

CENSO 2022

-Visualizador Web Georeferenciación en provincia de Santa Fe-

Ing. Fernando Avogradini¹, Sergio Gomez¹

¹ IPEC-Instituto Provincial de Estadísticas y Censos de Santa Fe.
Santa Fe 1950 2° Piso Rosario (S2000AUF) Tel: 341-6193167
{avogradini@yahoo.com.ar, sgomez@santafe.gov.a}

Resumen: Dos elementos: Capas y Mapas Base, presentados en un visualizador web desarrollado con Leaflet componen el sitio web: www.economiaymapas.com.ar para su uso como apoyo a los censistas del CENSO 2022 del miércoles 18 de mayo de 2022, y en la planificación del mismo en el IPEC – Instituto provincial de Estadísticas y Censos de Santa Fe –

Elegimos Leaflet.com, por ser una librería JavaScript **open source** para crear mapas web interactivos, sencilla, con una API bien documentada, que es usada por Wikipedia, OpenStreetMap, Data.gov, USGS, entre otras muchas más fundaciones y empresas internacionales, y gobiernos, y localmente por **IDERA** e **IGN**,

Las capas propias son: 1) Distrito, 2) Fracción, 3) Radio Censal, 4) Segmento 5) Donde Censa Prefectura, 6) Escuelas Sedes Censo, y están en **GeoJson** que es un formato estándar abierto de uso generalizado. Con estos DataSet, el IPEC está imprimiendo los mapas que se usaran durante el censo.

Los Mapas base son de: 1) **MapBox**: Empresa líder en uso de software Open Source para cartografía digital, 2) **Argenmap**: Mapa Digital Oficial de la República Argentina, desarrollado por el Instituto Geográfico Nacional, enmarcado en el mandato de la Ley N° 22.963 (Ley de la Carta), 3) **Google Maps**: Servidor de aplicaciones de mapas en la web más popular, 4) **Esri**: Empresa líder nivel mundial en el desarrollo de software para Sistemas de Información Geográfica, 5) **OpenStreetMap**: Proyecto colaborativo para crear mapas editables y libres.

Palabras Claves: Censo 2022, Santa Fe, Geoestadística, Leaflet

1. INTRODUCCIÓN:

Presentación del sitio:

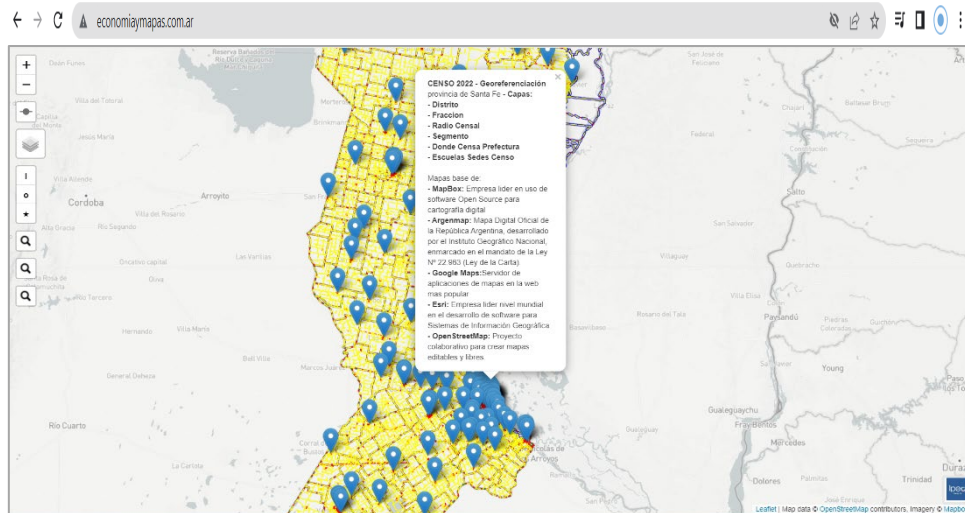


Figura 1: Pantalla de inicio del mapa web

Hoy día, las personas están habituadas a usar los mapas web como elemento de apoyo en sus actividades cotidianas, donde “Google Maps” al estar en la mayoría de los teléfonos celulares, brindando servicios de geolocalización y navegación, nos obliga a pensar en cómo usar esta cotidianidad en beneficio de la estadística.

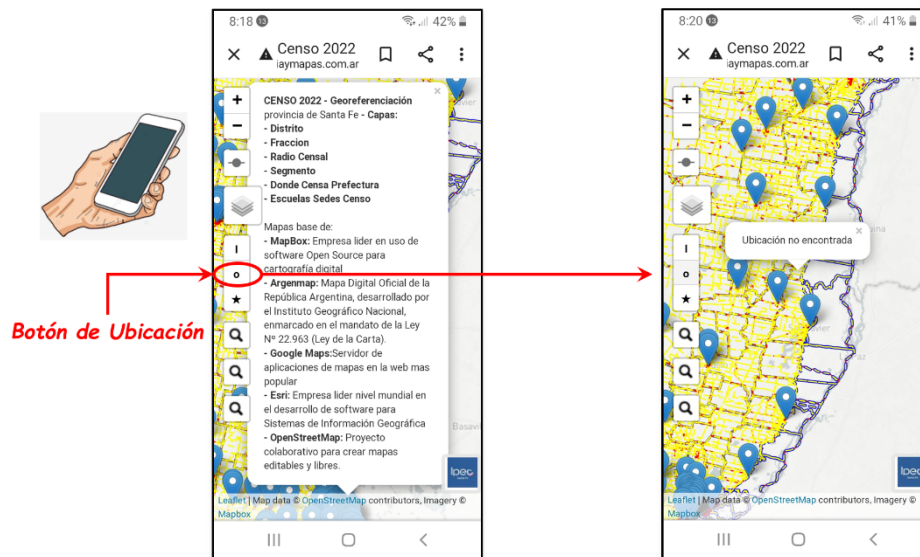


Figura 2: Pantalla de inicio en el Móvil y posterior geolocalización

Pensamos en dar un paso, y desarrollamos una página web con una estética similar, para validar aprovechando la realización del Censo Nacional 2022, mucha de la información Geoespacial que genera la provincia.

La estructura censal en Santa Fe está próxima a las 50.000 personas, que incluyen fundamentalmente a los censistas para el trabajo concreto del 18 de mayo, pero también distintos cargos que tienen que ver con jefes de departamento, de fracción, de radio.

Todos ellos necesitan saber dónde están las Sedes Censales, las Fracciones, Radios y Segmentos censales, en relación a donde ellos se encuentran o en algún otro lugar.

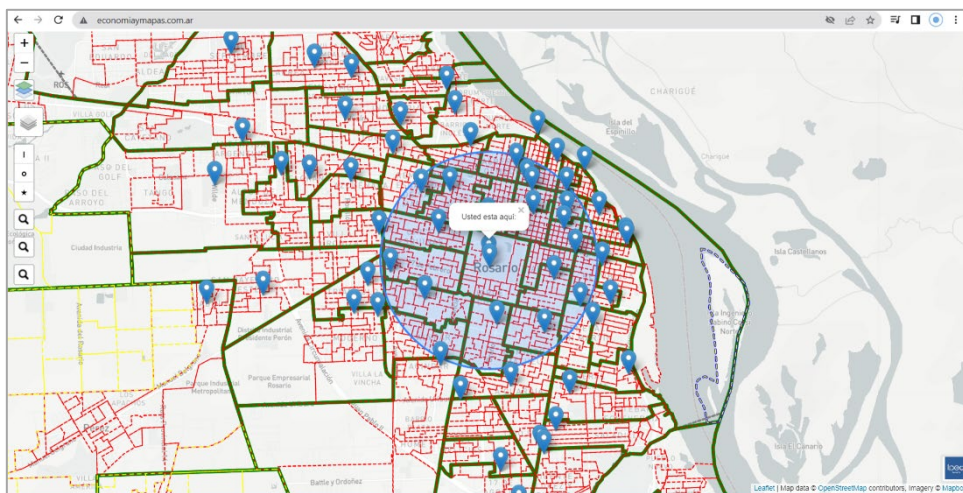


Figura 3: Geolocalización con click el Botón "Mi Ubicación"

2. PREPARANDO LA PAGINA HTML:

Antes de escribir cualquier código para el mapa, debe incluirse en la sección <head> de documento HTML los vínculos para Incluir el archivo CSS y JavaScript del Leaflet

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A==" crossorigin="" />

<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js" integrity="sha512-XQoY MqMTK8LvdXXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3ynxwA==" crossorigin=""></script>
```

Luego debe crearse en <body> un <div> elemento que contenga el mapa:

```
<div id="map"></div>
```

Hay que asegurarse de que el contenedor del mapa tenga una altura definida, configurándolo en <head>:

```
<style>
  body {padding: 0; margin: 0;}
  html, body, #map {height: 100%;}
</style>
```

Ahora inicializamos el mapa y configuraremos su vista:

```
var map = L.map('map', { zoom: 8, layers: [Gris]});
```

De forma predeterminada, todas las interacciones táctiles y del mouse en el mapa están habilitadas y tiene controles de zoom y atribución.

3. GEOLOCALIZACION:

Para acercar la vista del mapa a la nuestra ubicación tenemos el “locate” método:

```
map.locate({setView: true, maxZoom:22 })
```

Tan pronto como el usuario acepte compartir su ubicación y el navegador lo detecte, el mapa actualizará y configurará la vista.

```
function buscarLocalizacion(e) {
  var locationMarker = L.marker(e.latlng,{draggable: true}).addTo(map)
  .bindPopup("Usted esta aquí: ", {maxWidth : 200}).openPopup();
  L.circle(e.latlng, e.accuracy).addTo(map);
}

function errorLocalizacion(e) {
```

```

new L.marker(map.getCenter(),{opacity: 0,draggable: false}).addTo(map)
    .bindPopup("Ubicación no encontrada", {maxWidth : 200})
    .openPopup();
}

```

```

map.on('locationfound', buscarLocalizacion);
map.on('locationerror', errorLocalizacion);

```

Completada la geolocalización, tenemos los eventos “locationfound” y “locationerror” para configurar que hacer cuando se encuentra la localización y qué hacer cuando ocurre un error.

4. **CAPAS:**

Con este zoom ya podemos distinguir las diferentes capas, compuestas por archivos GeoJson que se encuentran en la carpeta “shapes” del directorio donde se encuentra la aplicación, las cuales hemos incorporado a nuestro documento HTML con las siguientes sentencias:

```

<script src="shapes/Distritos.js" type="text/javascript"></script>
<script src="shapes/SedesCenso.js" type="text/javascript"></script>
<script src="shapes/Radios.js" type="text/javascript"></script>
<script src="shapes/Fracciones.js" type="text/javascript"></script>
<script src="shapes/Prefectura.js" type="text/javascript"></script>
<script src="shapes/SegmentosRurales.js" type="text/javascript"></script>

```

Los objetos GeoJSON se importan al mapa a través de una capa de GeoJSON. Para crearla y añadirla al mapa, usamos el siguiente código:

```

var DistritosLayer = L.geoJSON(Distritos,{ weight: 3,opacity: 1,
    color: 'Grey', dashArray: '4',fillOpacity: 0,
    onEachFeature: onEachFeatureDistrito,
  }).addTo(map);

var FraccionesLayer = L.geoJSON(Fracciones,{ weight: 5.5, opacity: 1,
    color: 'green', fillOpacity: 0,
    onEachFeature: onEachFeatureFracciones,
  }).addTo(map);

```

```

var RadiosCensalesLayer = L.geoJSON(Radios,{ weight: 1.5, opacity: 1,
    color: 'red', dashArray: '4', fillOpacity: 0,
    onEachFeature: onEachFeatureRadiosCensales,
}).addTo(map);

var PrefecturaLayer = L.geoJSON(Prefectura,{ weight: 3.5, opacity: 1,
    color: 'blue', dashArray: '6', fillOpacity: 0,
    onEachFeature: onEachFeaturePrefectura,
}).addTo(map);

var SegmentosRuralesLayer = L.geoJSON(SegmentosRurales,{ weight: 1.5, o
    pacity: 1,
    color: 'yellow', dashArray: '6', fillOpacity: 0,
    onEachFeature: onEachFeatureSegmentosRurales,
}).addTo(map);

var SedesLayer = L.geoJSON(SedesCenso,{ radius: 8, fillColor: "#ff7800",
    weight: 1, opacity: 1, fillOpacity: 0.8,
    onEachFeature: onEachFeatureSedes,
}).addTo(map);

var TipoLayer = L.geoJSON().addTo(map);

```

Para hacer la leyenda utilizamos el complemento (plugin) “Leaflet.Legend” que muestra y alterna superposiciones. Y como con todos los complementos, deben incluirse en el encabezado del documento HTML la referencia al mismo:

```

<link rel="stylesheet" href="src/leaflet.legend.css" />
<script type="text/javascript" src="src/leaflet.legend.js"></script>

```

Luego en el <body> incluimos las siguientes líneas

```

const legend = L.control.Legend({
    title: "Censo 2022" ,
    position: "topleft",
    collapsed: true,
    symbolWidth: 24, opacity: 1, column: 1,

```

```

legends: [
  {label: "Distrito", type: "polygon", sides: 4, color: "Grey",
    dashArray: [7, 5], weight: 3.5, layers: [DistritosLayer]},
  {label: "Dpto Fracción", type: "polygon", sides: 4, color: "green",
    weight: 4.5, layers: [FraccionesLayer]},
  {label: "Dpto Frac Radio", type: "polygon", sides: 4, color: "red",
    dashArray: [7, 5], weight: 2.5, layers: [RadiosCensalesLayer]},
  {label: "Dpto Frac Radio y Seg", type: "polygon", sides: 4, color: "yellow",
    dashArray: [7, 5], weight: 2.5, layers: [SegmentosRuralesLayer]},
  {label: "Censa Prefectura", type: "polygon", sides: 4, color: "blue",
    dashArray: [7, 5], weight: 2.5, layers: [PrefecturaLayer]},
  {label: "Sedes del Censo", type: "image", url: "images/Globo.png",
    weight: 3, layers: [SedesLayer]},
]
})
.addTo(map);

```

El resultado es un botón, que se expande al pasar por arriba con el mouse mostrando la leyenda. Esta leyenda es interactiva y se usa para activar y desactivar las capas.

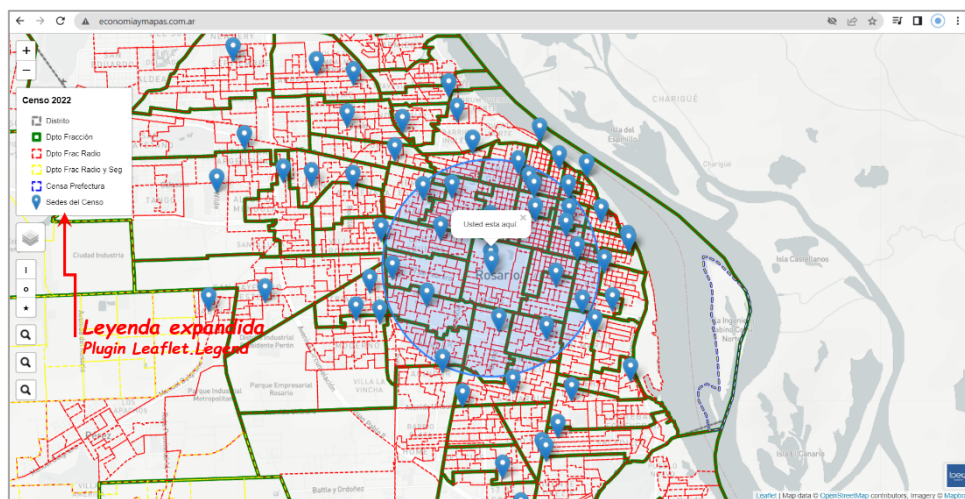


Figura 4: Capas, control de capas y leyenda

5. BUSQUEDAS:

Será necesario realizar búsquedas en los campos de los atributos de las capas (en formato GeoJSON). Para ello utilizaremos el control "Leaflet.search" el cual

deberemos descargar, así como ya lo tuvimos que hacer con el control “Leaflet.Legend” usado anteriormente, desde la página de plugins de Leaflet que está en: <http://leafletjs.com/plugins.html>

Descargado el plugin (en formato Zip), descomprimos y lo colocamos en el directorio en el que se encuentre nuestro HTML.



Figura 5: Botones de búsqueda

Luego con un `<script>` en la cabecera del HTML añadiremos la librería JavaScript y la hoja de estilos CSS de este complemento

```
<script type="text/javascript" src="src/leaflet-search.js"></script>
<link rel="stylesheet" href="src/leaflet-search.css" />
```

Luego en el `<body>` incluimos las siguientes líneas para la búsqueda por Distrito, donde la variable denominada “DistritosLayer” tiene en ella la capa de Distritos que tenemos en formato GeoJSON (en un archivo externo) y que fue definida al momento de la incorporación de la capa al mapa.

Antes de añadir el control al mapa, crearemos la función “searchControl.on” para definir el estilo del objeto geográfico resultado de la búsqueda.

```
//Boton de busqueda de Distrito
var searchControl1 = new L.Control.Search({
  layer: DistritosLayer,
  propertyName: 'NOMBRE',
```

```

textPlaceholder: 'Distrito...',
initial: false,
marker: false,

moveToLocation: function(latlng, title, map) {
    var zoom = map.getBoundsZoom(latlng.layer.getBounds());
    map.setView(latlng, zoom);
}
});

searchControl1.on('search:locationfound', function(e) {
    e.layer.setStyle({fillColor: '#00ffdd', color: '#00bbff', weight: 20, opacity: 1, fillOpacity: 0.7});
})
.on('search:collapsed', function(e) {
    DistritosLayer.eachLayer(function(layer) { //restauramos el color del elemento
        DistritosLayer.resetStyle(layer);
    });
});
});
map.addControl(searchControl1);

```

Similar para la búsqueda por Departamento, Fracción y Radio:

```

//Boton de busqueda de Radio Censal
var searchControl2 = new L.Control.Search({
    layer: RadiosCensalesLayer,
    propertyName: 'D_F_R',
    textPlaceholder: 'Depto-Frac-Radio:',
    initial: false,
    marker: false,
    autoCollapse: true,

    moveToLocation: function(latlng, title, map) {
        var zoom = map.getBoundsZoom(latlng.layer.getBounds());
        map.setView(latlng, zoom); // access the zoom
    });
});

```

```

searchControl2.on('search:locationfound', function(e) {
    e.layer.setStyle({fillColor: '#00ffdd', color: '#00bbff', weight: 20, opacity: 1, fillOpacity: 0.7});
})
.on('search:collapsed', function(e) {
    RadiosCensalesLayer.eachLayer(function(layer) { //restauramos el color del elemento
        RadiosCensalesLayer.resetStyle(layer);
    });
});
map.addControl(searchControl2);

var credctrl = L.controlCredits({
    image: "images/Logo-IPEC.jpg",
    link: "http://www.estadisticasantafe.gob.ar/",
    text: "IPEC<br/>Santa Fe"
}).addTo(map);

```

Una de las virtudes más grandes que puedes encontrar en un mapa es buscar una dirección y geolocalizarla. Y esto lo hacemos apuntando la opción “url” del control “Leaflet.Search” al proyecto “Openstreetmap”

```

// Boton de busqueda de una direccion
map.addControl( new L.Control.Search({
    url: 'https://nominatim.openstreetmap.org/search?format=json&q={s}',
    jsonpParam: 'json_callback',
    propertyName: 'display_name',
    propertyLoc: ['lat','lon'],
    marker: L.circleMarker([0,0],{radius:10}),
    marker: { icon: true },
    textPlaceholder: 'Dirección...(nº,calle,ciudad,prov,pais...)',
    autoCollapse: true,
    autoType: false,
    minLength: 2,
    zoom: 16
}) );

```

6. INSPECCIONANDO:

Con estos controles ya podemos llegar a los lugares que necesitamos inspeccionar. Veamos por ejemplo el distrito “Villa Minetti” y veamos la cartografía que produce, para este Censo el organismo encargado de hacerlo: IPEC

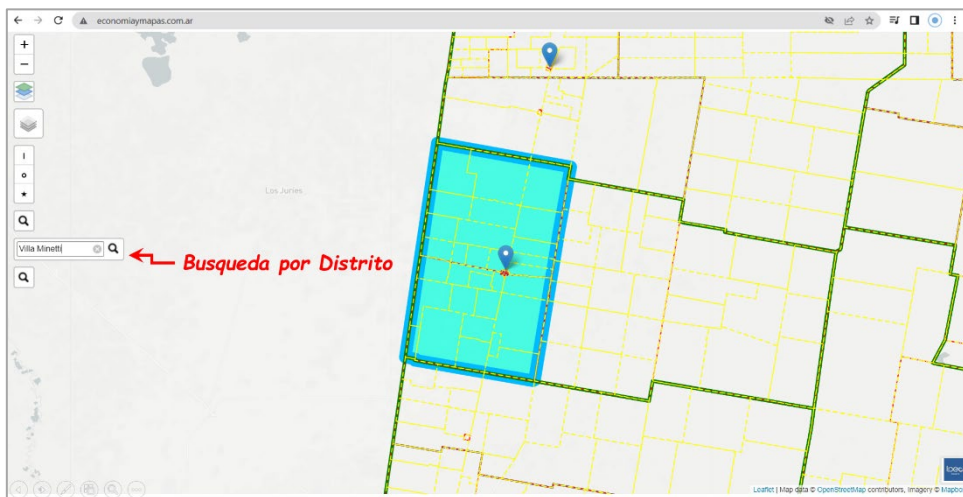


Figura 6: Localizando un Distrito

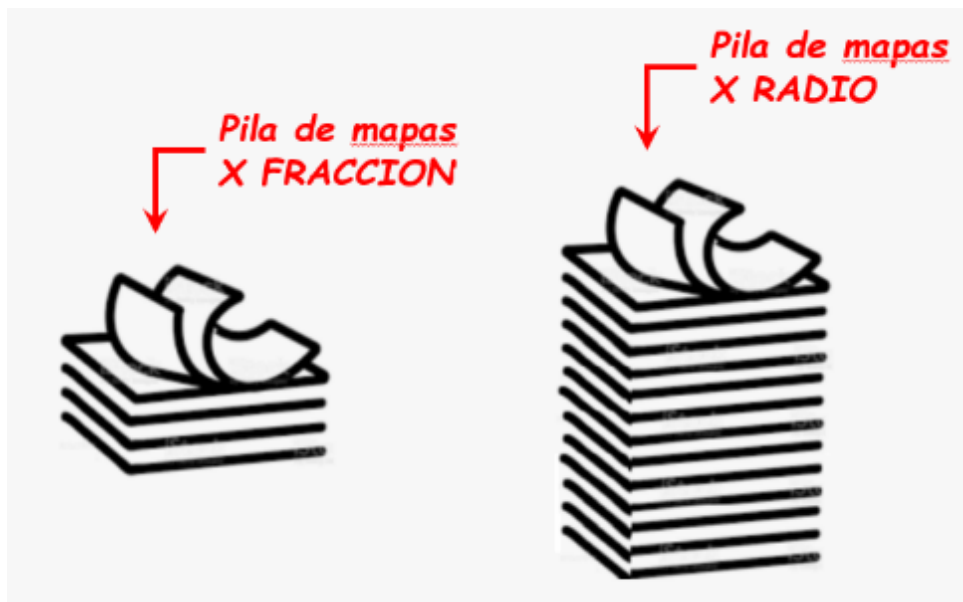


Figura 7: Donde estamos



Figura 8: Cartografía para las FRACCIONES del CENSO 2022

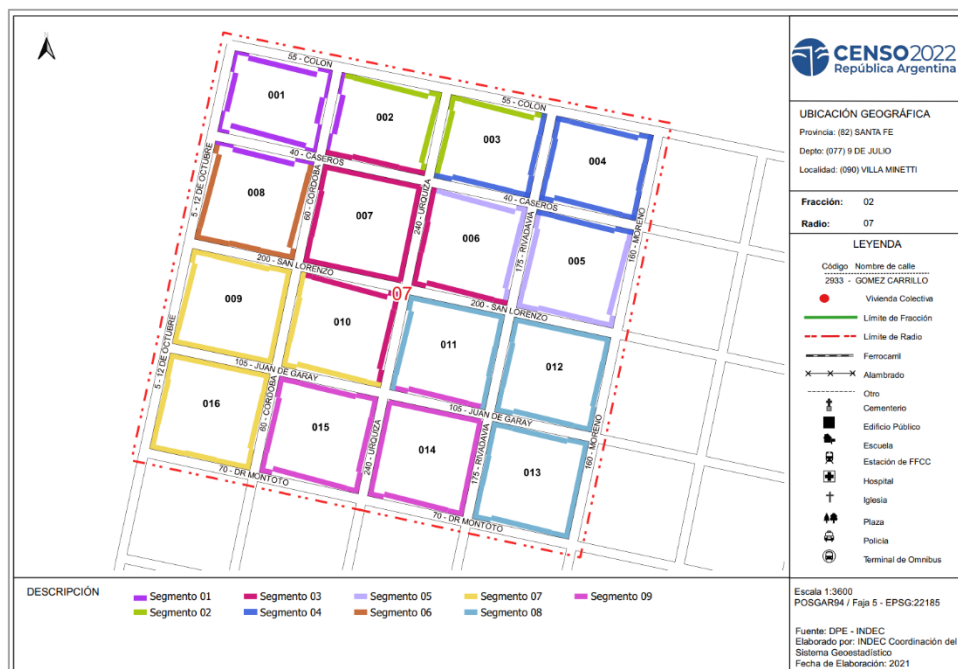


Figura 9: Cartografía para los RADIOS del CENSO 2022

Ahora que estamos en la escala de la localidad podemos a cambiar de mapa base, para ver otros detalles.

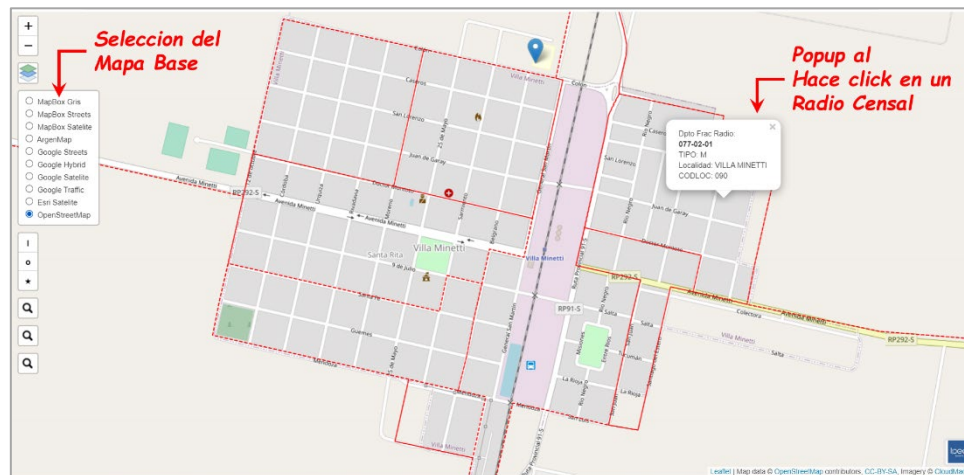


Figura 10: Con mapa base de Openstreetmap

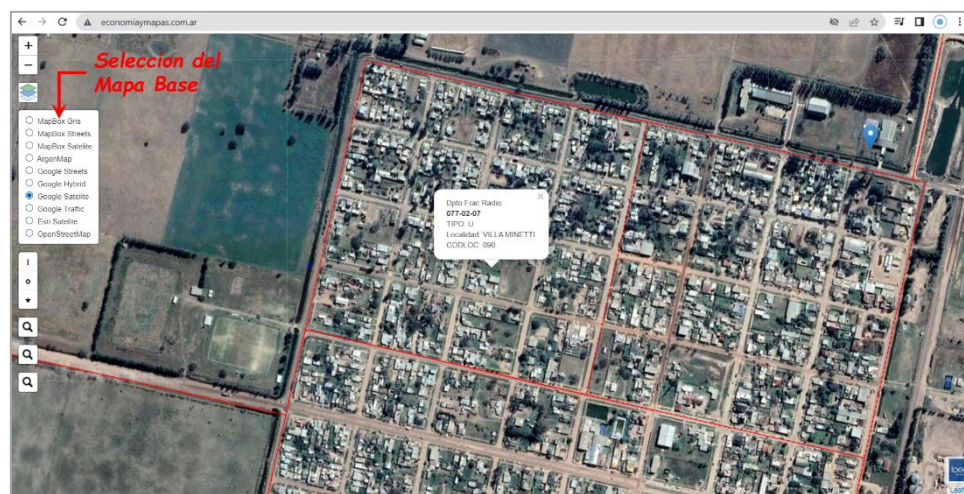


Figura 11: Con mapa base de Google Satellite

Para agregar estos servicios de mapas como mapas base hay que incorporar a nuestro HTML el siguiente código: En una variable guardamos la url y en otra variable componemos el mensaje de atribución con los créditos correspondientes.

```
<script>
```

```

var mbAttr = 'Map data &copy; <a href="https://www.openstreetmap.org/copyrig
ht">OpenStreetMap</a> contributors, Imagery © <a href="https://www.map
box.com/">Mapbox</a>';
var mbUrl = 'https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token
=pk.eyJ1IjoibWFwYm94IiwiaWUiYSI6ImNpejY4NXVycTA2emYycXBndHRqcmZ
3N3gifQ.rJcFIG214AriISLbB6B5aw';

var ignUrl = 'https://wms.ign.gob.ar/geoserver/gwc/service/tms/1.0.0/capabasea
rgenmap@EPSG%3A3857@png/{z}/{x}/{-y}.png';
var ignAttr = '<a href="http://leafletjs.com" title="A JS library for interactive map
s">Leaflet</a> | <a href="http://www.ign.gob.ar/AreaServicios/Argenmap/Int
roduccionV2" target="_blank">Instituto Geográfico Nacional</a> + <a href=
"http://www.osm.org/copyright" target="_blank">OpenStreetMap</a>';

var GoogleStreetsUrl = 'https://{s}.google.com/vt/lyrs=m&x={x}&y={y}&z={z}';
var GoogleHybridUrl = 'https://{s}.google.com/vt/lyrs=s,h&x={x}&y={y}&z={z}';
var GoogleSatUrl = 'https://{s}.google.com/vt/lyrs=s&x={x}&y={y}&z={z}';
var googleTrafficUrl = 'https://{s}.google.com/vt/lyrs=m@221097413,traffic&x={
x}&y={y}&z={z}';

var mapLink = '<a href="http://www.esri.com/">Esri</a>';
var wholink = 'i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IG
N, IGP, UPR-EGP, and the GIS User Community';
var EsriUrl = 'http://server.arcgisonline.com/ArcGIS/rest/services/World_Imager
y/MapServer/tile/{z}/{y}/{x}';

var capaOSMUrl = 'http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png';
var capaOSMAttr = 'Map data &copy; <a href="http://openstreetmap.org">Open
StreetMap</a> contributors, <a href="http://creativecommons.org/licenses/
by-sa/2.0/">CC-BY-SA</a>, Imagery © <a href="http://cloudmade.com">Cl
oudMade</a>';
</script>

```

Con el control “Leaflet.TileLayer” se cargan y muestran las capas de mosaicos en el mapa. Y se muestra el mensaje de atribución que los servidores de teselas requieren.

```

var Gris = L.tileLayer(mbUrl, {id: 'mapbox/light-v9', tileSize: 512, zoomOffset: -1,
    attribution: mbAttr});
var streets = L.tileLayer(mbUrl, {id: 'mapbox/streets-v11', tileSize: 512, zoomOff
    set: -1, attribution: mbAttr});
var satellite = L.tileLayer(mbUrl, {id: 'mapbox/satellite-v9', tileSize: 512, zoomOff
    set: -1, attribution: mbAttr});
var Argenmap = L.tileLayer(ignUrl, {tileSize: 512, zoomOffset: -1, attribution: ig
    nAttr});
var Googlestreets = L.tileLayer(GoogleStreetsUrl, {maxZoom: 20, subdomains:
    ['mt0','mt1','mt2','mt3']});
var GoogleHybrid = L.tileLayer(GoogleHybridUrl, {maxZoom: 20, subdomains:['
    mt0','mt1','mt2','mt3']});
var GoogleSat = L.tileLayer(GoogleSatUrl, {maxZoom: 20, subdomains:['mt0','
    mt1','mt2','mt3']});
var googleTraffic = L.tileLayer(googleTrafficUrl, {maxZoom: 20, subdomains:['
    mt0','mt1','mt2','mt3']});
var Esri = L.tileLayer(EsriUrl, {attribution: '&copy; '+mapLink+', '+wholink, maxZ
    oom: 18 });
var capaOSM = L.tileLayer(capasOSMUrl, {attribution: capasOSMAttr});

```

Agrupamos en una sola variable todas las conexiones a los servidores de mapas

```

var baseLayers = {
    'MapBox Gris': Gris,
    'MapBox Streets': streets,
    'MapBox Satellite': satellite,
    'ArgenMap': Argenmap,
    'Google Streets': Googlestreets,
    'Google Hybrid': GoogleHybrid,
    'Google Satellite': GoogleSat,
    'Google Traffic': googleTraffic,
    'Esri Satellite': Esri,
    'OpenStreetMap': capaOSM,
};

```

Finalmente incorporamos los mapas base con el control Leaflet.layers

```
L.control.layers(baseLayers,{position: "topleft", collapsed: true}).addTo(map);
```

7. CONCLUSIONES

Se implementó una aplicación web íntegramente con *open Source*, con la base cartográfica del IPEC -Instituto Provincial de Estadísticas y Censos- de la provincia de Santa Fe, sobre la cual se planificó el Censo 2022 del miércoles 18 de mayo de 2022.

8. AGRADECIMIENTO

Se agradece el apoyo del Área de Cartografía del IPEC –Instituto Provincial de Estadísticas y Censos de la provincia de Santa Fe

9. REFERENCIAS

Biblioteca JavaScript de código abierto para mapas interactivos aptos para dispositivos móviles: <https://leafletjs.com/index.html>, [accedido 1 abril 2022].

IGN – Instituto Geográfico Nacional. Cómo incorporar el mapa base de IGN en una página web utilizando Leaflet: <https://www.ign.gob.ar/ServiciosOGC/Leaflet>, [accedido 1 abril 2022].