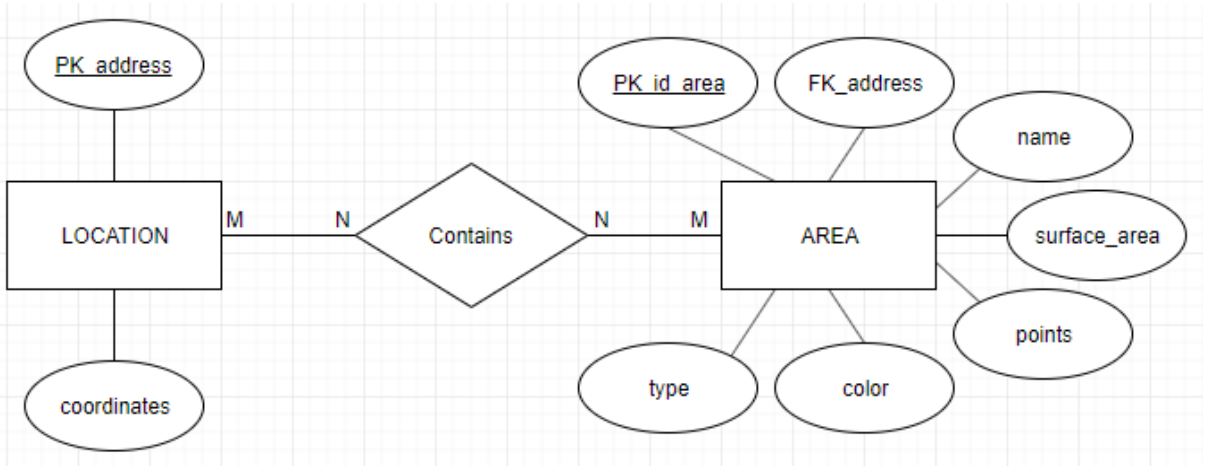



*Figura 3. Diagrama Entidad Relación de la base de datos*

| | address | coordinates |
|---|---|---|
| 1 | 1227 26th Ave, Fairbanks, AK 99701, USA | {"lat":64.823195, "lng":-147.732415} |
| 2 | 123 N White Station Rd, Memphis, TN 38117, USA | {"lat":35.134325, "lng":-89.887505} |
| 3 | 1256 Snowden Ave, Memphis, TN 38107, USA | {"lat":35.157825, "lng":-90.016677} |
| 4 | 212 Las Colinas Ln NE, Albuquerque, NM 87113, USA | {"lat":35.160715, "lng":-106.617212} |
| 5 | 2121 E Glenalden Dr, Germantown, TN 38139, USA | {"lat":35.091213, "lng":-89.760043} |
| 6 | 3150 E 23rd St, Tucson, AZ 85713, USA | {"lat": 64.823195, "lng": -147.732415} |
| 7 | 4553 Vesper Ave Sherman Oaks, CA 91403, USA | {"lat":34.154631, "lng":-118.451229} |

*Figura 4. Tabla Location*

**Interfaz con ReactJs:**

React es una biblioteca de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones de una sola página. Esta basado en componentes que son reutilizables. La interfaz del sistema web contiene 12 componentes, cada uno con sus archivos js y css correspondientes:

1. AddSurfaceArea: contiene el módulo para añadir una nueva área. Se puede editar el color y el label.
2. App: contiene todo el sistema en sí. Es el componente que pasa parámetros a todos los demás.
3. Area: contiene un area individual, con su nombre, su área en pies cuadrados, así como los botones de editar y eliminar.
4. AreaList: contiene una lista de todas las áreas individuales de una dirección.
5. EditSurfaceArea: contiene el modelo para editar un área. Se puede editar el color, la posición, la figura y el label.
6. LeftPanel: contiene todos los submódulos del panel izquierdo. Recibe parámetros de App y los envía a cada uno de los submodulos.
7. PanelContent: decide que submódulo se mostrara abajo del titulo en el panel izquierdo.
8. PanelHeader: contiene el título de la dirección buscada.
9. SearchBar: permite al usuario ingresar y buscar una dirección.
10. AddressListApi: no un componente de React en sí. Hace las peticiones a la API utilizando Axios.
11. Map: contiene todo lo relacionado con la API de Google Maps, todos los métodos utilizados para dibujar y calcular áreas.
12. MapScript: hace las llamadas a la API de Google Maps mediante una API Key.

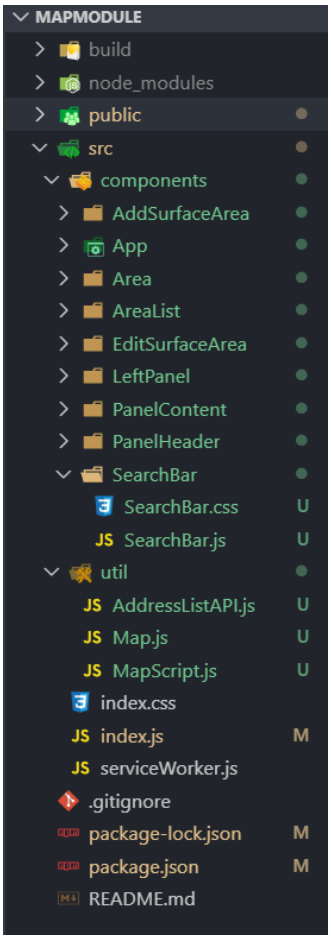
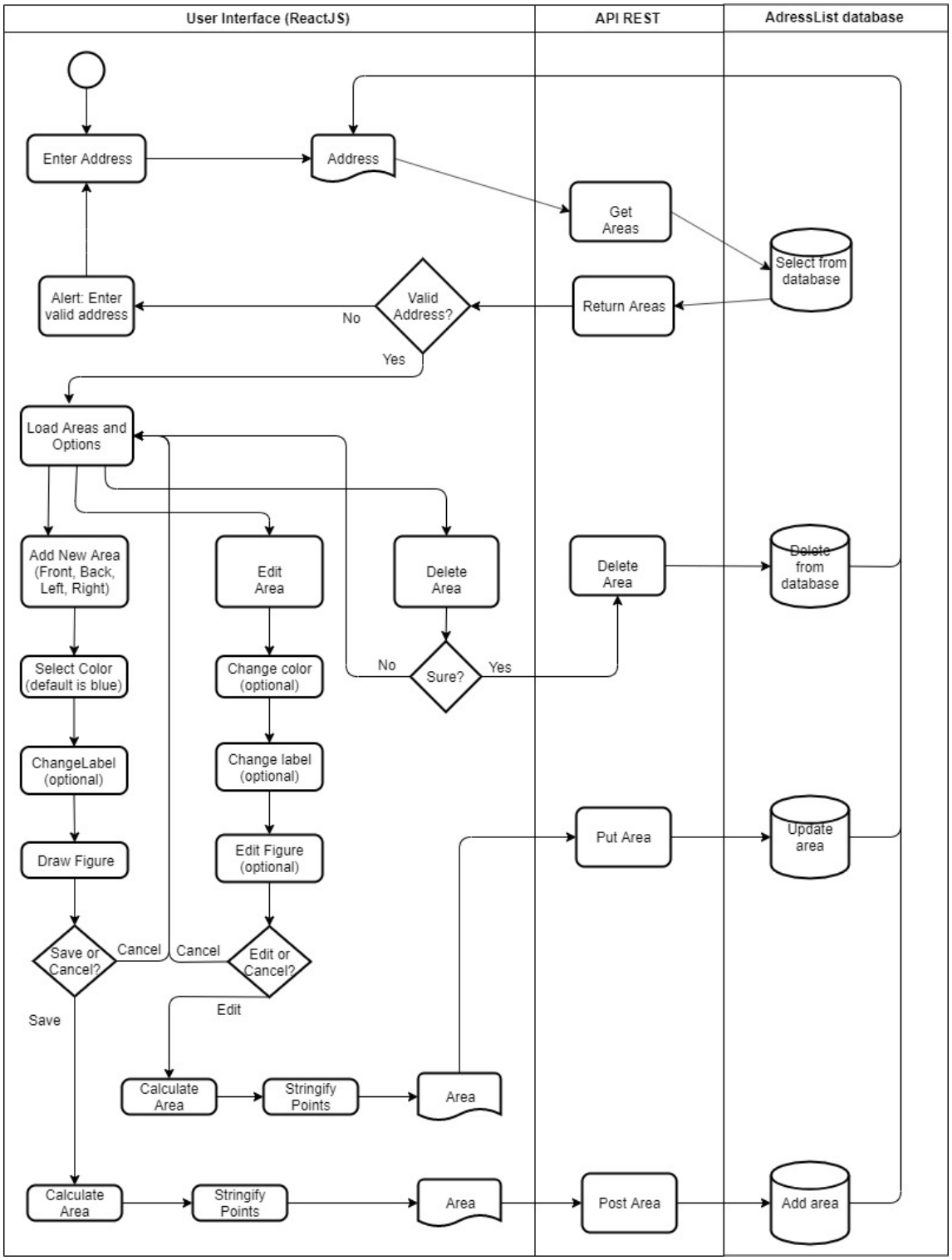



*Figura 5. Estructura del código de ReactJS*

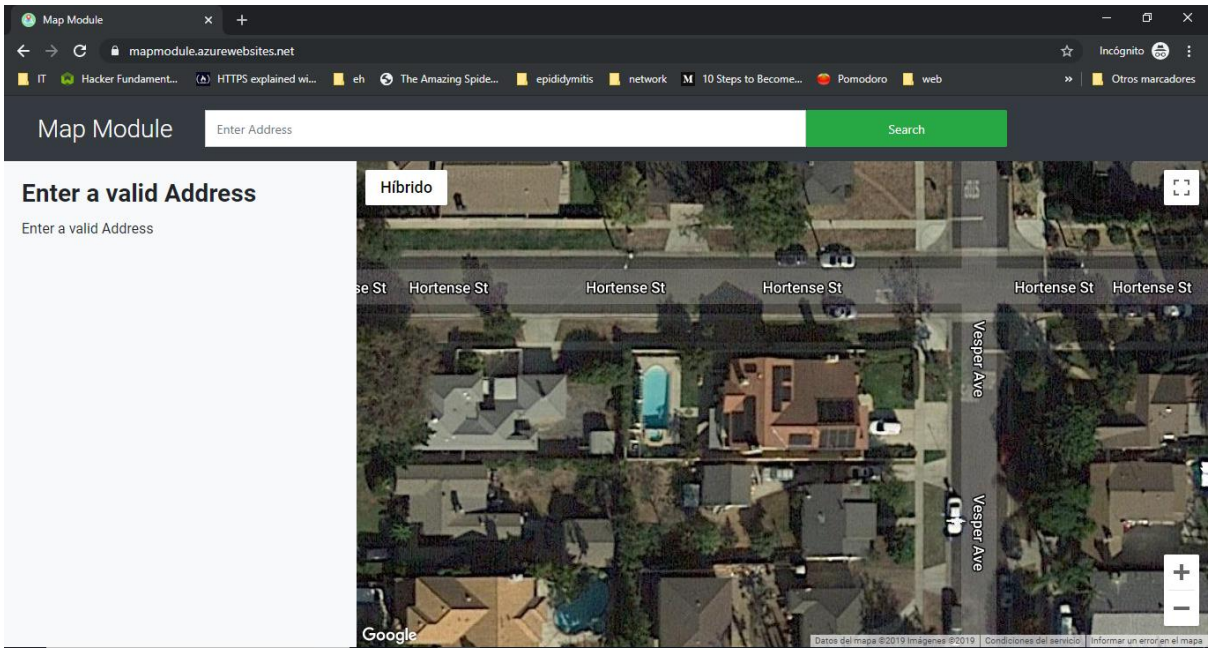
## Diagrama de flujo



# Manual de usuario

## Buscar Dirección

1. Ingresar a la dirección https://mapmodule.azurewebsites.net/ en el navegador.




2. Ingresar una dirección valida en la barra de busqueda, en el formato de dirección de Estados Unidos, y presionar buscar. La lista de direcciones validas es la siguiente (algunas direcciones ya contienen áreas dibujadas, otras no):

   1227 26th Ave, Fairbanks, AK 99701, USA

   123 N White Station Rd, Memphis, TN 38117, USA

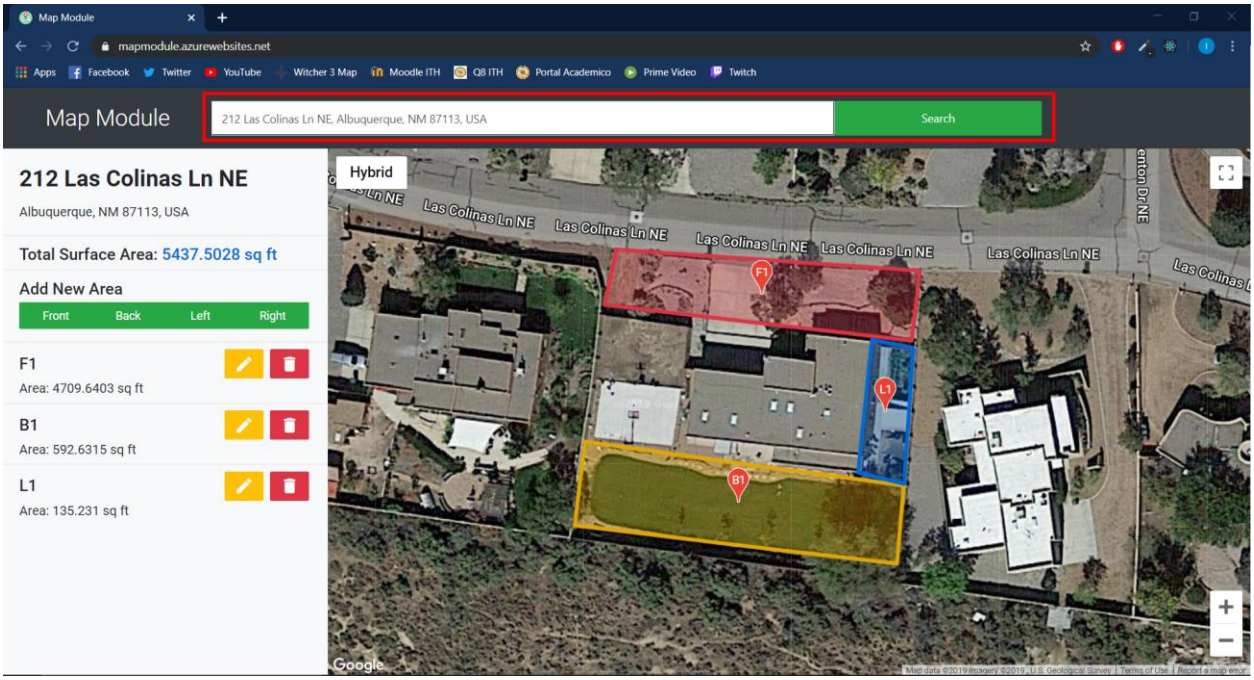
   1256 Snowden Ave, Memphis, TN 38107, USA

   212 Las Colinas Ln NE, Albuquerque, NM 87113, USA

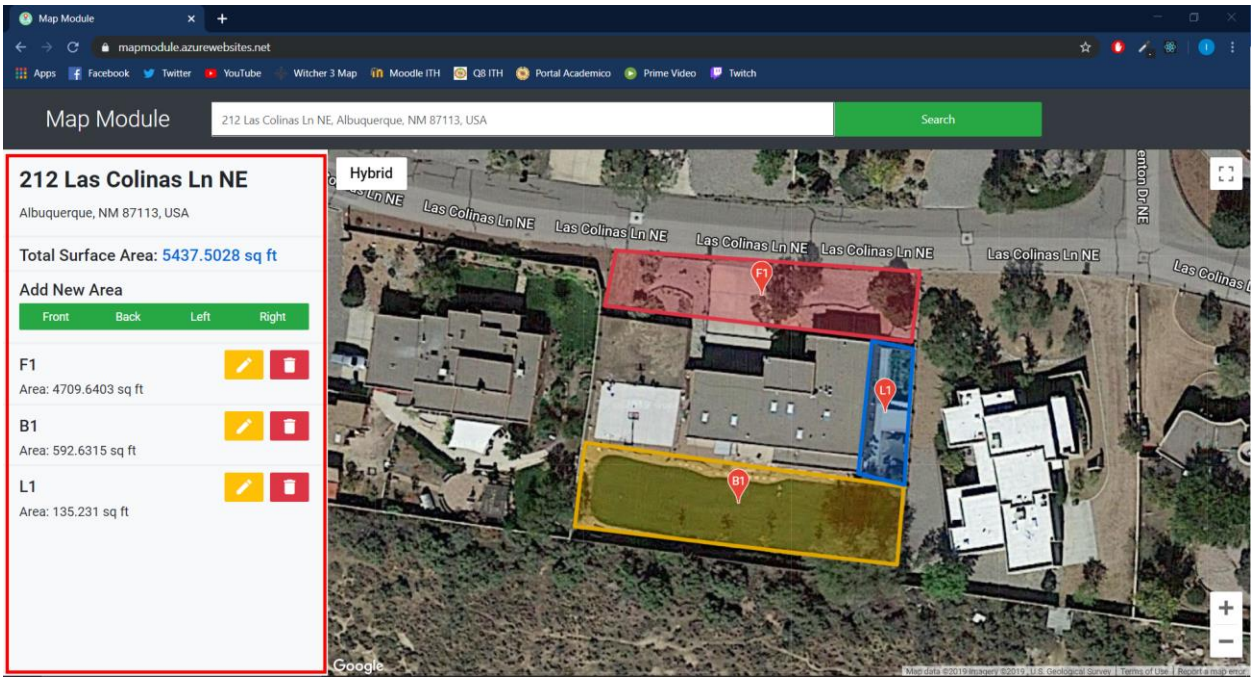
   2121 E Glenalden Dr, Germantown, TN 38139, USA

   3150 E 23rd St, Tucson, AZ 85713, USA

   4553 Vesper Ave Sherman Oaks, CA 91403, USA

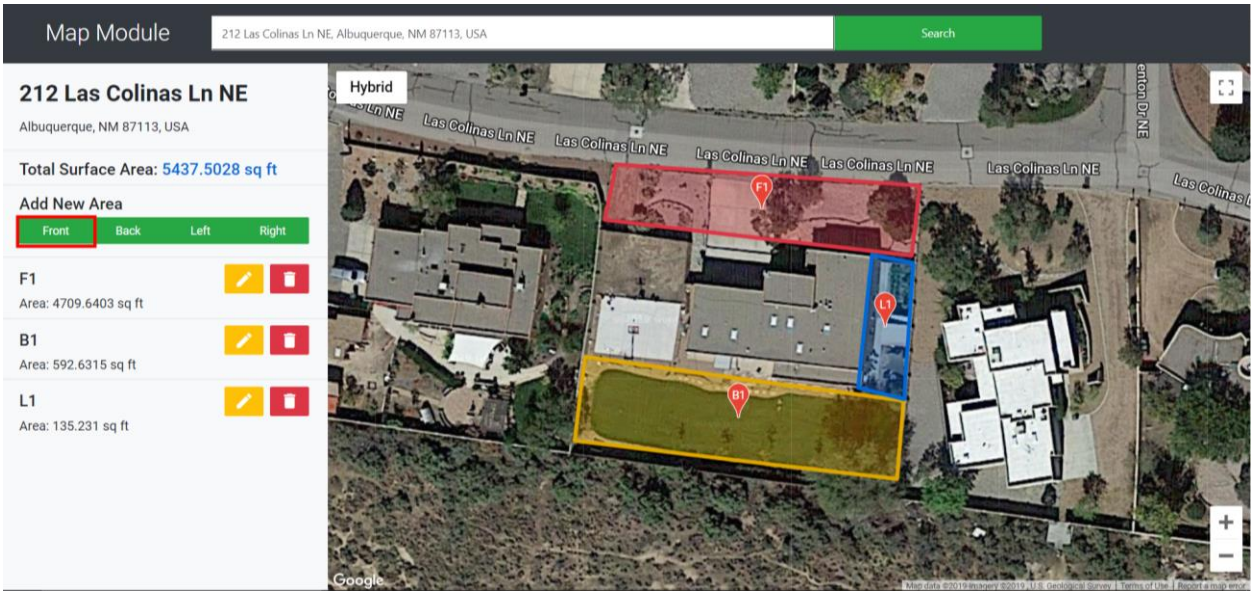



3.- Una vez buscada un área valida, el mapa nos llevara esas coordenadas y nos mostrara todas las áreas de la figura en una lista, así como el área total de la suma de todas y opciones para agregar, editar y eliminar un área.

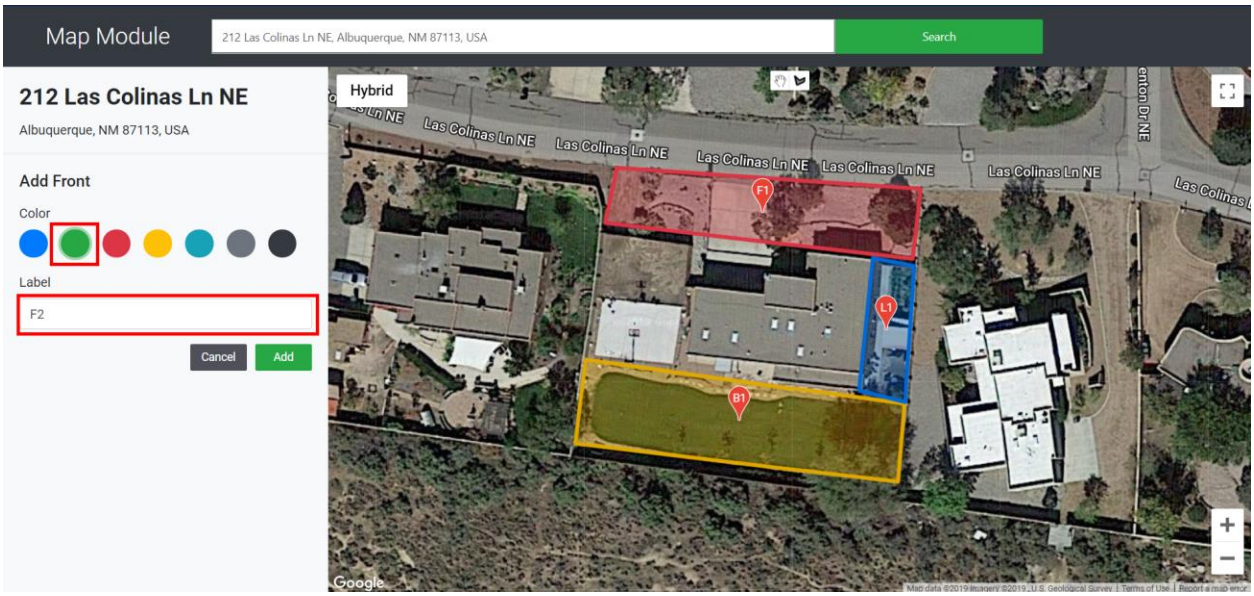


## Agregar un Área

1. Agregaremos una nueva Area. Se pueden agregar una de cuatro áreas (frente, atrás, izquierda o derecha). Hacemos click en el botón del tipo de área que deseamos agregar. En este caso seleccionamos el frente.

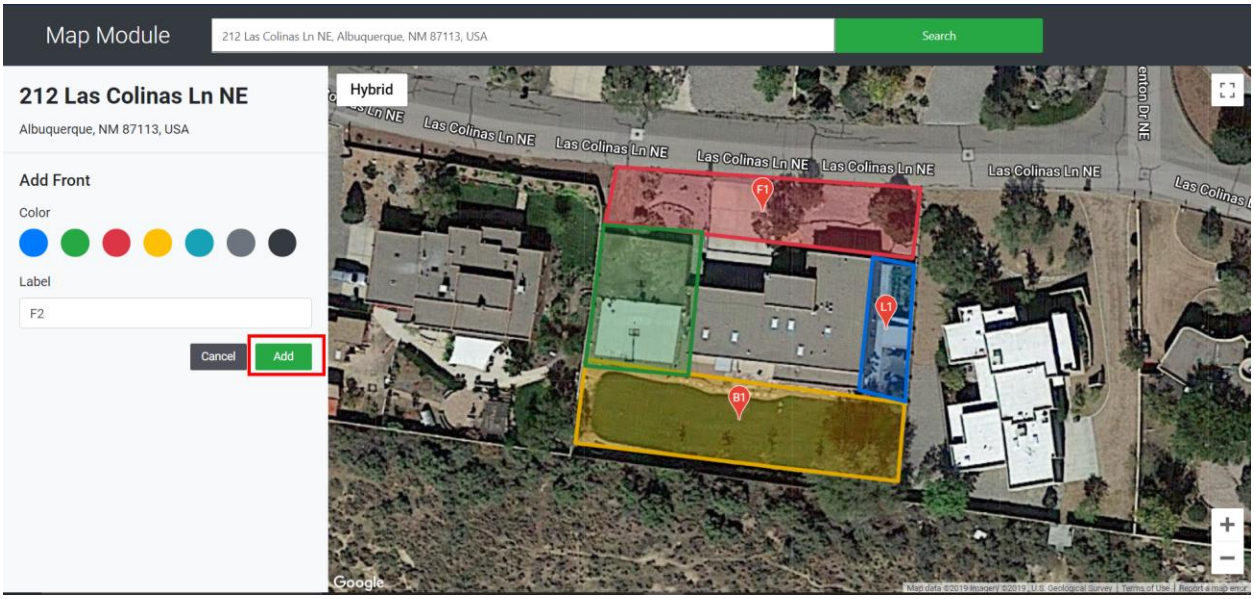



2. Seleccionamos un color y una etiqueta para el área. La etiqueta que viene por defecto es calculada basada en la cantidad de áreas del mismo tipo en esa dirección (por ejemplo, si hay 2 frentes nos sugerirá la etiqueta F3). En este caso había un frente, por eso la etiqueta sugerida es F2.

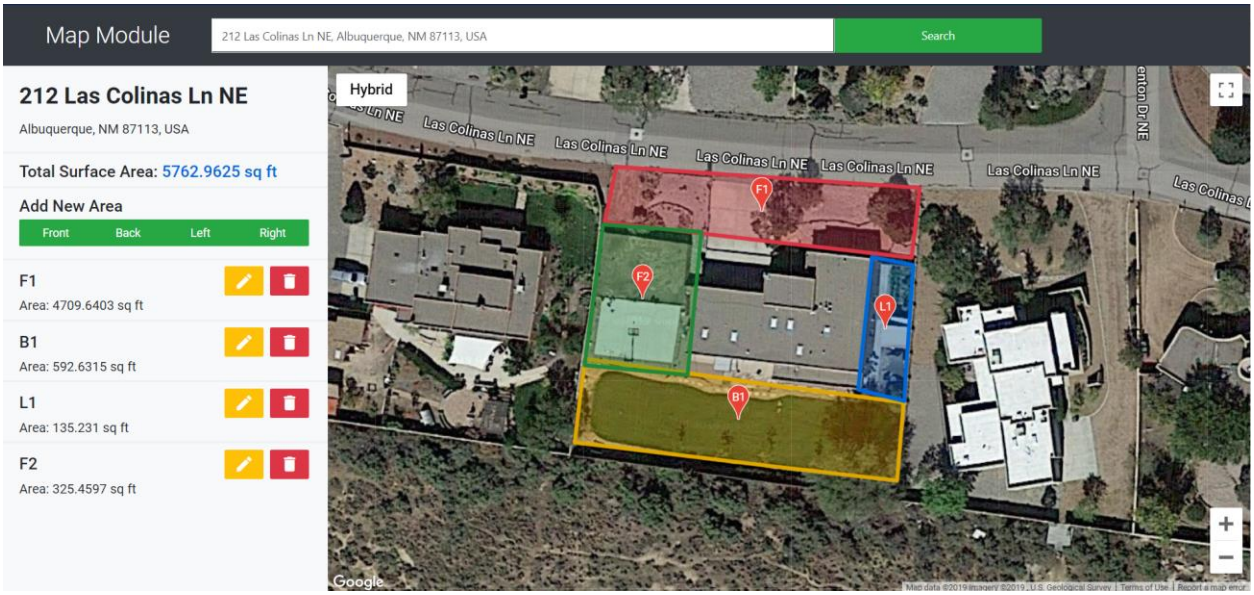


3. Después procedemos a dibujar un polígono en el mapa. Para terminar de dibujar una figura se cierra en el primer punto dibujado. (Los pasos 2 y 3 pueden realizarse en cualquier orden).


4. Ahora tenemos la opción de Agregar o Cancelar. Si cancelamos la figura nueva se borrará y volveremos a la pantalla del paso 1, sin afectar la base de datos. Si presionamos guardar, se calculará el área de esta figura nueva, se pasarán los puntos a string y se guardara en la base de datos con todos sus campos correspondientes, incluyendo el color y el label.

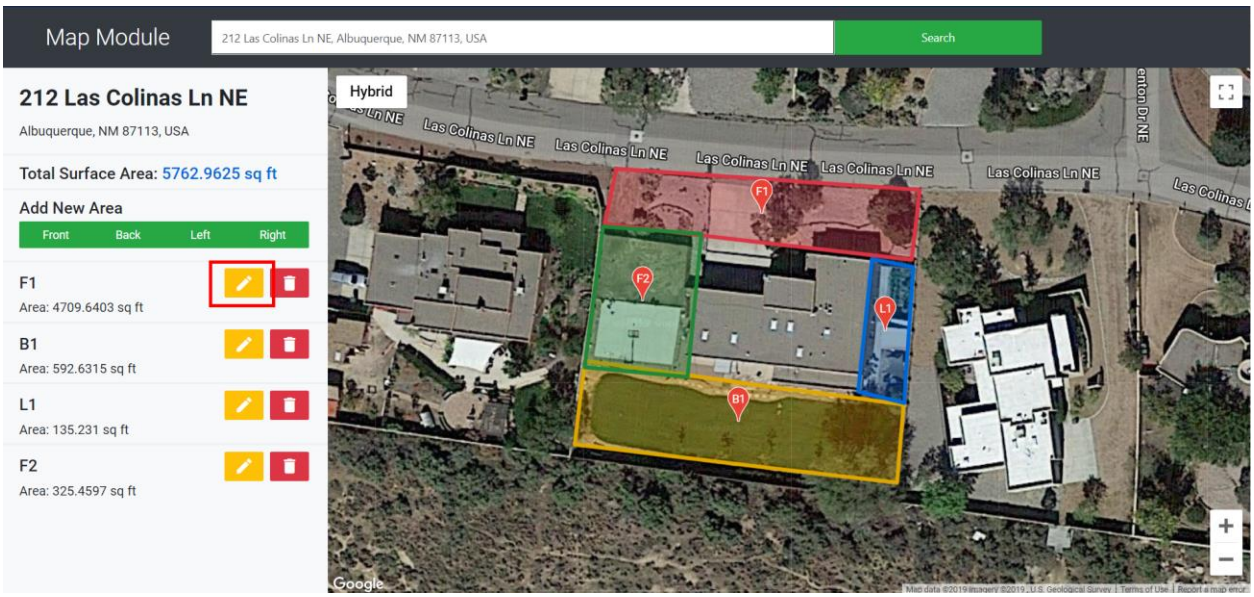


5. Al presionar Agregar, automáticamente se manda a llamar a la dirección con la que estamos trabajando para así recargar todas las figuras y que se vea reflejada la figura agregada. En la siguiente imagen podemos ver que se actualizo el mapa, así como la lista y el área total.



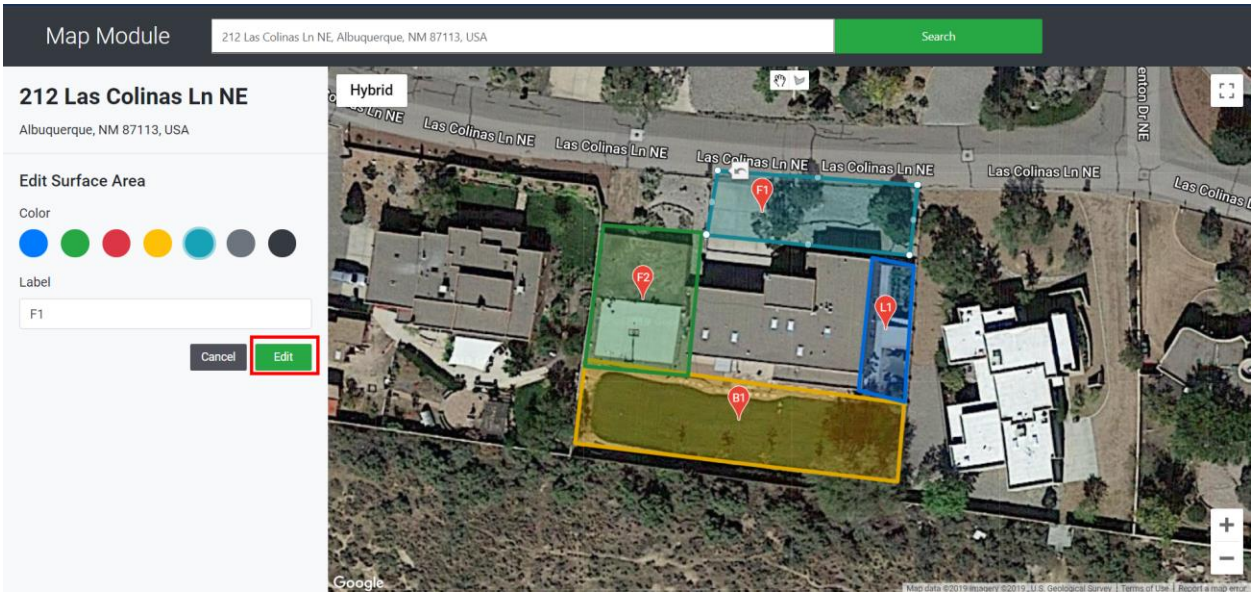

## Editar un Área

1. Para editar un área, hacemos click en el botón amarillo enseguida del área que queramos editar. En este caso editaremos el área F1.

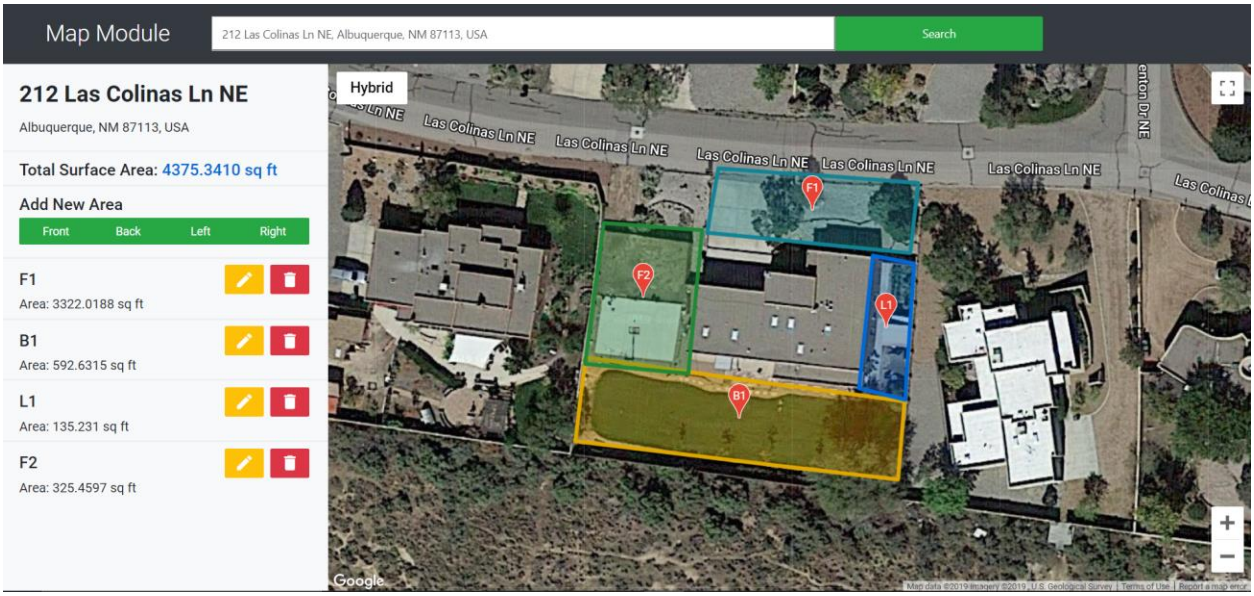


2. Una vez que estemos en el modulo de editar, podemos hacer varias cosas.
    - Cambiar el color de la figura
    - Cambiar el label de la figura
    - Editar la figura en el mapa
    - Arrastrar la figura


3. Ahora tenemos dos opciones. Podemos presionar cancelar o editar. Si presionamos cancelar se borra la figura que estábamos editando y se muestra la que estaba sin editar. Si presionamos editar se editará en la base de datos, con su color, área, etiqueta y posiciones nuevas.

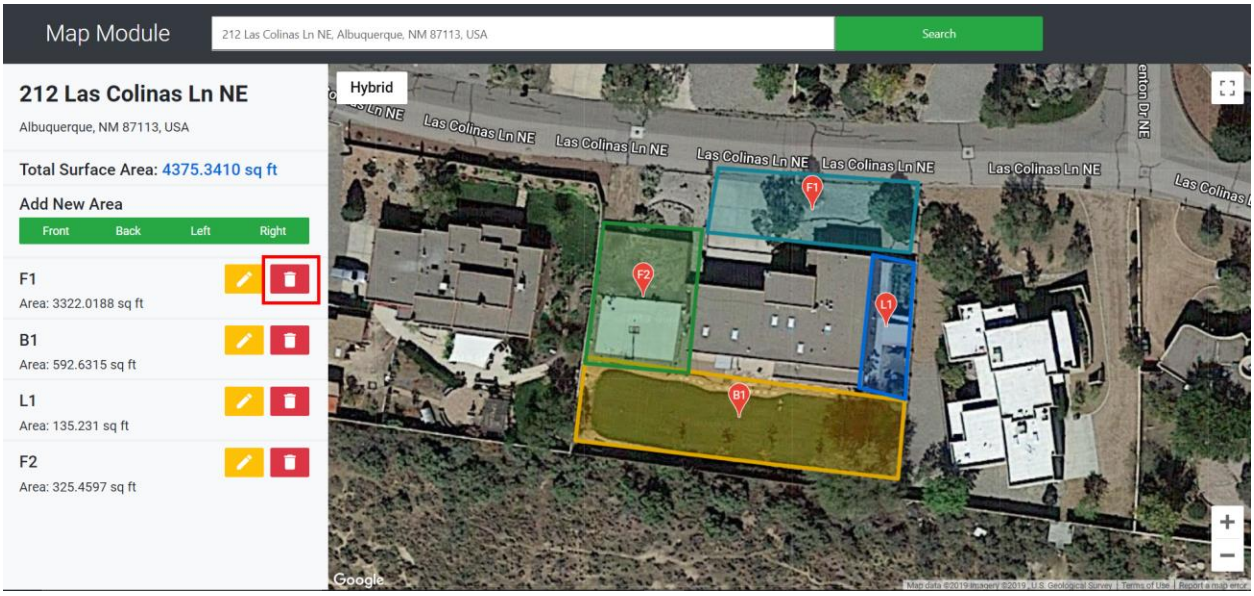


4. Al presionar editar, automáticamente se manda a llamar a la dirección con la que estamos trabajando para así recargar todas las figuras y que se vea reflejada la figura editada. En la siguiente imagen podemos ver que se actualizo el mapa, así como la lista y el área total.

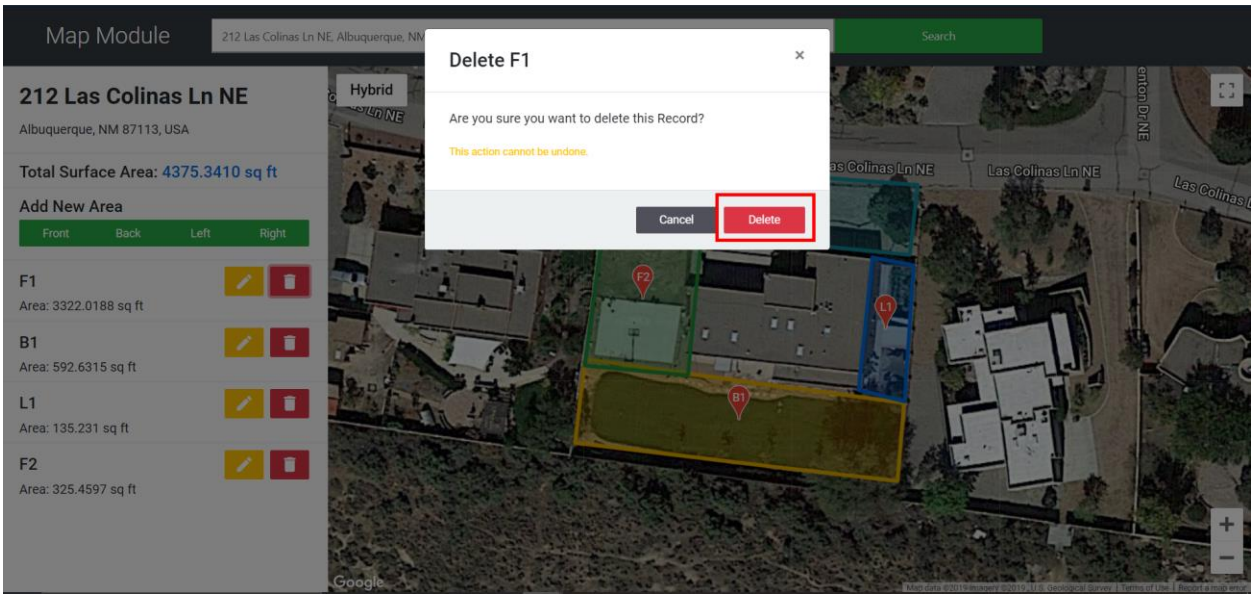



## Eliminar un Área

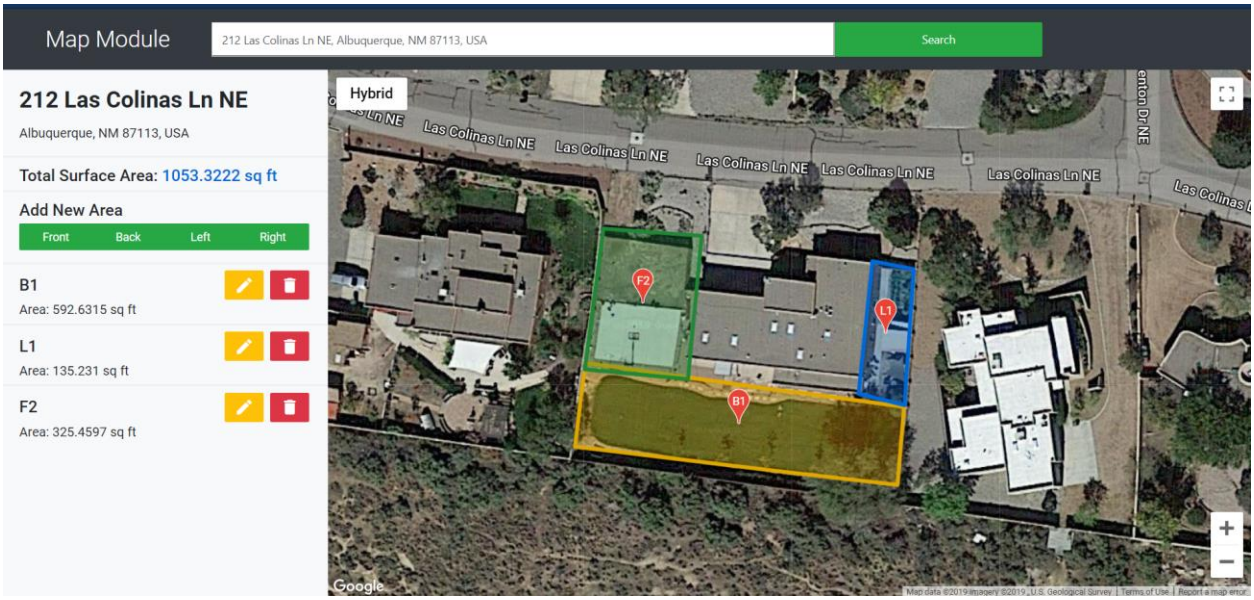
1. Para borrar un área presionamos el botón rojo que esta enseguida del área que queremos eliminar. En este caso eliminaremos F1.



2. Nos aparecerá una alerta, que nos dice que, si borramos el área, esta no se puede recuperar. Podemos cancelar y volver a la pantalla anterior o presionar el botón Delete para eliminar el área de la base de datos.




3. Al presionar eliminar, automáticamente se manda a llamar a la dirección con la que estamos trabajando para así recargar todas las figuras y que se vea reflejado que se eliminó la figura. En la siguiente imagen podemos ver que se actualizo el mapa, así como la lista y el área total.



**Referencias**

Russinovich M. (2019, Julio 15). Advancing Microsoft Azure reliability. 28 de Noviembre del 2019, de Microsoft Azure. Recuperado de: https://azure.microsoft.com/en-us/blog/advancing-microsoft-azure-reliability/

Anderson R.(2019, Julio 9). Publish an ASP.NET Core app to Azure with Visual Studio. 26 de Noviembre del 2019, de Microsoft Azure. Recuperado de: https://docs.microsoft.com/en-us/aspnet/core/tutorials/publish-to-azure-webapp-using-vs?view=aspnetcore-3.0

React(2018, Junio 26). Getting Started. 29 de Octubre del 2019, de React. Recuperado de: https://reactjs.org/docs/getting-started.html

Google Cloud.(2018, Junio 18).Google Maps Platform Service Level Agreement (SLA). 28 de Noviembre del 2019. Recuperado de: https://cloud.google.com/maps-platform/terms/sla/

Microsoft Azure.(2014, Abril 5). Pricing Calculator. 29 de Noviembre del 2019, de Microsoft Azure. Recuperado de: https://azure.microsoft.com/en-us/pricing/calculator/#app-service6e05a653-1651-46a2-a36d-e24c499625b2

GoDaddy(2016, Marzo 16). Inicia tu búsqueda de nombres de dominio. 29 de Noviembre del 2019. Recuperado de: https://mx.godaddy.com/domains/domain-name-search

Nuevoo(2016, Abril 27). Salario de Desarrollador Web Jr en México. 29 de Noviembre del 2019. Recuperado de: https://neuvoo.com.mx/salario/?job=Desarrollador%20Web%20Jr