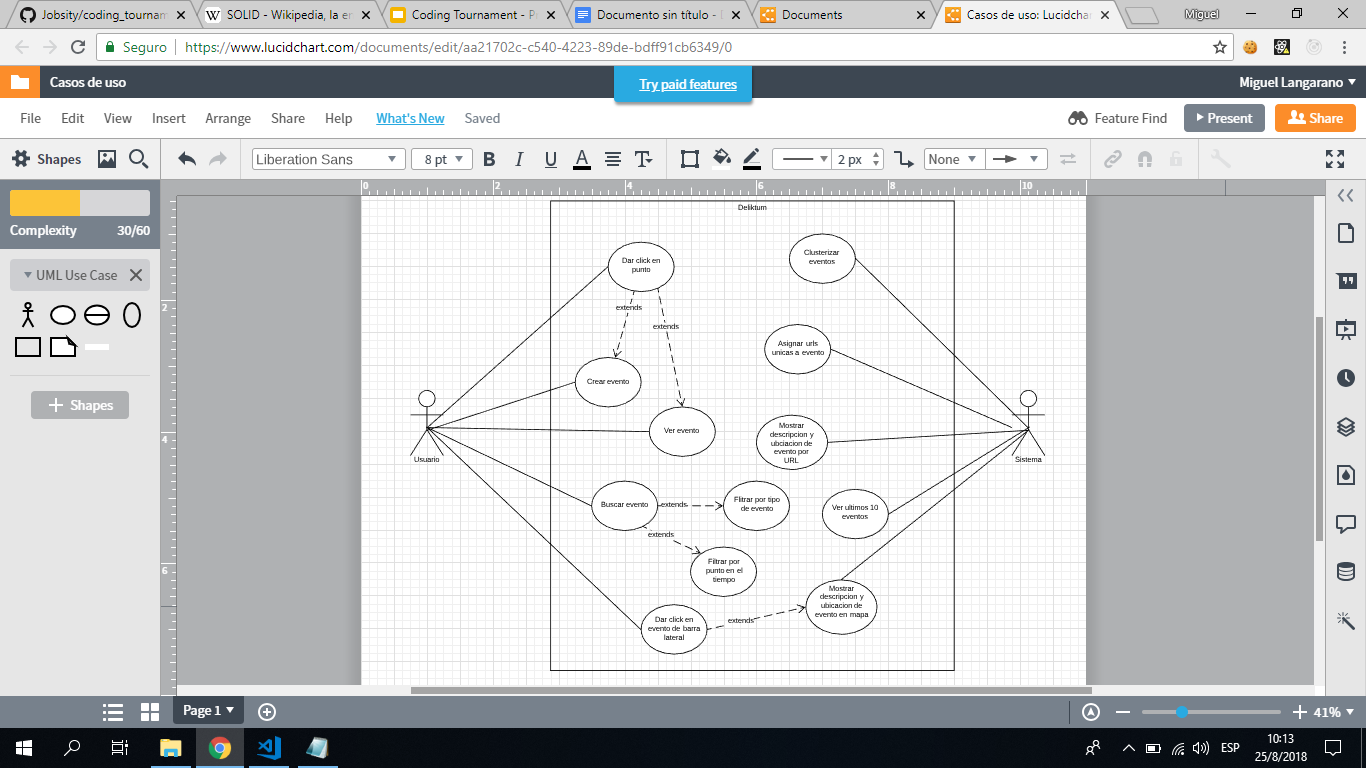
Requerimientos y Documentacion Deliktum



-Mapa donde se vean todos los eventos

-Permitir dar click en el mapa para crear o ver un evento

-Clusterizar(agrupar) dependiendo del nivel de zoom

-Asignar urls unicas a cada evento para poderlos compartir.....cuando se introduzca ese url nos debe llevar directamente

a la posicion del evento y mostrar su descripcion.

-Debe haber un boton para reportar un evento y mostrar un form para crearlo con los siguientes datos:

-Descripcion

-Tipo de evento

-Fecha y hora del evento(valor por defecto es ahora pero se puede elegir otra fecha y hora

-Ubicacion(longitud y latitud del mapa)

-Imagen correspondiente al evento

-Se debe permitir buscar un evento y filtrarlo por tipo de evento y punto el tiempo(o sea todos, hoy, semana pasada,

mes pasado, año pasado)

-Debe haber una barra lateral que muestre los ultimo 10 eventos con la opcion de cargar mas.

Si clickeo un evento se debe abrir su descripcion y ubicacion en el mapa.

--------------------------------------------------------------------------------------------------------------------------

Extras

--------------------------------------------------------------------------------------------------------------------------

-Obtener la ubicacion actual del usuario

-Usar librerias para que se vea bonito

-Que sea responsive

-Configurable para usar en otras ciudades o regiones

-Soporte multilenguaje

-Testing automatico

**“Documentacion”**

La carpeta server es donde se encuentra el backend

-Dependencias instaladas express body-parser mongoose concurrently

-Para el desarrollo instalar -D nodemon

-En la carpeta config se encuentra las keys del proyecto y en keys.js se puede cambiar el URI de la base de datos

-En models se encuentran los modelos de los objetos.

-En routes/api se encuentras las acciones que se pueden hacer con los modelos.

-En el archivo server.js se detalla la conexion con la base de datos, se manejan las keys de los servicios y se

redireccionan los pedidos al servidor, dependiendo de los modelos con los que quiere interactuar el cliente.

La carpeta client es donde se encuentra el frontend

-Dependencias instaladas axios bootstrap google-maps-react react-router react-transition-group reactstrap

-Home esta descompuesto en componentes pero es desde aqui a donde todo llega.

-En la carpeta components se encuentran todos los componentes de Home

-En home se cargan los markers para el mapa en el constructor.

-En ReportForm.js se envia a la base de datos el evento en el metodo onCreateEvent

-Los post y get se hicieron con axios

-Para iniciar el backend y frontend ir a la carpeta server y escribir en consola npm run dev

-La base de datos esta alojada en mLab y el URI se encuentras en /server/config/keys.js

-El server corre en el puerto 5000 y la pagina en el 3000

Nota: Existe un error al descargar los markers asi que hay que recargar la pagina hasta que funcione. Es un error de sincronizacion en los tiempos entre cuando se descarga y cuando se pone el mapa. No lo corregi porque ya se termino el tiempo.