Gobierno de Santa Fe

Ministerio de Gobierno y Reforma del Estado

Secretaría de Tecnologías para la Gestión

IDESF

Manual de uso para desarrolladores

Módulo de Visualización y Selección

Fecha de creación: 02/07/2018

Fecha de última modificación: 27/09/2022

Versión del modulo: 2.2.4

Novedades:

12/12/2018	Nuevos métodos	Se incorporaron métodos para carga de polígonos en diferentes formatos y diferentes fuentes
21/12/2018	Nuevos métodos y corrección de ejemplo	Se agregaron métodos nuevos para control de selector de capas, y agregado de capas adicionales
17/05/2019	Métodos para control de intersección de capas y mejoras	Se agregaron métodos para carga de capas vectoriales, y controles de intersección entre geometrías. Y mejoras en métodos anteriores
14/08/2019	Carga de capas vectoriales	Se permite la carga de capas vectoriales adicionales, con las cuales se puede interactuar
27/09/2022	Cálculo de áreas y exportación de polígonos seleccionados	Se permite calcular el área de los polígonos seleccionados , y exportar en formato WKT o GeoJSON

Índice de contenido

1- Introducción

El Módulo de Selección es una aplicación embebible que permite superponer objetos geográficos sobre un mapa de infraestructura base, realizar la selección de los mismos y su posterior exportación en diferentes formatos basados en estándares, siempre dentro de los límites provinciales, utilizando los servicios de la Infraestructura de Datos Espaciales de la provincia de Santa Fe (IDESF).

2- Funcionalidades

- El Módulo de ubicación permite superponer una capa vectorial, basada en filtros de selección, ya sea desde Geoserver como desde un objeto GeoJSON, sobre una capa base WMST de infraestructura Provincial.
- Permite realizar la selección directa por elemento geográfico cargado.
- Dispone de funciones de teselado de polígonos para realizar selecciones por sectores.
- Brinda la opción de exportar los elementos seleccionados en formatos estándar de intercambio de datos geográficos.
- Es embebible e interactivo.

3 - Cómo embeber el widget "Módulo de Selección"

Para hacer uso del Módulo se deben incorporar algunas librerías Javascript que permitirán la inicialización correcta del mismo. A continuación se explican los pasos a seguir para lograr su correcto funcionamiento:

3.1.- Librerías básicas

```
<script
src="https://www.santafe.gob.ar/idesf/mod-seleccion/web/js/easyXDM.min.js
" type="text/javascript"></script>
```

3.2.- Página principal del cliente

En la posición donde se desea embeber el módulo, embeber el siguiente bloque de código HTML:

```
<div id='embebido' class="fieldset" height="auto" scrolling="no"
frameborder="0"></div>
```

3.3.- Funcionalidad principal <script>

A continuación del elemento *div* anterior, insertar una sección **<script>.** En dicha sección debe ir el código JavaScript que se encarga de cargar el módulo. Éste utiliza la librería incorporada en la sección 3.1 y permite inyectar un elemento HTML -iframe- dentro del contenedor definido, realizando el intercambio de mensajes entre el cliente y el módulo:

```
<script>
var socket;
var resultado;
var datos;
var imagen;
var comunicacion;

$(document).ready(function(){

    comunicacion = {  //instancia local de XDM
                    easyXDM:
window.easyXDM.noConflict("comunicacion")
                };

    socket = new comunicacion.easyXDM.Socket({
        remote:
"https://www.santafe.gob.ar/idesf/mod-seleccion/index_iframe.php?
NAMESPACE=comunicacion", //NAMESPACE : como debe llamarse la
instancia llamada
        container:embebido,
        onReady: function(){
            this.container.getElementsByTagName("iframe")
[0].style.height = "550px";
            this.container.getElementsByTagName("iframe")
[0].style.width = "900px";
            this.container.getElementsByTagName("iframe")
[0].scrolling = "no";
            this.container.getElementsByTagName("iframe")
[0].frameborder = "0";
        },
```

```
        onMessage:function(message, origin) //capturo el mensaje
de respuesta
        {
            if(message)
            {
               //manipulo las respuestas
                if(typeof message === 'Object')
                    resultado = JSON.parse(message);
                else
                    resultado = message;

          //almaceno los resultados en variables locales al cliente
                if(resultado.imagen !== undefined)
                {
                    imagen = resultado.imagen;
                    datos = resultado.info;
                }
                else
                {
                    datos =  resultado;
                    imagen = null;
                }
            }
        }
    });
}); </script>
```

3.4.- Ejemplo práctico

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

     <title>Gobierno de Santa Fe - Buscador</title>

     <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
     <meta HTTP-EQUIV="Cache-Control" content="no-cache" />
     <meta HTTP-EQUIV="Pragma" content="no-cache" />
     <meta HTTP-EQUIV="Expires" content="0" />
        <script src="https://www.santafe.gob.ar/idesf/mod-
seleccion/js/jquery/jquery.js"></script>
        <script src="https://www.santafe.gob.ar/idesf/mod-
seleccion/web/js/easyXDM.min.js"></script>
```

```
        <link rel="icon"
href="https://www.santafe.gob.ar/favicon.ico" type="image/x-
icon"/>
        <style>
            html
            {
                background-color: rgba(255,255,255,0);
            }
        </style>
        <script>
                var coordenadas = "";
        </script>
</head>

<body style="background-color: rgba(255,255,255,0);">
<!--////////////////////// embebido -->
<h4>Versión del modulo 1.0.0</h4>
<div id='embebido' class="fieldset" height="auto" scrolling="no"
frameborder="0"></div>
<script>
    var resultado;
    var datos;
    var imagen;
    var comunicacion;
    var socket;

    comunicacion = {  //instancia local de XDM
                    easyXDM:
window.easyXDM.noConflict("comunicacion")
                };
    if(coordenadas !== '')
    {
        string_coordenadas = '&COORDENADAS='+coordenadas;
    }
    else
    {
        string_coordenadas = '';
    }


$(document).ready(function()
{
```

```
    socket = new comunicacion.easyXDM.Socket({
        remote: "<?php echo ENTORNO?>index_iframe.php?
NAMESPACE=comunicacion"+string_coordenadas, //NAMESPACE : como
debe llamarse la instancia llamada
        container:embebido,
        onReady: function(){
            this.container.getElementsByTagName("iframe")
[0].style.height = "550px";
            this.container.getElementsByTagName("iframe")
[0].style.width = "900px";
            this.container.getElementsByTagName("iframe")
[0].scrolling = "no";
            this.container.getElementsByTagName("iframe")
[0].frameborder = "0";
        },
        onMessage:function(message, origin)
        {
            //TODOS LOS MENSAJES SON ASÍNCRONOS Y DEBEN MANEJARSE
DENTRO DE ESTE CONTEXTO
            // ENVIAR MENSAJES Y REALIZAR TAREAS SÓLO LUEGO DE LA
RECEPCIÓN DE LOS MISMOS
            // PARA EVITAR PÉRDIDA DE INFORMACIÓN

            if(typeof  message === "string")
                console.log(message);

            if(message)
            {
                if(message.error !== undefined)
                {
                    console.log(message.error);
                    return;
                }

                if(typeof message === 'object')
                {
                    if(message.GeoJSON !== undefined)
                        resultado = message.GeoJSON;
                    else
                        resultado = JSON.parse(message);
                }
                else
                    resultado = message;

                if(resultado.imagen !== undefined)
                {
                    imagen = resultado.imagen;
                    datos = resultado.info;
```

```
                }
                else
                {
                    datos =  resultado;
                    imagen = null;
                }
            }
        }
    });
});

</script>

<button id="bt1" type="button">Cargar desde DB1</button><br>
<button id="bt2" type="button" >Cargar desde DB_2</button><br>
<button id="bt3" type="button" >Ajustar Vista</button><br>

<button id="bt4" type="button">Solo Selección</button><br>
<button id="bt5" type="button">Teselar Polígono</button><br>
<button id="bt6" type="button">Teselar Por Polígono</button><br>
<button id="bt7" type="button">Get Selección</button><br>

<script>
$("#bt1").on('click',function(evt)
{
   socket.postMessage({ "message":"loadPoligonos",
"capa":"idesf:scit_parcelas_query","parametros":
"NOMENCLATURA:0101PA000000112~0101PA000000113~0101PA000000734~0101
PA000000733"});

});
$("#bt2").on('click',function(evt)
{
   socket.postMessage({ "message":"loadPoligonos",
"capa":"idesf:scit_parcelas_query","parametros":
"NOMENCLATURA:0101PA000000742~0101PA000000741"});
});

$("#bt3").on('click',function(evt)
{
   socket.postMessage({ "message":"ajustarVista"});
});

$("#bt4").on('click',function(evt)
{
   socket.postMessage({ "message":"iniciarSeleccion"});
});
```

```
$("#bt5").on('click',function(evt)
{
   socket.postMessage({ "message":"teselarPoligonos"});
});

$("#bt6").on('click',function(evt)
{
   socket.postMessage({ "message":"teselarPorPoligono"});
});

$("#bt7").on('click',function(evt)
{
   socket.postMessage({ "message":"getSeleccion"});
});

</script>
</body>
</html>
```

NOTA: Ejecutar siempre el código del script dentro de un bloque $(document).ready pero teniendo la precaución de dejar como globales a las siguientes variables:

```
var resultado;
var datos;
var imagen;
var comunicacion;
var socket;
```

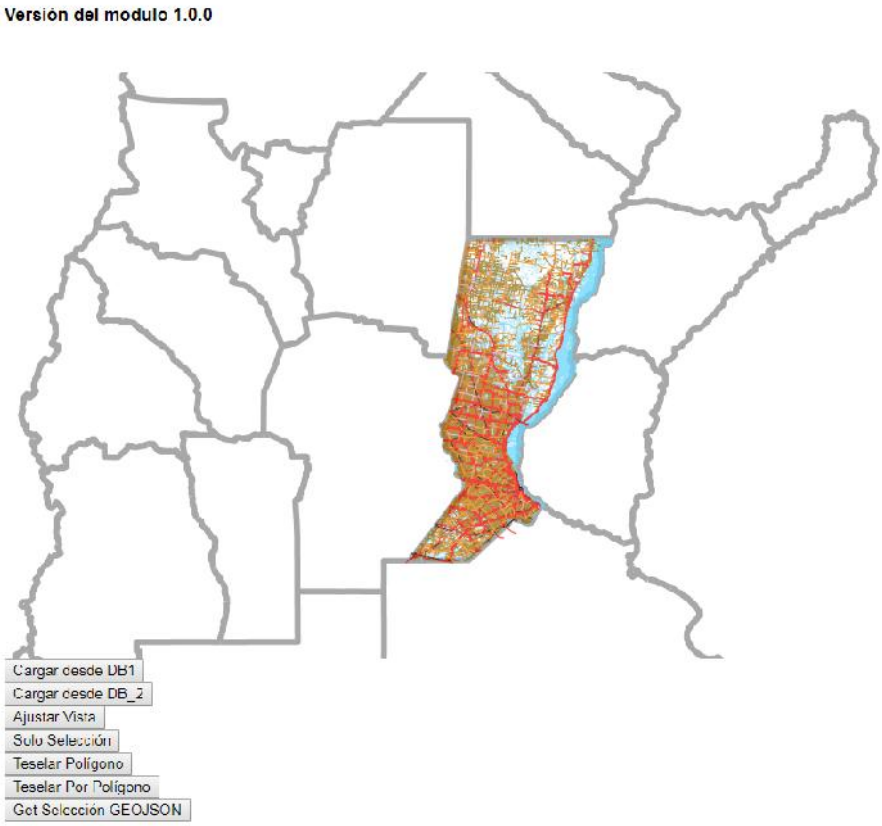
3.5.- Cambiar las dimensiones del módulo

Para modificar el tamaño del módulo, se debe modificar el valor de las siguientes líneas presentes en el script:

```
this.container.getElementsByTagName("iframe")[0].style.height = "600px";
this.container.getElementsByTagName("iframe")[0].style.width =  "600px";
```

3.6.- Uso del módulo

Una vez inicializado el módulo, se debería ver algo similar a lo siguiente:



A cada mensaje enviado utilizando el socket, el cliente que tiene embebido el módulo, recibirá un mensaje en formato String representando un objeto *JSON,* con la información solicitada o el resultado de la operación. Este objeto podrá ser procesado en la porción del cuerpo del script bajo la etiqueta **onMessage**:

NOTA: Tener en cuenta que los mensajes son siempre asíncronos.

```
onMessage:function(message, origin)
          {
               if(message)
            {
                 resultado = JSON.parse(message);
                ……….
                ……..……
            }
      }
```

3.7.- Invocación de métodos del IFRAME

Para invocar cualquier método disponible en el widget embebido se debe utilizar el socket generado al momento de la instanciación del módulo.

```
socket.postMessage({"message":"<unMensaje>"
[,"capa":"idesf:scit_parcelas_query","parametros":" < params
>"]});
```

El campo **message** es obligatorio. Los demás campos son opcionales y varían según el método utilizado.

3.8.- Cargar objetos geográficos desde Geoserver

3.8.1 - Superponer Polígonos:

Para superponer elementos del tipo polígono sobre las capas base, filtrados desde Geoserver, utilizamos:

```
socket.postMessage({
      "message":"loadPoligonos",
      "capa":"idesf:scit_parcelas_query",
      "nombreFantasia":"SCIT",
      "parametros":"NOMENCLATURA:<UN-ID-PARCELA>~<OTRO-ID-PARCELA>~….<N>"
            });
```

Donde capa y parámetros pueden tener 2 combinaciones

Combinación 1:

capa: `idesf:scit_parcelas_query`
parametros `NOMENCLATURA:<UN-ID-PARCELA>~<OTRO-ID-PARCELA>~….<N>`

Esta combinación nos permite recuperar un conjunto especifico de parcelas Catastrales concatenando los identificadores con un caracter virgulilla **“ ~ ”**

Combinación 2:

capa: `idesf:scit_parcelas_query_multi`
parametros: `NOMENCLATURA:0101P`

Esta combinación en cambio recupera toda una región completa, donde debemos pasar como parámetro los primeros caracteres que representan al distrito. Podemos seguir agregando caracteres, por ejemplo, para traer solo las parcelas rurales, agregamos la letra P luego del distrito, lo cual filtra por las parcelas rurales ya sean PA, PB , PC….

3.8.2 - Superponer Lineas: EN DESARROLLO

3.8.3 - Superponer Puntos: EN DESARROLLO

3.9.- Cargar objetos geográficos desde Objeto GeoJSON

```
socket.postMessage({
                message:"muestraPoligonos",
                GeoJSON:<el objeto GeoJSON>,
                estilo:{
                          delineado_color: "rgba(28, 0, 153, 0.5)" ,
                          delineado_grosor: 2,
                          relleno: "rgba(28, 0, 153,0.6)",
                          texto_color: "#fff" ,
                          texto_size: 12
                     }
                });
```

Ejemplo:

```
socket.postMessage({
                message:"muestraPoligonos",
                GeoJSON:resultado,
                estilo:{
                        delineado_color: "rgba(28, 0, 153, 0.5)" ,
                        delineado_grosor: 2,
                        relleno: "rgba(28, 0, 153,0.6)",
                        texto_color: "#fff" ,
                        texto_size: 12
                    }
                });
```

3.10.– Métodos Adicionales

3.10.1 – Ajustar vista

Realiza un zoom a los objetos geográficos cargados para facilitar su visualización.

```
socket.postMessage({ "message":"ajustarVista"});
```

3.10.2 – Iniciar Selección

Inicia el modo de selección por objeto geográfico.

```
socket.postMessage({ "message":"iniciarSeleccion"});
```

NOTA: Este modo habilita la selección múltiple de objetos geográficos. Para selecciones de este tipo mantener presionada la tecla Shift y hacer clic sobre los elementos deseados.

3.10.3 – Teselar polígonos / Dividir polígonos

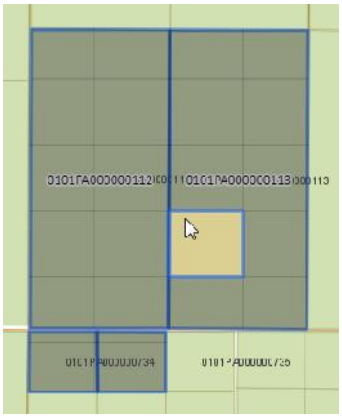
Particiona los elementos geográficos del tipo polígono en conjunto y los deja disponibles para selección por porciones.
El particionado se realiza superponiendo una grilla sobre la extensión del conjunto de polígonos, y realizando la intersección de las geometrías.

```
socket.postMessage({ "message":"teselarPoligonos"});
```

Ejemplo:

Previo al teselado

Posterior al teselado



NOTA: Este modo habilita la selección múltiple de objetos geográficos. Para selecciones de este tipo mantener presionada la tecla Shift y hacer clic sobre los elementos deseados.

3.10.4 – Teselar por polígonos / Dividir polígonos

Particiona los elementos geográficos del tipo polígono de manera individual y los deja disponibles para selección por porciones.

El particionado se realiza superponiendo una grilla sobre cada polígono de manera individual, y realizando la intersección de las geometrías.

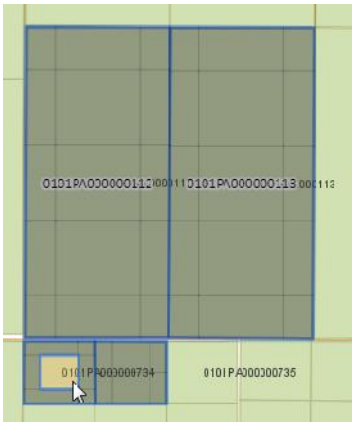
```
socket.postMessage({ "message":"teselarPorPoligono","factorCorte":8});
```

“factorCorte”: representa la cantidad de divisiones sobre el lado mas corto [mínimo :1 | máximo: 15]

Ejemplo:

Previo al teselado

Posterior al teselado



NOTA: Este modo habilita la selección múltiple de objetos geográficos. Para selecciones de este tipo mantener presionada la tecla Shift y hacer clic sobre los elementos deseados.

3.10.5 – Recuperar elementos Seleccionados

El método retorna un mensaje conteniendo el objeto GeoJSON que representa los elementos seleccionados en pantalla, o un objeto GeoJSON vacío.

3.10.5.1 – Formato GeoJSON

```
socket.postMessage({ "message":"getSeleccionGEOJSON"});
```

Respuesta tipo: (Objeto GeoJSON)


```
{"type":"FeatureCollection","features":
[{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-
61.520184,-28.081392],[-61.520184,-28.070639],[-61.507994,-
28.070639],[-61.507994,-28.081392],[-61.520184,-
28.081392]]]},"properties":{"nomenclatura":"0101PA000000112"}},
{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-
61.520184,-28.070639],[-61.520184,-28.059885],[-61.507994,-
28.059885],[-61.507994,-28.070639],[-61.520184,-
28.070639]]]},"properties":{"nomenclatura":"0101PA000000112"}}]}
```


Respuesta tipo: (sin elementos seleccionados)

```
{"type":"FeatureCollection","features":[]}
```

3.10.5.2 – Formato WKT

Con este tipo de respuesta se recuperan solo las geometrías seleccionadas, se pierden por tanto las propiedades asociadas al elemento geográfico. Para los casos en que sea necesario guardar una relación entre el vector y la información alfanumérica se recomienda utilizar el formato GeoJSON.

```
socket.postMessage({ "message":"getSeleccionWKT"});
```

Respuesta tipo: (String WKT)

```
"GEOMETRYCOLLECTION(POLYGON((-61.520184 -28.070639,-61.520184
-28.059885,-61.507994 -28.059885,-61.507994 -28.070639,-
61.520184 -28.070639)),POLYGON((-61.507994 -28.070639,-
61.507994 -28.059885,-61.50380359291716 -28.059885,-
61.5038636751898 -28.070639,-61.507994 -28.070639)))"
```

Respuesta tipo: (sin elementos seleccionados)

```
"GEOMETRYCOLLECTION EMPTY"
```

3.10.6 – Recuperar Área de los elementos seleccionados

El método retorna el área total de las parcelas seleccionadas, o la sumatoria de las porciones.

```
socket.postMessage({ "message":"getAreaSeleccion"});
```

3.10.7 – Convertir JSON a GeoJSON

El método retorna un GeoJSON conteniendo el objeto JSON, pero en el formato compatible con el método **muestraPoligonos.**

```
socket.postMessage({
                message:"JsonToGeoJSON",
                json:'< JSON con atributo geográfico >'
            });
```

Ejemplo:

```
      socket.postMessage({
                message:"JsonToGeoJSON",
               json:'[{"_id":"5c1102afcc250c0367c66e24","idAlert":3220,"nReport":
null,"type":"AC","title":"TORMENTAS FUERTES CON LLUVIAS INTENSAS.
","status":null,"date":"2018-12-
12","hour":"07:18:00","description":null,"zones":{"0":"SANTA FE:  Gral Obligado
- San Javier - Vera.","1":"CORRIENTES:  Bella Vista - Concepcion - Curuzu Cuatia
- Empedrado - Esquina - Goya - Gral Alvear - Ituzaingo - Lavalle - Mercedes -
Monte Caseros - Paso de los Libres - Saladas - San Martin - San Roque -
Sauce."},"severity":"N","polygon":"[-28.13,-59.42],[-28.34,-57.90],[-28.88,-
56.45],[-29.72,-57.20],[-30.19,-57.67],[-29.77,-60.71],[-28.91,-60.60],[-28.90,-
59.48]","urls":[{"value":"http://estaticos.smn.gob.ar/pronosticos/avisomet/
datos_aviso/nX9Y4MACqVjiIBg/avi_gral.gif","description":"gmp_general"},
{"value":"http://estaticos.smn.gob.ar/pronosticos/avisomet/datos_aviso/
nX9Y4MACqVjiIBg/aviso.gif","description":"gmp_ezeiza"},
{"value":null,"description":"cmax_240_ezeiza"},
{"value":null,"description":"gmp_pergamino"},{"value":"http://
estaticos.smn.gob.ar/radar/CMAX_PAR_Z_1_01.gif","description":"gmp_parana"},
{"value":null,"description":"gmp_anguil"},{"value":"http://
estaticos.smn.gob.ar/vmsr/goes16/
mtopnorte_24.jpg","description":"topes_nubosos"}],"region_topes_nubosos":"Sector
Norte","partial":"Y","update":null}]'
            });
```

3.10.8 – Convertir GeoRSS a GeoJSON

El método retorna un GeoJSON conteniendo el documento XML/GeoRSS, pero en el formato compatible con el método **muestraPoligonos.**

3.10.9 – Agregar capas WMS-IDESF

Permite agregar capas WMS sobre las de base, desde servicios WMS de IDESF:

Ejemplo:

```
https://aswe.santafe.gov.ar/idesf/wms?
service=WMS&version=1.1.1&request=GetCapabilities
```

```
socket.postMessage({

          message:'agregarCapasGeograficas',
          capas:{
               "listado":[
               {
                    "nombreCapa": "departamentos",
                    "nombreCapaFantasia": "Departamentos"
               },
               {
                    "nombreCapa": "catastro_regiones",
                    "nombreCapaFantasia": "Regiones"
               }
               ]}
});
```

"nombreCapa": nombre real de la capa tomada del tag <Name>
"nombreCapaFantasia": Nombre que deseamos mostrar en el selector de capas

3.10.10 – Agregar capas WFS-IDESF

Permite agregar capas vectoriales basadas en protocolo WFS sobre las básicas, desde:

https://aswe.santafe.gov.ar/idesf/wms?service=WMS&version=1.1.1&request=GetCapabilities

Ejemplo:

```
socket.postMessage({ "message":"agregarCapasVectoriales",
      "capas":{
       "listado":[

      {"nombreCapaFantasia":"buffer500","nombreCapa":"SaludVegetal:escuelas_buff
      er_500","servidor":"https://tapp.santafe.gob.ar/idesf/geoserver/
      wfs","color":"rgba(255,0,0,0.7)","width":3},
      {"nombreCapaFantasia":"buffer1000","nombreCapa":"SaludVegetal:escuelas_buf
      fer_1000","servidor":"https://tapp.santafe.gob.ar/idesf/geoserver/
      wfs","color":"rgba(255,136,0,0.6)","width":2},
      {"nombreCapaFantasia":"buffer1500","nombreCapa":"SaludVegetal:escuelas_buf
      fer_1500","servidor":"https://tapp.santafe.gob.ar/idesf/geoserver/
      wfs","color":"rgba(255,234,0,0.5)","width":2},
      {"nombreCapaFantasia":"buffer2000","nombreCapa":"SaludVegetal:escuelas_buf
      fer_2000","servidor":"https://tapp.santafe.gob.ar/idesf/geoserver/
      wfs","color":"rgba(0,191,255,0.4)","width":1}
       ]
          }
});
```

"nombreCapa": nombre real de la capa tomada del tag <Name>.

"nombreCapaFantasia": Nombre que deseamos mostrar en el selector de capas.

“servidor”: Fuente WFS desde donde se solicita la geometría.

“color”: Color de la linea perimetral, formatos validos
[#000000 | rgb(0,0,0) | rgba(0,0,0,1)]

“width”: Grosor de la linea perimetral

3.10.11 – Control de intersección entre capas vectoriales

Permite controlar si 2 capas especificadas se intersecan, y representa el resultado gráficamente sobre el mapa.

Ejemplo:

```
socket.postMessage({"message":"controlarInterseccion","nombreCapaBase":"seleccio
      n","nombreCapaControl":"buffer2000","representarEnMapa":true});
```

```
"nombreCapaBase": nombre de la capa a tomar de referencia, si el contenido es
                  selección, se utilizan las porciones seleccionadas de la capa,
                  sino toda la capa especificada.

"nombreCapaControl": nombre de la capa utilizada como parámetro de control ,
                  por ej: un buffer de 500m de una escuela

"representarEnMapa":  true muestra el resultado en el mapa
                      false solo retorna por variables


Retorno: Array con los GeoJSON obtenidos, o un array vacío.

      {"tipo":"array", "valores":resultados}
```

3.10.12 – Limpiar resultado de control de Intersección.

Elimina los polígonos superpuestos luego del control de intersección, limpia el mapa de los elementos superpuestos, NO elimina la capa cargada.

```
socket.postMessage({message:"clearControlInterseccion"});
```

3.10.13 – Mostrar selector de capas

Activa selector de Capas:

```
socket.postMessage({message:"activarSelectorCapas",estado:true});
```

Desactiva selector de capas:

```
socket.postMessage({message:"activarSelectorCapas",estado:false});
```

3.10.14 – Verificar estado de popups

Retorna el estado actual de los popups:

```
socket.postMessage({message:'popupAbierto'})
```

Retorno por mensaje: se recibe un mensaje conteniendo un String parseable a JSON:

```
'{"funcion":"popupAbierto" , "visible":false}'
'{"funcion":"popupAbierto" , "visible":true}'
```

3.11 – Múltiples instancias del modulo de selección

Inicializar el buscador de la siguiente manera:

```
comunicacion = {
      easyXDM:  window.easyXDM.noConflict("comunicacion")
                };
 //Definir variable de no-conflict para referirse a la
 //instancia del módulo

var socket = new comunicacion.easyXDM.Socket({
      .
      .
      .
      . });
```

3.12 – Exportar imagen PNG

Genera una imagen en Base64, y la retorna en un mensaje, en el ejemplo queda almacenada en una variable global llamada **imagen**.

```
socket.postMessage({message:'exporta2Png'});
```

3.13 – MUY IMPORTANTE - Recomendaciones de desarrollo

Inicializar el modulo siempre dentro de una función:

```
$( document ).ready(function() {
     <llamada al script>
});
```