**ArcGis**

La API de ArcGis posee una gran cantidad de funcionalidades, que van desde la implementación simple de mapas interactivos para la consulta de los usuarios (búsquedas, cálculos de rutas, tiempos y distancias hacia puntos en el mapa, etc.), hasta la confección de mapas personalizados, utilizando estadísticas demográficas.

En nuestro proyecto utilizaremos la funcionalidad básica de la API para implementar los siguientes servicios:

* Creación de un mapa interactivo.
* Zoom.
* Search.
* Geolocalización.
* Selección de puntos en el mapa.
* Mostrar ubicaciones de los planes del viajero en el mapa.

A continuación explicaremos cómo se implementara la API de ArcGis en nuestro proyecto.

Se utilizaran los servidores de ArcGis para el procesamiento de las consultas mediante una referencia en el HTML.

Mediante javascript se implementan las funcionalidades utilizando los módulos de la api.

En una primer instancia se debe importar los módulos necesarios para la implementación de los servicios:

require([

"esri/Map", //Módulo para crear el mapa.

"esri/views/MapView",//Módulo para crear la vista del mapa.

"esri/widgets/Locate",//Módulo para localizar ubicaciones.

"esri/widgets/Track",//Módulo para crear el widget de rastreo.

"esri/Graphic",//Módulo para crear gráficos.

"esri/widgets/Compass",//Módulo para crear la brújula.

"esri/widgets/Search",//Módulo para crear el widget de búsqueda

"esri/layers/GraphicsLayer"//Módulo para crear una capa para dibujar sobre la vista.

],

function(Map, MapView, Locate , Track, Graphic , Compass, Search, GraphicsLayer) {

**Creación de un mapa interactivo:**

Creación de mapa y configuración del tipo de visualización.

var map = new Map({

basemap:"streets-navigation-vector"

});

Creación de la vista del mapa, donde se ubica el mapa creado y se configura el punto y el zoom por defecto que va a tener al momento de cargar el mismo.

var view = new MapView({

container: "map",

map: map,

center: [ -59.136686,-37.327560],

zoom: 13

});

//La funcionalidad de **zoom** es propia de la vista del mapa.

**Search:**

Generación y ubicación en la vista del widget de búsqueda:

var search = new Search({

view: view

});

view.ui.add(search, "bottom-right");

**Selección de puntos en el mapa**

**Capa gráfica:**

Creación de una capa gráfica, donde luego se dibujarán los puntos seleccionados

var graphicsLayer = new GraphicsLayer();

map.add(graphicsLayer);

El evento retorna los atributos asociados al click y las coordenadas donde se hizo click en el mapa.

view.on("click" , function (e){...

// Se crea el objeto point con la ubicación generada por el click.

var point = {

type: "point",

longitude: e.mapPoint.longitude,

latitude: e.mapPoint.latitude

};

//Con las coordenadas se obtienen los datos para crear el objeto que contendrá los datos de la ubicación seleccionada para poder ingresarlos a la base de datos

search.clear();

if (search.activeSource) {

var geocoder = search.activeSource.locator;

var params = {

location: e.map Point

};

geocoder.location Address(params)

.then(function(response) {

var address = response.attributes.Address;

var longitude = response.location.longitude;

var latitude = response.location.latitude;

});

};

//Se crea un gráfico (un punto rojo) que se va a dibujar en lugar donde se hizo click.

graficarPunto(point , "red");

}

**Dibujar punto en el mapa:**

function graficarPunto( punto , color){

var graphic = new Graphic({

geometry: punto,

symbol: {

type: "simple-marker",

size: "12px",

color: color,

outline: {

color: "#efefef",

width: "1.5px"

}

}

});

//Se adhiere el punto creado a la capa gráfica

graphicsLayer.add(graphic);

}

**Geolocalizacion:**

Creación del widget de geolocalización

var track = new Track({

view: view,

graphic: new Graphic({

symbol: {

type: "simple-marker",

size: "12px",

color: "green",

outline: {

color: "#efefef",

width: "1.5px"

}

}

}),

useHeadingEnabled: true

});

view.ui.add(track, "top-left");

**Mostrar ubicaciones de los planes del viajero en el mapa:**

Una vez obtenidos los planes se recorre el arreglo de ubicación y se los dibuja sobre la capa gráfica.

function showPlans(){

for (var i = 0 ; i < points.length ; i++){

console.log(points[i]);

var point = {

type: "point",

longitude: points[i].longitude,

latitude: points[i].latitude

}

graficarPunto( point , "blue" );

}

};