



Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), julio-agosto 2025,
Volumen 9, Número 4.

https://doi.org/10.37811/cl_rcm.v9i2

APLICACIÓN SIG MÓVIL PARA EL ANÁLISIS GEOESPACIAL EN HIDALGO, MÉXICO, BASADA EN REACT NATIVE Y JSON

**MOBILE GIS APPLICATION FOR GEOSPATIAL ANALYSIS IN
HIDALGO, MEXICO, BASED ON REACT NATIVE AND JSON**

Arturo Juárez-Monroy

Universidad Autónoma del Estado de Hidalgo

Ismael Domínguez-Jiménez

Universidad Autónoma del Estado de Hidalgo

Julia Y. Arana-Llanes

Universidad Autónoma del Estado de Hidalgo

Gustavo Padron-Rivera

Universidad Autónoma del Estado de Hidalgo

DOI: https://doi.org/10.37811/cl_rcm.v9i4.19604

Aplicación SIG móvil para el análisis geoespacial en Hidalgo, México, basada en React Native y JSON

Arturo Juárez-Monroy¹juarezmonroyarturo574@gmail.com<https://orcid.org/0000-0002-4014-3589>Universidad Autónoma del Estado de Hidalgo -
Licenciatura en Ingeniería de Software
México**Ismael Domínguez-Jiménez**ismael@uaeh.edu.mx<https://orcid.org/0000-0003-0959-5875>Universidad Autónoma del Estado de Hidalgo -
Licenciatura en Ingeniería de Software
México**Julia Y. Arana-Llanes**se385809@uaeh.edu.mx<https://orcid.org/0000-0002-4986-9765>Universidad Autónoma del Estado de Hidalgo -
Licenciatura en Ingeniería de Software
México**Gustavo Padron-Rivera**gustavo_padron@uaeh.edu.mx<https://orcid.org/0000-0002-1955-3989>Universidad Autónoma del Estado de Hidalgo -
Licenciatura en Ingeniería de Software
México

RESUMEN

Los Sistemas de Información Geográfica (SIG) amplían significativamente las capacidades de visualización y análisis de datos. Los mapas personalizados son fundamentales para mejorar la comprensión visual al presentar datos específicos. Estos proporcionan una representación clara y accesible de información, permitiendo a los usuarios identificar patrones y tendencias de manera intuitiva. Este artículo describe la implementación de una aplicación móvil de SIG para el estado de Hidalgo, México, utilizando el framework de React Native. El objetivo principal de esta iniciativa es visualizar datos detallados en formato geográfico a través de representaciones visuales, empleando GeoJSON para la manipulación eficiente de los datos. La aplicación propuesta tiene como finalidad facilitar la identificación de datos de población y vivienda entre los distintos municipios del estado.

Palabras clave: Sistema de Información Geográfico, Json, React, filtros dinámicos, censos de población.

¹ Autor principal.

Correspondencia: ismael@uaeh.edu.mx

Mobile GIS application for geospatial analysis in Hidalgo, Mexico, based on React Native and JSON

ABSTRACT

Geographic Information Systems (GIS) significantly expand data visualization and analysis capabilities. Custom maps are essential for improving visual understanding when presenting specific data. They provide a clear and accessible representation of information, allowing users to intuitively identify patterns and trends. This article describes the implementation of a GIS mobile application for the state of Hidalgo, Mexico, using the React Native framework. The main objective of this initiative is to visualize detailed data in geographic format through visual representations, employing GeoJSON for efficient data manipulation. The proposed application aims to facilitate the identification of population and housing data among the state's different municipalities.

Keywords: Geographic Information System, Json, React, dynamic filters, population censuses.

Artículo recibido 20 julio 2025

Aceptado para publicación: 20 agosto 2025



INTRODUCCIÓN

El mundo actual experimenta transformaciones socioeconómicas aceleradas debido a la reconfiguración de las cadenas globales de valor. Este artículo parte del concepto de que los datos no son simples abstracciones, sino representaciones que reflejan dinámicas subyacentes de elementos reales.

Para administrar los municipios y poblados, es esencial la distribución adecuada de recursos y servicios, basada en una estrategia eficiente de recaudación de recursos. Según Hynes (2021), "una recuperación sistémica debería implicar el diseño y la gestión de un sistema socioeconómico más resiliente, donde infraestructuras, cadenas de suministro, sistemas económicos, financieros y de salud pública sean recuperables y adaptables".

Los mapas, como herramientas de representación, permiten aproximarse a la complejidad de un territorio. No obstante, su limitación en detalle puede generar incertidumbre, especialmente cuando la falta de información da lugar a interpretaciones inconclusas. El objetivo de los mapas es facilitar una comprensión más profunda del territorio para planificar transformaciones futuras basadas en los datos disponibles. El mapa convierte la inmensidad en trazos y contornos, mientras los datos se encapsulan para volverse interpretables. Ambos son herramientas cruciales, aunque en ocasiones insuficientes para un análisis completo del territorio.

La evolución tecnológica ha permitido integrar mapas en dispositivos móviles, transformándolos en herramientas dinámicas para la exploración y la toma de decisiones. La cartografía digital facilita la representación de regiones específicas y fomenta análisis espaciales profundos, conectando datos históricos y actuales. Tobler (1970) afirmaba en su primera ley de la geografía: "todo está relacionado con todo lo demás, pero las cosas cercanas están más relacionadas que las distantes".

Un Sistema de Información Geográfica (SIG) es un software diseñado para representar datos geográficos de manera visual y comprensible. Los SIG se destacan por la creación de mapas personalizados que transparentan y facilitan la comunicación de información compleja. Esto permite a los usuarios identificar patrones y manipular datos en tiempo real, aplicando filtros dinámicos para analizar diferencias.

Sin embargo, trabajar con SIG puede ser complejo y costoso. Calixto Escariz (2017) señala que, pese

a sus ventajas, los elevados costos asociados a la generación de planos y documentación geoespacial suelen derivar en información parcial o desactualizada a nivel local. El Banco Mundial (2021) sugiere que la integración segura de diversos actores en un sistema nacional de datos amplía los usos y el impacto de estos en el desarrollo, cuanto mayor es la diversidad de los participantes, más alto es el valor y el provecho que se puede obtener de los datos disponibles.

Por otro lado, Arancibia (2008) opina que el cambio en la escala del análisis geográfico no solo altera la representación de los elementos, sino también las interacciones entre ellos y la importancia de cada componente en el sistema. Además, resalta la necesidad de integrar criterios de sostenibilidad y planificación territorial desde un enfoque multidisciplinario, incorporando herramientas de Sistemas de Información Geográfica (SIG).

En Hidalgo, la ausencia de un SIG limita la creación de mapas detallados del territorio y dificulta el aprovechamiento de datos confiables, como los proporcionados por el INEGI, actualmente disponibles solo en formato de tablas de Excel. Este trabajo busca explotar esa información mediante el desarrollo de un SIG específico para el estado.

La solución propuesta permite analizar datos geográficos de manera efectiva utilizando herramientas como React Native y GeoJSON. React Native facilita el desarrollo de aplicaciones móviles funcionales y atractivas, mientras React Native Maps optimiza la creación de mapas interactivos con marcadores, polígonos y rutas. Además, el uso de GeoJSON mejora la manipulación de datos geoespaciales.

Esta herramienta busca beneficiar a diversos actores:

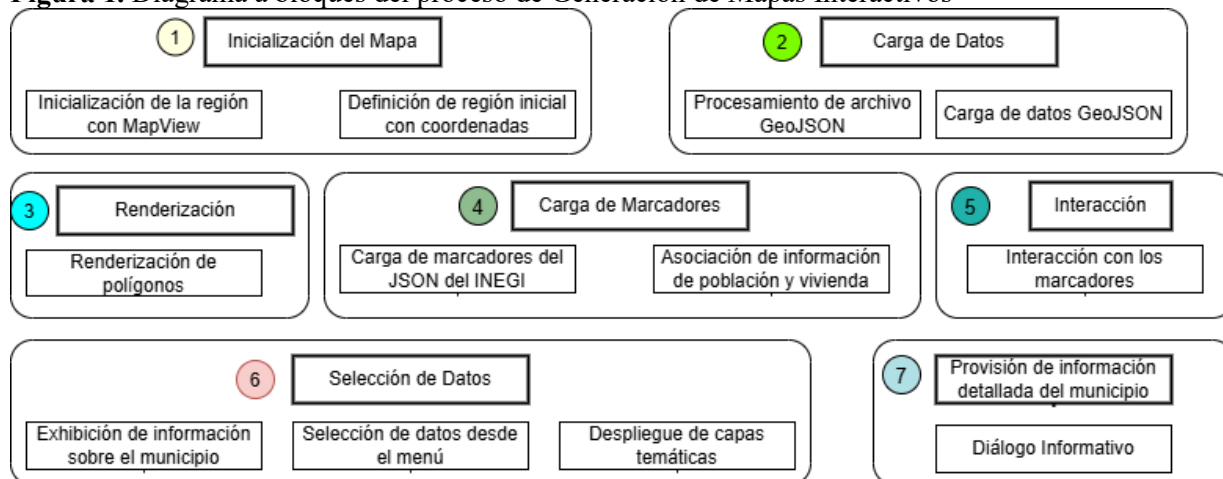
- Servidores públicos: Identificación de zonas con carencias básicas (agua potable, electricidad, internet) para implementar programas sociales eficientes.
- Investigadores: Análisis agilizado de datos geoespaciales, fortaleciendo estudios sobre desarrollo urbano, movilidad social y desigualdades económicas.
- Ciudadanos: Promoción de conciencia social, transparencia en la gestión de recursos y rendición de cuentas gubernamentales.

La aplicación también permitirá prever escenarios futuros basados en tendencias históricas, facilitando una correcta toma de decisiones y planificación.

METODOLOGÍA

El desarrollo del SIG sigue una metodología precisa para integrar información geoespacial de forma interactiva, utilizando JSON, React Native y React Native Maps. El proceso incluye pasos clave para generar mapas dinámicos y optimizar el análisis territorial (ver Figura 1).

Figura 1. Diagrama a bloques del proceso de Generación de Mapas Interactivos



Fuente: Elaboración propia.

El proceso para generar mapas interactivos se compone de siete pasos principales, los cuales se describen a continuación (ver Figura 1):

- 1. Inicialización de una región con la librería MapView:** Se define la región inicial del mapa con coordenadas específicas, lo que permite centrar el mapa en el área de interés.
- 2. Carga de datos GeoJSON:** Se procesa un archivo GeoJSON que contiene las coordenadas de los polígonos que representan los municipios del estado de Hidalgo.
- 3. Renderización de los polígonos:** Los polígonos son dibujados en el mapa utilizando las coordenadas procesadas.
- 4. Carga de marcadores e información del JSON del INEGI:** Se agregan marcadores al mapa para cada municipio y se asocia información relevante de población y vivienda extraída del JSON.
- 5. Interacción con los marcadores:** Al presionar un marcador, se muestra la información solicitada sobre el municipio en función de las variables seleccionadas.
- 6. Selección de datos desde el menú:** Se despliega una lista con las capas temáticas disponibles, como "Población" y "Vivienda".

7. Muestra de cuadro de diálogo: Se proporciona información detallada acerca del municipio seleccionado, presentada de manera visual y accesible.

Estos pasos permiten generar cualquier mapa solicitado que muestre información estadística basada en los datos extraídos de archivos Excel.

Transformación de datos

Los archivos JSON son esenciales para una manipulación eficiente de datos en JavaScript, gracias a su estructura similar a los objetos de este lenguaje. Esta similitud facilita la integración y gestión de información entre aplicaciones y servidores.

En este contexto, convertir un archivo Excel a JSON es un paso crítico, ya que JSON es un formato ligero, estructurado y altamente eficiente para entornos móviles. Mientras que Excel organiza los datos en filas y columnas, JSON transforma esa información en una estructura jerárquica, optimizada para consultas y manipulaciones dinámicas. La conversión de Excel a JSON permite que los datos sean accedidos rápidamente y visualizados de manera dinámica, sin la necesidad de procesar archivos tabulares complejos.

Dos archivos JSON son fundamentales para el funcionamiento de la aplicación:

1. El primer archivo contiene los datos necesarios para alimentar el SIG, basados en el censo de población y vivienda de 2020 proporcionado por el INEGI. Estos datos fueron extraídos en formato Excel, procesados y organizados antes de ser convertidos a JSON.
2. El segundo archivo traza un mapa del estado de Hidalgo, delimitando cada municipio mediante polígonos.

Durante el procesamiento, se identificaron registros incompletos en algunos municipios. Los valores faltantes fueron reemplazados por ceros para garantizar la consistencia de los datos.

Para convertir los datos de Excel a JSON, se utilizó la librería SheetJS, una herramienta de JavaScript que facilita la lectura y manipulación de hojas de cálculo como .xlsx y .xls. Esta librería permite extraer y procesar datos tabulares de manera eficiente, directamente desde el navegador.

En la Figura 2 se presenta el código encargado de la conversión del archivo Excel a JSON,

Figura 2. Script conversión de *Excel* a *JSON*

```
27     const reader = new FileReader();
28     reader.onload = function(event) {
29         const data = event.target.result;
30         const workbook = XLSX.read(data, { type: 'binary' });
31         const sheetName = workbook.SheetNames[0];
32         const sheet = workbook.Sheets[sheetName];
33         jsonData = XLSX.utils.sheet_to_json(sheet, { header: 1 });
34         document.getElementById('output').textContent = JSON.stringify(jsonData, null, 2);
35         document.getElementById('downloadBtn').style.display = 'inline';
36     };
37     reader.readAsBinaryString(file);
38 }
```

El archivo JSON resultante del proceso de conversión, ilustrado en la Figura 3, constituye la base de datos geoespacial y estadística fundamental para la aplicación. Este archivo, generado a partir de los datos originales del INEGI, contiene información estructurada que permite tanto la geolocalización precisa de los municipios como el despliegue de indicadores sociodemográficos relevantes.

Figura 3. Script. Archivo del INEGI en formato JSON.

```
1  [
2      {
3          "MUN": 1,
4          "NOM_MUN": "Acatlán",
5          "LOC": 0,
6          "POBTOT": 22268,
7          "P_18YMAS": 14467,
8          "P18YM_PB": 3215,
9          "GRAPROES": 7.35,
10         "PEA": 11462,
11         "PE_INAC": 5568,
12         "POCUPADA": 11368,
13         "PDESOCUP": 94,
14         "PSINDER": 9798,
15         "VIVTOT": 8021,
16         "TVIVHAB": 5870,
17         "TVIVPAR": 8010,
```

En cuanto al proceso de conversión, las funciones implementadas realizan las siguientes operaciones: en la línea 30, la función `XLSX.read` procesa el archivo de Excel almacenado y lo transforma en un objeto `workbook` que contiene todas las hojas del documento original. Posteriormente, en la línea 31, `SheetNames` recupera un arreglo con los nombres de todas las hojas disponibles en el libro de trabajo. La línea 32 se encarga de seleccionar específicamente la primera hoja de cálculo mediante su identificador nominal. Finalmente, en la línea 33, la función `sheet_to_json` realiza la conversión propiamente dicha, transformando los datos tabulares en un objeto JSON donde los encabezados de

columna se convierten en claves para la estructura de datos resultante.

El archivo JSON generado integra un conjunto comprehensivo de variables asociadas a cada municipio, entre las que destacan: datos poblacionales desagregados por grupos etarios; indicadores educativos como el grado promedio de escolaridad; información sobre la población económicamente activa; características de vivienda que incluyen acceso a servicios básicos como agua potable, energía eléctrica y drenaje; equipamiento doméstico que comprende electrodomésticos, vehículos automotores y conectividad a internet; variables laborales como ocupación, desempleo y distribución por sectores económicos; así como indicadores de acceso a servicios de salud. Esta estructura de datos integral facilita el análisis geoespacial y la visualización de indicadores sociodemográficos en la aplicación móvil desarrollada.

Uso de GeoJSON para Representación Geoespacial

La delimitación precisa de fronteras geográficas representa un proceso técnicamente complejo y computacionalmente costoso. Como señala Gazaba (2021), "si bien resulta conveniente representar capas de provincias o estados con un nivel de detalle que respete sus límites administrativos, la definición vértice por vértice de cada superficie regional implica una sofisticación gráfica significativa, lo que obliga a evaluar estratégicamente la distribución de esfuerzos entre el detalle cartográfico y la calidad de los atributos asociados a cada entidad geográfica".

Para optimizar este proceso, se implementó el formato GeoJSON, un estándar especializado de JSON diseñado para representar estructuras geoespaciales mediante coordenadas y geometrías vectoriales. Este formato permite describir tanto la ubicación como la morfología de objetos geográficos mediante propiedades metadatos que enriquecen el contexto espacial. Las primitivas geométricas implementadas incluyen: Point para ubicaciones puntuales unívocas, LineString para trazos definidos por secuencias de coordenadas, y Polygon para áreas delimitadas por anillos exteriores—y opcionalmente interiores para representar perforaciones.

Como se ilustra en la Figura 4, la implementación de este proyecto utilizó archivos GeoJSON para definir los límites municipales del estado de Hidalgo mediante polígonos vectoriales, equilibrando así precisión cartográfica con eficiencia computacional.

Figura 4. Script. Archivo GeoJSON

```
1  {
2    "type": "FeatureCollection",
3    "crs": {
4      "type": "name",
5      "properties": {
6        "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
7      }
8    },
9    "features": [
10     {
11       "type": "Feature",
12       "properties": {
13         "CVEGEO": "13029",
14         "CVE_ENT": "13",
15         "CVE_MUN": "029",
16         "NOMGEO": "Huichapan",
17         "NOM_ENT": "Hidalgo",
18         "COV_": 461,
19         "COV_ID": 2475,
20         "AREA": 65553.470426700005,
21         "PERIMETER": 176488.27853899999
22       },
23       "geometry": {
24         "type": "Polygon",
25         "coordinates": [
26           [
27             [
28               -99.510013599914004,
29               20.47418862043342
30             ],
31             [
32               -99.509587138713954,
33               20.473948779729056
34             ],
35           ]
36         ]
37       }
38     }
39   ]
40 }
```

La estructura del archivo GeoJSON implementa componentes esenciales para la representación geoespacial eficiente. En la línea 9, el array 'features' actúa como contenedor principal que organiza y agrupa las entidades geoespaciales individuales, estableciendo la base estructural del documento. Cada elemento dentro de este array se define mediante objetos específicos que, como se especifica en la línea 11, contienen la propiedad 'type: "Feature"', indicando que se trata de entidades geográficas con ubicación espacial definida y atributos descriptivos asociados.

La línea 13 introduce el objeto 'properties', que funciona como un repositorio de metadatos alfanuméricos, almacenando información descriptiva relevante como el nombre del municipio, su área territorial y perímetro, entre otros atributos significativos. Finalmente, en la línea 23, el componente 'geometry' constituye el núcleo espacial del formato, especificando mediante coordenadas georreferenciadas la forma geométrica precisa y la ubicación exacta de cada entidad en el sistema de referencia cartográfico, permitiendo así la visualización y análisis espacial de los datos municipales.

Esta estructura jerárquica garantiza tanto la precisión técnica como la eficiencia en el procesamiento de la información geoespacial.

Desarrollo de la aplicación

El framework *React Native* constituye el núcleo tecnológico del desarrollo de esta aplicación SIG. Basado en *JavaScript*, este framework adopta un enfoque declarativo que permite a los desarrolladores especificar el estado deseado de la interfaz de usuario, mientras que *React Native* gestiona automáticamente las actualizaciones y optimizaciones subyacentes. Esta arquitectura facilita la creación de interfaces dinámicas y altamente interactivas, mejorando significativamente la experiencia de desarrollo.

Una ventaja destacable de *React Native* radica en su capacidad para acceder directamente a las *API* nativas del dispositivo, lo que permite aprovechar al máximo las capacidades del hardware subyacente. Esta característica resulta particularmente valiosa en aplicaciones de sistemas de información geográfica, donde se requiere un alto rendimiento para el procesamiento de datos espaciales y una integración eficiente con funcionalidades específicas del dispositivo, como GPS, sensores y capacidades gráficas.

Gracias a su capacidad de generar código para múltiples plataformas a partir de una base de código única, junto con su rendimiento optimizado, *React Native* se consolida como una herramienta fundamental para el desarrollo de aplicaciones SIG modernas, escalables y cross-platform.

React Native Maps: Llevando los Mapas a las Aplicaciones Móviles

React Native Maps es una biblioteca integrada en el ecosistema de *React Native* que permite acceder, manipular y visualizar mapas de manera eficiente. En el contexto de los Sistemas de Información Geográfica (SIG), es fundamental la capacidad de superponer múltiples capas de información sobre un mapa para enriquecer la visualización de datos. Esta funcionalidad es invaluable, ya que permite a los usuarios analizar y abstraer datos de conjuntos complejos. Además, facilita la realización de consultas geoespaciales en tiempo real, así como la integración de información enviada desde ubicaciones específicas.

La biblioteca ofrece un amplio conjunto de herramientas ideales para el desarrollo de aplicaciones con funcionalidades geoespaciales. Entre las características más destacadas se encuentran:

- **Renderizado de mapas:** Permite la integración de mapas vectoriales o basados en imágenes, ofreciendo flexibilidad en la elección del proveedor, como Google Maps, Apple Maps u otros. En este proyecto, el mapa se centra en el estado de Hidalgo, facilitando la exploración de sus municipios.
- **Marcadores personalizables:** Los desarrolladores pueden agregar marcadores en ubicaciones específicas para resaltar puntos de interés. Estos marcadores admiten íconos e imágenes personalizadas. Al seleccionarlos, se despliega información detallada sobre el municipio correspondiente.
- **Polígonos dinámicos:** Los municipios se representan mediante polígonos que pueden colorearse según variables seleccionadas. La visualización se actualiza inmediatamente al cambiar los criterios.
- **Interacción fluida con el mapa:** La biblioteca admite gestos intuitivos como zoom, desplazamiento y rotación, mejorando la experiencia del usuario. También incluye una vista inmersiva a nivel de calle.

Inicialización de la Región con MapView

MapView es un componente clave de React Native Maps que proporciona opciones avanzadas de personalización y control sobre la visualización. Este permite configurar una región inicial con coordenadas específicas y un nivel de zoom ajustable. Adicionalmente, MapView facilita la interacción con los usuarios mediante la incorporación de marcadores, rutas y eventos, convirtiéndose en una herramienta esencial para aplicaciones basadas en mapas.

En la Figura 5 se muestra un fragmento de código que define un componente `<MapView>`. La propiedad `initialRegion` establece las coordenadas iniciales del mapa, mientras que las propiedades `latitudeDelta` y `longitudeDelta` controlan el nivel de zoom, donde valores mayores muestran áreas extensas y valores menores acercan la vista.

Figura 5. Script. MapView con región predefinida

```

65   <MapView
66     style={{ flex: 1 }}
67     initialRegion={{ latitude: 20.4799, longitude: -99.2456, latitudeDelta: 3, longitudeDelta: 3 }}
68     customMapStyle={mapStyle}
69   >

```

Carga de Datos GeoJSON

En la Figura 6 se ilustra un bloque de código que procesa los datos de un archivo GeoJSON (llamado `hidalgoData`) para extraer y transformar las coordenadas de polígonos o multipolígonos que representan áreas geográficas. Las operaciones principales incluyen:

- Iteración sobre cada feature del archivo GeoJSON.
- Identificación de geometrías como MultiPolygon (varios polígonos).
- Conversión de las coordenadas de formato `[longitud, latitud]` a objetos con las propiedades `latitud` y `longitud`, requeridas por `MapView`.
- Almacenamiento de las geometrías procesadas en el estado `polygons` mediante `setPolygons(newPolygons)` para su posterior renderización en el mapa.

Figura 6. Script. Extracción de geometrías del GeoJSON.

```
30  const newPolygons = hidalgoData.features.map(feature => {
31    const coordinates = feature.geometry.type === 'MultiPolygon' ?
32      feature.geometry.coordinates.flat(2) :
33      feature.geometry.coordinates[0];
34
35    return coordinates.map(coordSet => ({
36      latitude: coordSet[1],
37      longitude: coordSet[0],
38    }));
39  });
40  setPolygons(newPolygons);
```

Renderización de los polígonos

En la Figura 7 se muestra un fragmento de código que utiliza el componente `Polygon` de `React Native Maps` para renderizar las áreas geográficas en el mapa. Las propiedades configuradas incluyen:

- `coordinates`: Define las coordenadas del polígono.
- `fillColor`: Establece el color de relleno, calculado dinámicamente según el criterio seleccionado.
- `strokeColor` y `strokeWidth`: Ajustan el color y grosor del borde.
- Propiedades de estilo: Como `lineCap`, `lineJoin` y `miterLimit`, para personalizar el dibujo de líneas.

Este enfoque permite visualizar claramente las áreas geográficas de los municipios en el mapa de manera estilizada.

Figura 7. Script. Uso del componente Polygon.

```
71 <Polygon
72   key={index}
73   coordinates={coords}
74   fillColor={getFillColorForButton(selectedButton, index)}
75   strokeColor="#0B4F2A"
76   strokeWidth={1.4}
77   lineCap="round"
78   lineJoin="bevel"
79   miterLimit={5}
80 />
81 )}}
```

Carga de Marcadores y Archivo JSON del INEGI

La colocación de marcadores en un mapa mediante un archivo JSON permite centralizar todos los puntos de interés en un solo recurso, facilitando su visualización organizada. Esto optimiza la legibilidad del código, y simplifica el proceso de actualización, ya que agregar o modificar un marcador requiere únicamente editar el archivo de datos.

En la Figura 8 se muestra el código que recorre los datos de `populationData` y agrega marcadores en el mapa para cada municipio. El proceso incluye:

- Verificación de la existencia de coordenadas para cada marcador.
- Conversión de coordenadas almacenadas como cadenas (strings) a valores numéricos.
- Creación de marcadores con el nombre del municipio como etiqueta.
- Exclusión de municipios sin coordenadas para evitar errores.

Finalmente, los marcadores se renderizan en el mapa, proporcionando puntos interactivos en cada ubicación.

Figura 8. Script. Colocación de un marcador por municipio.

```
42 const newMarkers = populationData.map(data => {
43   if (data.Marcador) {
44     const [latitude, longitude] = data.Marcador.split(',');
45     return {
46       coordinate: {
47         latitude: parseFloat(latitude),
48         longitude: parseFloat(longitude),
49       },
50       municipality: data.NOM_MUN
51     };
52   }
53   return null;
54 }).filter(marker => marker !== null);
```

Interacción con Marcadores

Respecto a la interacción con marcadores, dentro de la Figura 9 se observa el código que permite agregar funcionalidad interactiva a los marcadores. Las propiedades del componente Marker incluyen:

- `coordinate`: Define la ubicación del marcador.
- `title`: Muestra el nombre del municipio.
- `image`: Representa un ícono personalizado.
- `onPress`: Llama a una función (`handleMarkerPress`) que despliega información detallada del municipio en un cuadro de diálogo.

Colorear Polígonos Basados en Datos

Las capas son conjuntos concretos de datos temáticos. Estas se superponen con el propósito de conformar mapas detallados. Luna Marín (2023) sostiene que cada territorio tiene características particulares que determinan su nivel de riesgo, el cual puede representarse de forma provisional para facilitar la planeación de acciones preventivas y correctivas orientadas a reducir desastres. Por lo tanto, al combinar capas temáticas, es posible construir escenarios de riesgo más precisos.

Figura 9. Script. Uso de propiedades interactivas en el componente Marker.

```
203 <Marker
204   key={index}
205   coordinate={marker.coordinate}
206   title={marker.municipality}
207   image={markerImage}
208   onPress={() => {
209     handleMarkerPress(markerAlert);
210   }}
211 />
```

En las figuras 10 y 11 se muestran fragmentos de código que asignan colores a los polígonos en función de atributos específicos, como el porcentaje de viviendas sin computadora ni Internet.

Algunas características clave adicionales que se integraron en la aplicación son:

lowColor y *highColor* definen los extremos de la escala de colores.

getColor calcula un color intermedio según el valor del atributo y su proporción respecto al máximo.

Los polígonos se colorean dinámicamente para representar visualmente las variaciones entre municipios, facilitando la interpretación de los datos.

Figura 10. Script. Colorear el polígono del municipio.

```
226   const lowColor = [255, 255, 255]; // Blanco
227   const highColor = [0, 0, 0]; // Negro
228
229   const getColor = (value, maxValue) => {
230     const normalizedValue = value / maxValue;
231     const color = lowColor.map((low, index) => {
232       const high = highColor[index];
233       return Math.round(low + (high - low) * normalizedValue);
234     });
235     return `rgba(${color[0]}, ${color[1]}, ${color[2]}, 1)`;
236   };
```

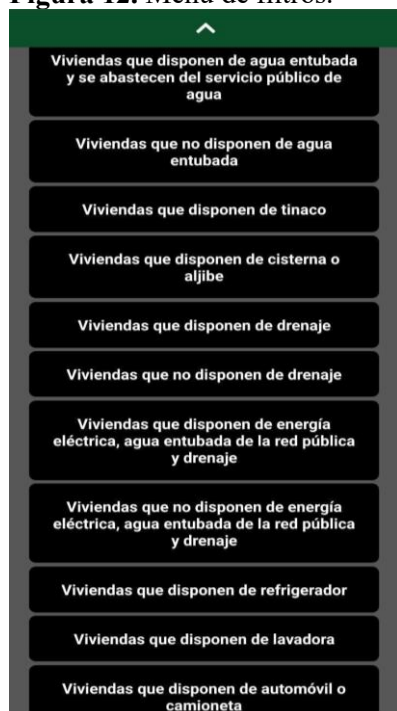
Figura 11. Script. Asignación del color según el valor comparativo.

```
365   const getColorForPopulation33 = (populationDataForMunicipality) => {
366     const valorComparativa = populationDataForMunicipality.VPH_SINCINT / populationDataForMunicipality.VIVTOT;
367     return getColor(valorComparativa * 100, 100);
368   };
```

Menú de Selección de Datos

En la Figura 12 se presenta el diseño de un menú interactivo que permite a los usuarios filtrar los datos mostrados en el mapa. Cada botón, como el de la Figura 13, activa un filtro específico, actualizando dinámicamente la visualización.

Figura 12. Menú de filtros.



En la Figura 13, se presenta el bloque de código que define un botón (TouchableOpacity) que, al ser presionado, ejecuta dos acciones principales: llama a la función handleButtonPress con el argumento 'VPH_SINTIC' para aplicar un filtro relacionado con las viviendas que no cuentan con algún aparato o dispositivo, línea telefónica o Internet, y cierra el menú desplegable (setShowMenu(false)). El botón tiene un estilo dinámico que cambia cuando está seleccionado (selectedButton === 'VPH_SINTIC'), y muestra un texto descriptivo sobre la opción que representa.

Figura 13. Script. Botón temático.

```

817 <TouchableOpacity
818   style={([stylesBotones.button, selectedButton === 'VPH_SINTIC' && stylesBotones.selectedButton])}
819   onPress={() => {
820     handleButtonPress('VPH_SINTIC');
821     setShowMenu(false);
822   }}
823 >
824   <Text style={stylesBotones.buttonText}>Total de viviendas que no cuentan con
825     algún aparato o dispositivo, línea telefónica o Internet</Text>
826 </TouchableOpacity>

```

Mostrar Cuadro de Diálogo

En la Figura 14 se muestra un código que calcula un porcentaje (como viviendas sin computadora ni Internet) y genera un mensaje detallado con el nombre del municipio, el número de viviendas afectadas, el total de viviendas y el porcentaje resultante. Este mensaje se muestra al interactuar con los marcadores en el mapa, brindando información geoespacial relevante al usuario.

Figura 14. Script. Cuadro de diálogo.

```

189 } else if (selectedButton === 'VPH_SINTIC') {
190   const porcentaje = (((
191     populationDataForMunicipality.VPH_SINTIC) /
192     populationDataForMunicipality.VIVTOT)
193     * 100).toFixed(3);
194   markerAlert = `${municipio} ${marker.municipality}
195     \n\nTotal de viviendas que no cuentan con algún aparato o dispositivo, línea telefónica o Internet:
196     ${populationDataForMunicipality ? populationDataForMunicipality.VPH_SINTIC : 'N/A'}
197     \n\nTotal de viviendas del municipio: ${populationDataForMunicipality
198       ? populationDataForMunicipality.VIVTOT : 'N/A'}
199     \n\nPorcentaje comparativo: ${porcentaje}%';
200 }

```

RESULTADOS

Tras la finalización de la codificación, se ejecutó el sistema, mostrando una interfaz de interacción inicial que incluye un mapa sin filtros aplicados, el cual presenta únicamente la división política del estado. También se habilitó un menú desplegable que ofrece las opciones disponibles para aplicar filtros y capas temáticas al mapa. A continuación, se ejemplifica el funcionamiento de la aplicación

Se presenta un ejemplo del mapa generado por la aplicación Hidalgo Horizon, con un atributo seleccionado por el usuario. Además, se destacan las interfaces y funcionalidades que permiten visualizar los datos censados del estado de Hidalgo.

Figura 15. Aplicación de una capa temática al mapa.

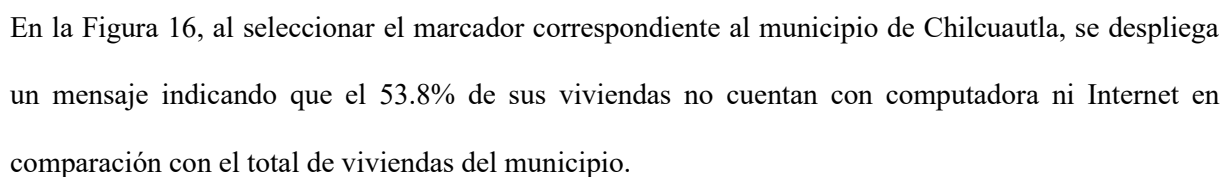


Figura 16. Ventana emergente.



Trabajo futuro

Se pretende que el sistema evolucione hacia un procesamiento completamente automatizado de archivos censales en formato Excel, eliminando la necesidad de intervención manual. El proceso se iniciará con la carga del archivo mediante la librería SheetJS (xlsx), que permite leer y manipular datos de hojas de cálculo de manera eficiente, eliminando la necesidad de preparar manualmente el JSON. Utilizando algoritmos de reconocimiento inteligente, el sistema identificará automáticamente columnas clave (ej. nombre del municipio, población, acceso a servicios básicos) y las mapeará a un esquema geoespacial estandarizado, adaptable a múltiples estructuras de datos censales.

Para garantizar la integridad de los datos, se implementará un módulo de validación que detectará inconsistencias (valores nulos, formatos erróneos) y aplicará correcciones predeterminadas (ej. rellenar vacíos con ceros o promedios). Adicionalmente, se integrará un servicio de geocodificación basado en APIs, que insertará las coordenadas geográficas a entidades administrativas (municipios, localidades)

y generará polígonos dinámicos para su visualización en MapView. Esto eliminará la dependencia de archivos GeoJSON predefinidos, permitiendo que cualquier archivo censal como los publicados por el INEGI o gobiernos estatales sea procesado y visualizado en tiempo real.

Como valor añadido, el sistema incorporará un panel de configuración donde los usuarios podrán personalizar el mapeo de columnas, ajustar umbrales de validación y seleccionar criterios de comparación. Esta escalabilidad no solo optimizará el análisis de datos para Hidalgo, sino que facilitará su adaptación a otras regiones o contextos (ej. salud, educación).

Se planea añadir gráficos interactivos utilizando la librería Chart.js, los cuales complementarán la visualización geoespacial de los datos. Actualmente se cuenta con un prototipo para la graficación de las variables.

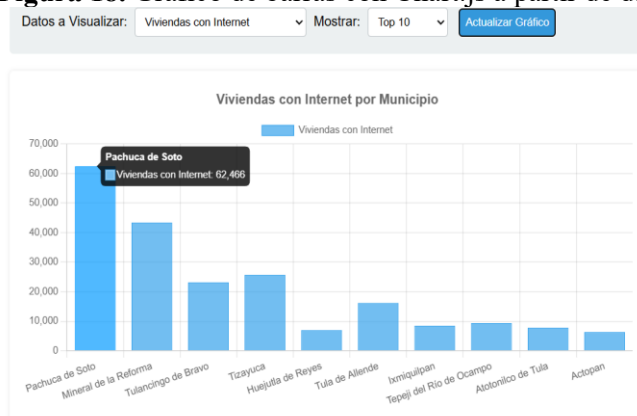
En el código mostrado en la figura 17 se toma la información de los municipios y se organiza para que Chart.js pueda crear el gráfico. Primero, selecciona solo los municipios que se mostrarán. Luego, extrae los nombres de los municipios (labels) y los valores numéricos (data) según la opción elegida. Los valores numéricos en el array “data” son los que se usan para determinar el tamaño relativo de cada barra. Sin esta asignación, todas las barras tendrían la misma altura.

Figura 17. Script. Gráficas de barras

```
153 // Obtener etiquetas y datos para el gráfico
154 const municipiosMostrar = datosMunicipios.slice(0, numMunicipios);
155 const labels = municipiosMostrar.map(m => m.NOM_MUN);
156 const data = municipiosMostrar.map(m => m[tipoDato]);
157
158 // Obtener nombre legible del dato seleccionado
159 const dataLabel = dataTypeSelect.options[dataTypeSelect.selectedIndex].text;
160
161 // Configuración del gráfico de barras
162 const config = {
163   type: 'bar',
164   data: {
165     labels: labels,
166     datasets: [{
167       label: dataLabel,
168       data: data,
169       backgroundColor: 'rgba(54, 162, 235, 0.7)',
170       borderColor: 'rgba(54, 162, 235, 1)',
171       borderWidth: 1
172     }]
173   },
174 }
```

Finalmente, se configura el gráfico de barras: los nombres de los municipios aparecerán en el eje X, los valores numéricos definirán la altura de cada barra, y se aplica un color. El título de la leyenda (dataLabel) se toma directamente del menú desplegable para que el usuario sepa qué datos está viendo.

Figura 18. Gráfico de barras con Chart.js a partir de datos municipales filtrados



CONCLUSIONES

Como señala Harley (1989), los mapas no son meras representaciones objetivas del territorio, sino construcciones sociales que reflejan relaciones de poder, decisiones metodológicas y puntos de vista particulares que se plasman en ellos. Por tanto, su diseño y uso deben entenderse como actos cargados de significado, más allá de su función operativa.

El desarrollo de este Sistema de Información Geográfica (SIG) facilita un análisis claro y comprensivo de datos informativos relacionados con cualquier municipio. Esto permite visualizar la información de una manera minuciosa que puede ser utilizada por asociaciones, estados, secretarías y cualquier persona interesada. De esta manera visualizar el territorio representa un ejercicio de interpretación plural, que adecua formas proyectar la realidad.

El desarrollo de esta aplicación ha permitido identificar conexiones complejas entre las variables seleccionadas. Por ejemplo, en algunos municipios del norte del estado, se detectó una alta proporción de carencia de servicios, como viviendas sin computadoras ni acceso a internet, a diferencia de los municipios del sur, donde el acceso a estos servicios es mayor (Figura 15). Esta información es fundamental para que las autoridades implementen estrategias destinadas a reducir las brechas y equilibrar la atención brindada en diferentes regiones.

Además, se observó una tendencia en la que los municipios con mayor acceso a servicios suelen ser aquellos más industrializados o próximos a la capital del estado. Esto coincide con lo señalado por Tan et al. (2025), "La concentración de valores más altos en las principales áreas urbanas se debe potencialmente a un alto nivel de movilidad poblacional alrededor de los centros urbanos. Esto está

relacionado con una mayor densidad poblacional, desplazamientos regulares y usos del suelo mixtos, con funciones principales en manufactura, comercio o servicios públicos".

Por otro lado, es fundamental reconocer que las personas tienden a ver su entorno como un factor determinante en sus posibilidades de desarrollo. Como señala Castillo (2016), "las personas suelen percibir su entorno como un obstáculo para la movilidad social, ya sea por su influencia directa o por la desmotivación que genera su círculo cercano". En este sentido, la aplicación busca invitar a los ciudadanos a reflexionar sobre cómo perciben su entorno, ofreciendo una visión más completa de los factores externos que influyen en su movilidad social.

La aplicación propone una forma integral de analizar las características tanto homogéneas como mixtas del entorno. Al presentar los datos de manera detallada, fomenta un ejercicio de autocomprensión colectiva, facilitando la identificación de cómo el espacio geográfico influye en las dinámicas sociales y económicas que estructuran la pertenencia y cohesión dentro de una sociedad.

Además, a través de este desarrollo se busca llegar a mapear regiones a nivel mundial y con ello poder mostrar gráficamente a través de los datos oficiales, lo que sucede en cada una de las regiones, brindando un fácil uso y poniéndolo a disponibilidad de cualquier usuario como herramienta en la visualización de su entorno y toma de decisiones. Por otra parte puede ser una herramienta muy poderosa que brinde en tiempo real lo que sucede con los recursos naturales de la región y como han sido consumidos y con ello realizar prevención de algunos desastres naturales, ecológicos, entre otros.

Agradecimientos

Los autores agradecen al Instituto Nacional de Estadística y Geografía (INEGI) por poner a disposición pública los datos de los censos poblacionales y económicos, los cuales fueron fundamentales para el desarrollo de este estudio. La accesibilidad y calidad de esta información permiten enriquecer el análisis científico y fomentar investigaciones basadas en evidencia.

REFERENCIAS BIBLIOGRÁFICAS

1. Hynes, W. (2021, Mayo 7). Preparing for future shocks. Politeia. <https://www.politeia.co.uk/preparing-for-future-shocks/>
2. Calixto Escariz. (2017, 22 agosto). Sistemas de Información Geográfica: ventajas para la gestión local del territorio. <https://www.calixtoescariz.com/blog/ventajas-sistemas-informacion-geografica/>

3. Banco Mundial. (2021). Informe sobre el desarrollo mundial 2021: Datos para una vida mejor. <http://hdl.handle.net/10986/35218> License: CC BY 3.0 IGO.
4. Arancibia, M. E. (2008). El uso de los sistemas de información geográfica -SIG- en la planificación estratégica de los recursos energéticos. Revista de Universidad Bolivariana. https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-65682008000100012&lng=en&nrm=iso&tlng=en
5. Gazaba, F. J. (2021, 17 marzo). Introducción a los Sistemas de Información Geográfica con QGIS. <https://pergamino.ar/descargas/introduccion-a-los-sig-con-qgis.pdf>
6. Luna Marín, V. (2023). Aplicación de los SIG en el análisis de Riesgo de Desastres generados por Amenazas naturales y antrópicas: Una Revisión Bibliográfica. Universidad de Antioquia, Medellín, Colombia. <https://bibliotecadigital.udea.edu.co/server/api/core/bitstreams/cb113039-1bc7-4fe8-b783-b5b6e327b327/content>
7. Harley, J. B. (1989). Desconstruyendo el mapa. Cartographica: Revista internacional de información geográfica y geovisualización. <https://doi.org/10.3138/E635-7827-1757-9T53>
8. Tan, X., Huang, B., Batty, M. et al. (2025) Las leyes de escala espaciotemporal de la dinámica de la población urbana. <https://doi.org/10.1038/s41467-025-58286-4>
9. Castillo, M. (2016). Fronteras simbólicas y clases medias: Movilidad social en Chile. <https://doi.org/10.18504/pl2448-009-2016>