
Automatización Cartográfica: Integración de Tecnologías para la Exportación Automatizada de Mapas

Integration of Technologies for the Automated Export of Maps

Joel Fernando Morera Robles^a
joelmorera@usantotomas.edu.co

Andres Felipe Ortiz Rico^b
andresortiz@usta.edu.co

Resumen

En el panorama contemporáneo, caracterizado por la automatización generalizada y la presencia creciente de inteligencia artificial (AI), este proyecto representa una iniciativa estratégica para incrementar la eficiencia en la gestión de información geoespacial o en el tramamiento de datos con esta característica. Su enfoque específico recae en la optimización de la exportación manual de mapas, una tarea laboriosa, mediante la introducción de la función Atlas adjunta en el software QGIS LTR.

La propuesta no solo busca adoptar tecnologías avanzadas, sino también ser un impulsor esencial para la mejora de la eficiencia operativa. Al quitar la limitante necesidad de realizar manualmente la exportación de los mapas, el proyecto libera tiempo valioso para los operadores, permitiéndoles concentrarse en tareas más estratégicas y analíticas. Además, aspira a contribuir significativamente a la optimización de flujos de trabajo cartográficos, proporcionando un enfoque más dinámico y adaptable a las cambiantes demandas del entorno geoespacial, fomentando así una gestión más ágil y eficaz de la información cartográfica.

—
Palabras clave: Automatización, programación autónoma, exportación de mapas, eficiencia en tareas repetitivas, código abierto.

1. Introducción

En el ámbito de la cartografía y el análisis geoespacial, la automatización de procesos se rige como una meta esencial para mejorar la eficiencia y la calidad de la producción cartográfica [2, 3]. Inicialmente, se contempló el desarrollo de un código en Python como estrategia para codificar la automatización de la exportación de mapas, buscando así una solución técnica y programática a los desafíos intrínsecos asociados con la generación manual de productos cartográficos. Sin embargo, durante esta fase inicial de diseño, se evidenciaron limitaciones operativas sustanciales, destacando la necesidad de una exploración más profunda en busca de alternativas más eficaces.

En consecuencia, se emprendió una indagación minuciosa, guiada por la premisa de encontrar una solución que conjugara eficiencia y accesibilidad. En este contexto, surgió como solución clave la función Atlas en QGIS, un Sistema de Información Geográfica (SIG) de código abierto ampliamente reconocido. La función Atlas en QGIS, además de ofrecer una automatización robusta en la generación de mapas, destaca por su versatilidad y capacidad para adaptarse a diversas exigencias cartográficas.

^aEstudiante de último año del programa de Maestría en Estadística Aplicada, Universidad Santo Tomás, sede Bogotá.

^bDirector. Profesor asociado a la Facultad de Estadística, Universidad Santo Tomás, sede Bogotá.

Esta elección estratégica de migrar hacia la funcionalidad Atlas en QGIS está respaldada por la convicción de que la plataforma no solo supera las limitaciones experimentadas durante la fase de codificación en Python, sino que también proporciona una solución más integrada y eficiente para la generación automatizada de series de mapas. Al aprovechar esta herramienta, se pretende no solo optimizar la eficiencia de la exportación cartográfica, sino también dar apertura un paradigma que permita una producción cartográfica automatizada y de alta calidad en el ámbito geoespacial.

En este sentido, el presente proyecto no solo aborda la selección estratégica de QGIS y su función Atlas, sino que también se propone explorar y comprender de manera detallada cómo aprovechar al máximo esta característica en el contexto de la automatización de procesos cartográficos. La diversidad en el rigor científico con soluciones prácticas en la generación automatizada de mapas, constituye el núcleo de este proyecto, encaminado a impulsar los límites de la eficiencia y la calidad en la producción cartográfica contemporánea, eficiente y distribuible.

2. Desarrollo teórico

En esta fase, abordaremos la descripción del desafío empresarial asignado, así como las dos soluciones propuestas para su resolución.

Capstone: *El desafío empresarial abordado por este proyecto se centra en la optimización de los recursos temporales necesarios para generar mapas con características cartográficas específicas. Tradicionalmente, la ejecución de esta tarea implica un proceso manual de depuración de información y diseño de documentos, lo que consume considerablemente tiempo. Cada mapa, junto con sus respectivos polígonos, se crea individualmente, lo que se traduce en una inversión significativa de tiempo. Ante esta situación, el encargado de la sistematización y diseño de este recurso identificó la posibilidad de automatizar este proceso mediante la consola de Python integrada en el software utilizado, ArcMap. Con el objetivo de abordar eficientemente este reto, se planteó a un estudiante de maestría de la Universidad Santo Tomás la tarea de llevar a cabo la investigación correspondiente para ofrecer una solución efectiva que optimice los recursos involucrados en la generación de estos mapas.*

2.1. Primera estrategia de resolución: Consola y código Python

La primera estrategia propuesta para abordar este desafío consistió en la creación de un código en lenguaje Python diseñado para automatizar la secuencia de tareas requerida para la exportación de mapas con características cartográficas específicas. Aunque esta aproximación inicial demostró ser prometedora, se encontró con desventajas relacionadas con el tiempo empleado en la depuración consecutiva del código a través de la consola integrada en ArcMap para Python. Estas limitaciones resaltaron la necesidad de explorar alternativas más eficientes y optimizadas para garantizar un proceso de automatización sin contratiempos.

Python en ArcMap es una herramienta poderosa para la automatización y personalización de tareas geoespaciales, pero también tiene algunas desventajas. Aquí enlisto algunas de las desventajas más comunes de usar la consola Python en ArcMap:

1. **Limitaciones de la versión de Python:** ArcMap suele utilizar una versión específica de Python (por ejemplo, Python 2.x), que puede estar desactualizada en términos de características y seguridad. Python 2.x ya no recibe soporte oficial, lo que puede ser un riesgo de seguridad. -Problemas con el basemap-
2. **Interfaz de usuario limitada:** La consola Python en ArcMap tiene una interfaz de usuario básica y no es tan rica como los entornos de desarrollo integrados (IDE) dedicados para Python, como PyCharm o Visual Studio Code. Esto puede hacer que la escritura y depuración de código sea menos eficiente.

3. **Rendimiento limitado:** El rendimiento de Python en ArcMap puede ser más lento en comparación con otros entornos, especialmente al realizar tareas geoespaciales complejas. Esto se debe a la integración con el software de ESRI y la complejidad de algunas operaciones GIS.
4. **Dependencia de ArcObjects:** En algunas situaciones, es necesario interactuar con ArcObjects, la API de objetos de ArcGIS. Esto puede requerir conocimientos adicionales y hacer que el código sea más complejo.
5. **Limitaciones en el acceso a bibliotecas externas:** La integración de bibliotecas y paquetes de Python externos puede ser más complicada en ArcMap debido a las restricciones de seguridad y al hecho de que no todas las bibliotecas externas son compatibles.
6. **Versión específica de ArcMap:** El código desarrollado en una versión específica de ArcMap puede no ser completamente compatible con versiones posteriores o anteriores. Esto puede crear problemas de portabilidad y mantener el código a lo largo del tiempo.
7. **Dificultad en la gestión de entornos virtuales:** La gestión de entornos virtuales de Python en ArcMap puede ser complicada y puede requerir un conocimiento técnico más avanzado para configurar y mantener.
8. **Dependencia del entorno de ArcMap:** El código Python escrito en ArcMap generalmente depende de que ArcMap esté abierto y en funcionamiento. Esto limita la automatización en segundo plano o la ejecución programática fuera del entorno de ArcMap.

La toma de decisiones para abordar las limitaciones inherentes al manejo de la automatización desde la consola del software ArcMap condujo al diseño integral de un código en Python. Este enfoque se gestó considerando cuidadosamente la selección de bibliotecas específicas que facilitarían la automatización de los procesos asociados con la exportación de polígonos.

El código resultante, ahora adjunto al documento [Apéndice A](#), representa un logro parcial en términos de las características esenciales del producto final, que se centra en las exportaciones precisas de los polígonos. Su inclusión en el documento no solo cumple con la necesidad de materializar la solución, sino que también sirve como una guía detallada, desglosando paso a paso cada línea del código para una comprensión integral y accesible. Este enfoque no solo aborda las limitaciones iniciales identificadas, sino que también proporciona un recurso valioso para futuras iteraciones y mejoras en la automatización de procesos cartográficos.

2.2. Descripción del código

Este código en Python realiza varias tareas relacionadas con la generación de un PDF que contiene polígonos destacados a partir de un archivo shapefile, el cual es una extracción realizada desde el portal MGN-DANE. A continuación, se proporciona una descripción detallada de lo que hace el código:

1. **Importación de bibliotecas:** El código importa varias bibliotecas que se utilizan en el proceso, como FPDF para generar el PDF, ReportLab para dibujar en el PDF, Matplotlib para crear gráficos, GeoPandas para trabajar con datos geoespaciales, Contextily para agregar mapas base y otras bibliotecas para funciones auxiliares.
2. **Definición de rutas de archivos:** Se definen las rutas de los archivos de entrada y salida. El `shapefile_path` especifica la ubicación del archivo shapefile que contiene los datos geoespaciales de los polígonos, y `pdf_output_path` especifica la ubicación donde se guardará el archivo PDF resultante.
3. **Función cargar_shapefile:** Esta función carga el archivo shapefile, realiza algunas transformaciones en los datos y devuelve el GeoDataFrame resultante y el sistema de coordenadas de proyección.

- a) Lee el shapefile utilizando GeoPandas.
 - b) Cambia el sistema de coordenadas del GeoDataFrame al proyectado adecuado (en este caso, EPSG:21897).
 - c) Selecciona un subconjunto de filas del GeoDataFrame, en este caso, las filas 10 a 20.
 - d) Devuelve el GeoDataFrame resultante y el sistema de coordenadas de proyección.
4. **Función generar_poligonos_en_pdf:** Esta función genera un archivo PDF que contiene los polígonos destacados a partir del GeoDataFrame de entrada.
- a) Comprueba si el GeoDataFrame de entrada (**shapefile**) es nulo; si lo es, muestra un mensaje de error y sale de la función.
 - b) Calcula el número total de polígonos en el GeoDataFrame.
 - c) Inicializa un objeto FPDF para crear el PDF y configura la función de salto de página automático.
 - d) Crea una barra de progreso visual utilizando **tqdm** para mostrar el progreso de generación del PDF.
 - e) Itera a través de cada polígono en el GeoDataFrame y realiza las siguientes acciones para cada uno:
 - f) Agrega una nueva página al PDF.
 - g) Crea una figura de Matplotlib para dibujar el polígono.
 - h) Dibuja los polígonos en la figura, resaltando el polígono actual en rojo y los demás en negro.
 - i) Agrega el número de polígono en el centro de cada polígono.
 - j) Configura los límites y agrega un mapa base de OpenStreetMap utilizando Contextily.
 - k) Elimina ejes y etiquetas en la figura.
 - l) Agrega un título de página y un número de polígono al PDF.
 - m) Guarda la figura como una imagen temporal.
 - n) Inserta la imagen en el PDF.
 - ñ) Actualiza la barra de progreso.
 - o) Cierra la barra de progreso y guarda el PDF en el archivo de salida especificado.
5. **Carga del shapefile y generación del PDF:** El código carga el shapefile utilizando la función **cargar_shapefile**, y luego genera el PDF utilizando la función **generar_poligonos_en_pdf**. Si no se pudo cargar el shapefile, se muestra un mensaje de error y no se genera el PDF.

En resumen, este código carga un archivo shapefile, selecciona un subconjunto de polígonos, crea un PDF que resalta cada polígono en una página diferente, agrega un mapa base y genera subtítulos con el número del polígono representado. Finalmente, guarda el PDF resultante en una ubicación específica. **No obstante, el código presentaba ciertas limitaciones, especialmente en lo referente a la exportación solicitada. La principal novedad que hacía que esta opción no fuera completamente viable residía en la disposición de los polígonos. En situaciones donde los polígonos estaban relativamente cercanos entre sí, la exportación se ejecutaba de manera adecuada. Sin embargo, cuando los polígonos se dispersaban, la calidad y eficiencia de la exportación se veían comprometidas. Este factor se convertía en un obstáculo importante para garantizar la consistencia y la precisión en el resultado final del proceso de exportación geoespacial.**

2.3. Segunda estrategia de resolución: Funciones en QGIS LTR

En la búsqueda de la solución más eficiente para abordar el desafío planteado, se exploraron diversas opciones para identificar las funcionalidades de software especializado en el manejo de datos georreferenciados. Durante este proceso de indagación, surgió como una opción destacada el software QGIS, que ofrece una funcionalidad específica llamada ".Atlas". Esta característica demostró ser altamente relevante y efectiva para la automatización de tareas, particularmente en el ámbito del procesamiento de información geoespacial.

La función Atlas en QGIS se reveló como una solución integral para la automatización de tareas de exportación, ofreciendo un enfoque específico y avanzado para la manipulación eficiente de datos geoespaciales. Esta opción no solo representa una alternativa eficaz sino que también se alinea con la naturaleza especializada de las tareas cartográficas. La inclusión de esta funcionalidad en la estrategia de automatización contribuye significativamente a la mejora de los flujos de trabajo, consolidando la eficiencia y la precisión en el proceso de exportación de datos geográficos.

2.3.1. Descripción de la Función Atlas en QGIS

La función Atlas en QGIS es una herramienta poderosa diseñada para facilitar la creación y automatización de mapas en situaciones en las que se requiere generar una serie de mapas relacionados, cada uno enfocado en una entidad espacial específica, como polígonos, puntos o líneas. Esta función es especialmente valiosa en proyectos cartográficos que involucran la representación detallada de múltiples áreas geográficas.

En esencia, el Atlas actúa como un generador dinámico de mapas, permitiendo la visualización y exportación eficiente de datos geoespaciales mediante una plantilla común. Sus características clave incluyen:

1. **Dinamismo Geoespacial:** La función Atlas puede adaptarse automáticamente a los límites de diferentes entidades geográficas, mostrando solo la información relevante para cada elemento del conjunto de datos.
2. **Personalización y Plantillas:** Permite la creación de plantillas personalizadas para la representación gráfica de mapas, con la capacidad de ajustar estilos, etiquetas y otros elementos cartográficos según las necesidades específicas de cada entidad.
3. **Automatización de Procesos:** Facilita la generación automática de una serie de mapas individualizados basados en una plantilla predefinida. Este proceso se realiza mediante la vinculación de la función Atlas con un conjunto de datos geoespaciales.
4. **Eficiencia en la Exportación:** Permite exportar la serie de mapas generados en varios formatos, como imágenes o archivos PDF, proporcionando así una herramienta eficaz para la comunicación visual de información geográfica.
5. **Exploración Iterativa:** Facilita la exploración iterativa de datos geográficos, permitiendo a los usuarios revisar y analizar áreas específicas de interés de manera rápida y sistemática.
6. **Flexibilidad en la Representación:** Ofrece flexibilidad en la representación gráfica de entidades, adaptándose a la complejidad y distribución geográfica de los datos.

En resumen, la función Atlas en QGIS simplifica y mejora significativamente el proceso de creación de mapas para conjuntos de datos geoespaciales, brindando a los usuarios una herramienta versátil y eficiente para la generación y automatización de mapas temáticos.

2.4. Diagramas de flujo

En el marco de este análisis, se ha considerado fundamental representar gráficamente la estructura lógica de las estrategias propuestas para abordar el desafío planteado. Estos diagramas de flujo no solo proporcionan una visión clara de la secuencia de pasos involucrados en cada enfoque de resolución, sino que también destacan las particularidades distintivas de cada etapa.

En primer lugar, el diagrama de flujo de la estrategia de código Python muestra cómo se realiza la importación de bibliotecas esenciales, la definición de rutas de archivos y la ejecución de funciones clave como la carga del shapefile y la generación del PDF. Cada paso está meticulosamente detallado, proporcionando una guía comprehensiva para entender la lógica detrás de la automatización del proceso de exportación cartográfica.

En contraste, el diagrama de flujo relacionado con la estrategia de Atlas en QGIS destaca la secuencia específica de acciones vinculadas a esta herramienta. Desde la carga del shapefile hasta la configuración de parámetros y la exportación automatizada, cada paso se presenta de manera visual y accesible. Este enfoque estratégico se distingue por su capacidad para gestionar la dispersión de polígonos, superando así la limitación observada en el código Python.

Estos diagramas de flujo sirven como herramientas valiosas para comprender la implementación detallada de cada estrategia, permitiendo una evaluación clara de las etapas involucradas y facilitando la identificación de posibles puntos de mejora. Además, ofrecen una base visual para la discusión y el análisis crítico de la eficacia y eficiencia de cada enfoque en la optimización de los flujos de trabajo cartográficos.

3. Configuración Atlas - QGIS LTR

Se ha desarrollado una serie de episodios en formato de video tutoriales que abordan exhaustivamente cada fase del procesamiento de mapas y la configuración de la exportación. Estos tutoriales en video se han concebido como una herramienta didáctica integral, proporcionando una explicación detallada de cada etapa involucrada en la resolución del capstone a través de la funcionalidad Atlas en QGIS.

Cada episodio se centra en una fase específica del proceso, desglosando meticulosamente las acciones y decisiones clave tomadas durante la configuración de Atlas en QGIS. Desde la carga inicial del shapefile hasta la definición de parámetros y la ejecución de la exportación automatizada, los tutoriales ofrecen una visión práctica y guiada de todo el flujo de trabajo.

Este recurso de video tutorial se reconoce como importante, ya que no solo complementa la documentación escrita y los diagramas de flujo, sino que también brinda a los usuarios la oportunidad de visualizar en tiempo real cómo se implementan las estrategias propuestas. Los episodios se han diseñado considerando la claridad y la accesibilidad, buscando proporcionar una guía visual comprensible que facilite la replicación del proceso y la comprensión de las complejidades involucradas en la optimización de los flujos de trabajo cartográficos mediante Atlas en QGIS.

Diagrama Proyecto Exportación de Mapas

Python

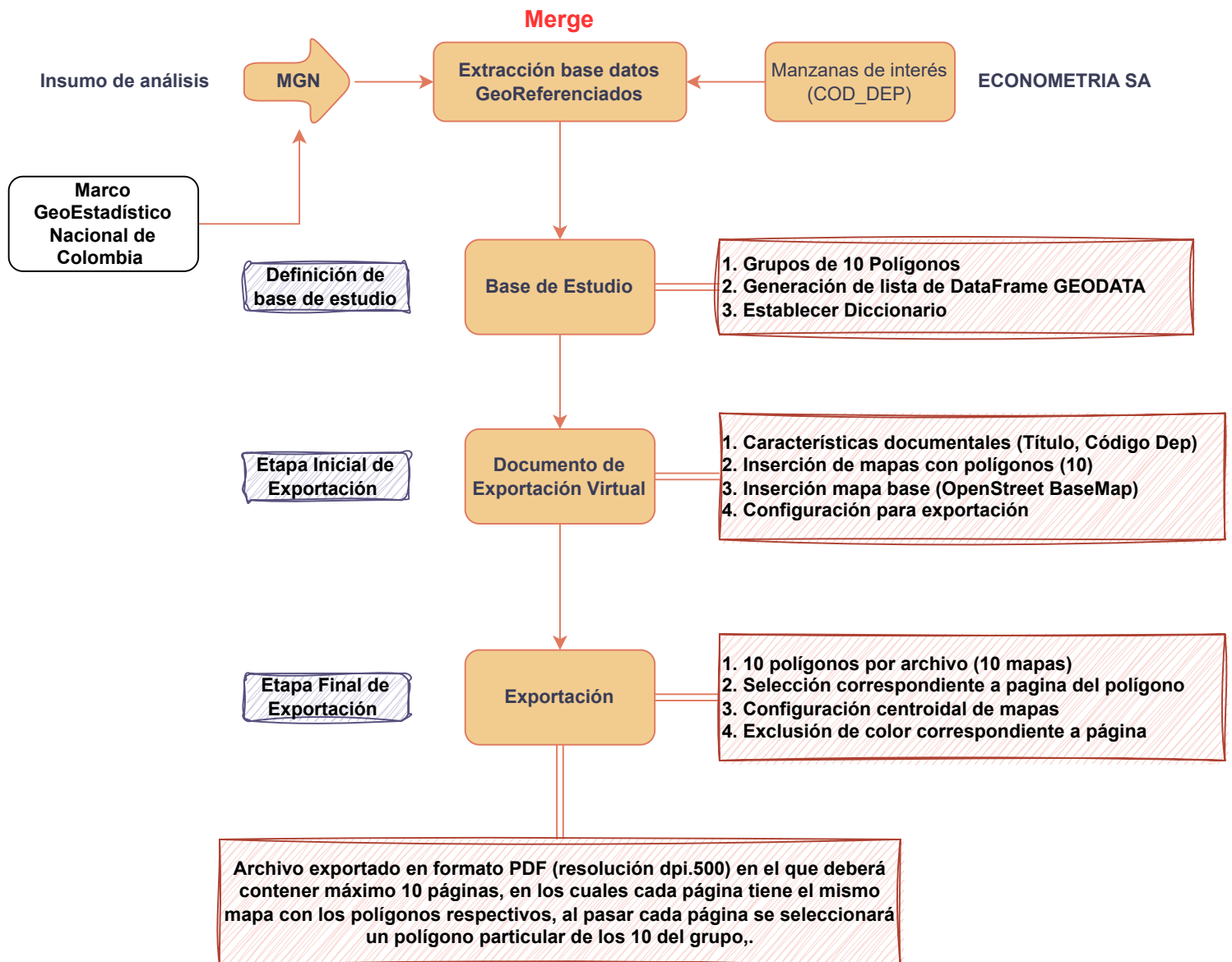
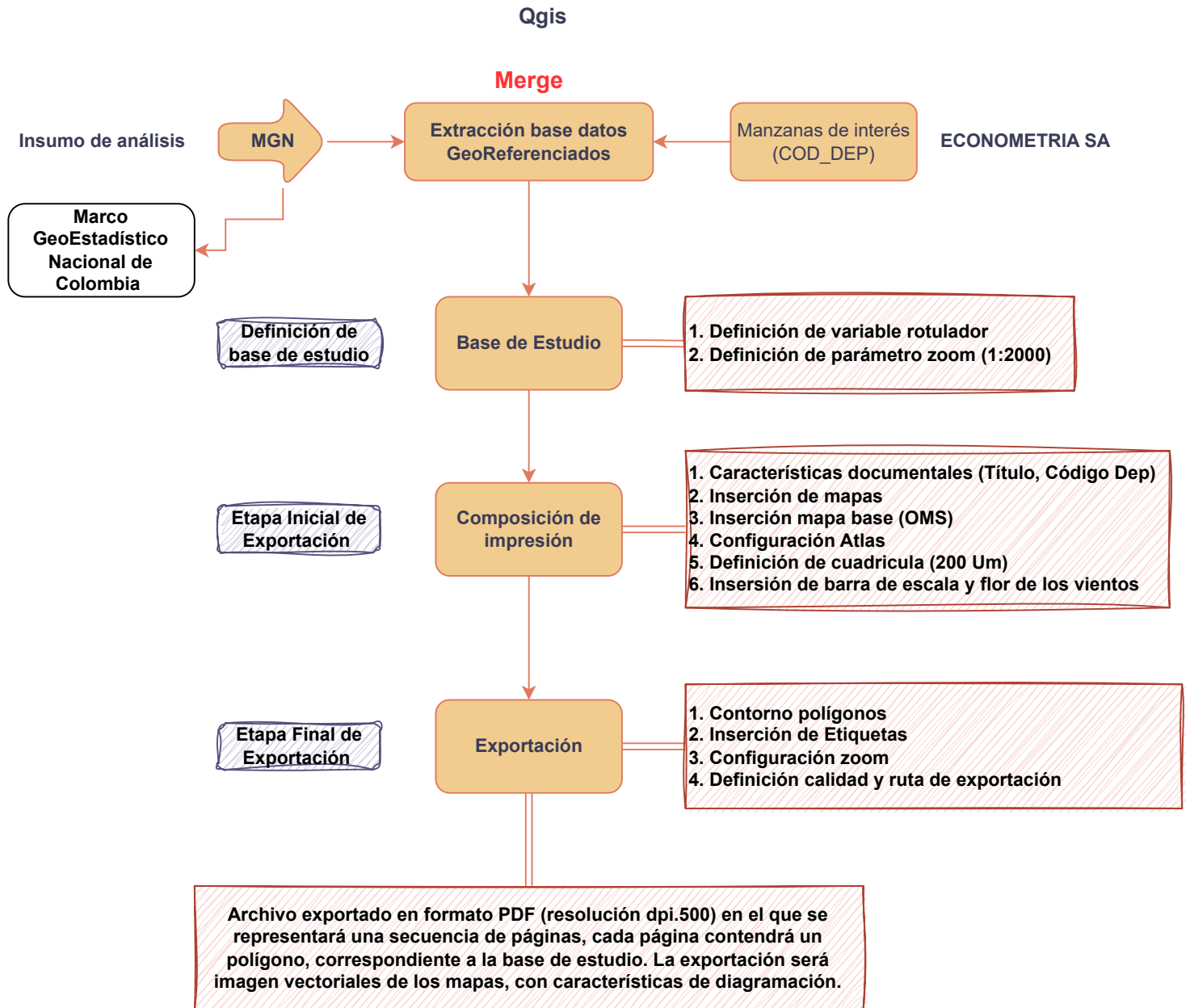


Diagrama Proyecto Exportación de Mapas



4. Objetivos

Objetivo General

Diseño de algoritmo en lenguaje Python - ArcMap o identificación de función en software para análisis de información geográfica, que permita automatizar las tareas de exportación de una serie determinada de mapas.

Objetivos Específicos

1. *Análisis descriptivo espacial sobre la no respuesta que se presente en las unidades de muestreo, para la identificación de patrones.*
2. *Generar protocolos de actualización del código de exportación para futuras tareas corporativas.*

5. Metodología

Existen varios aspectos importantes en la metodología de la indagación. A continuación se enlistará y describirá las etapas de indagación del presente estudio como el rango en etapas para el desarrollo de la misma. El proceso de indagación se organiza en las siguientes etapas de estudio:

1. **Análisis de Tareas:** Identificar las tareas específicas en QGIS que se desean automatizar.
2. **Selección de Plugins:** Elegir los complementos de QGIS apropiados para la automatización.
3. **Diseño del Flujo de Trabajo:** Crear un diseño detallado de cómo se realizará la automatización.
4. **Desarrollo de Scripts:** Escribir scripts o código para la consola en QGIS para llevar a cabo la automatización.
5. **Pruebas y Depuración:** Verificar que los scripts funcionen correctamente y corregir errores si es necesario.
6. **Integración con QGIS:** Incorporar los scripts en el entorno de QGIS.
7. **Exportación de Mapas con Atlas:** Utilizar la función de Atlas en QGIS para generar y exportar mapas de manera automatizada.
8. **Documentación:** Generar documentación que explique el funcionamiento de los scripts y su uso (vídeo tutoriales).
9. **Optimización y Eficiencia:** Identificar oportunidades para mejorar la eficiencia de los scripts.
10. **Mantenimiento y Actualización:** Establecer procedimientos para el mantenimiento a largo plazo.
11. **Evaluación Continua:** Evaluar regularmente el rendimiento y la eficacia de la automatización en QGIS.

6. Descripción espacial

En este segmento, nos sumergiremos en un análisis detallado de los polígonos seleccionados, centrándonos en entender las razones detrás de la ausencia de encuestas realizadas por la entidad encargada de la aplicación de instrumentos de recolección de información en estas áreas específicas. La muestra consiste en un conjunto de 160 registros, abarcando distintos polígonos situados en municipios específicos del departamento del Amazonas en Colombia.

El propósito fundamental de este análisis descriptivo es desentrañar las posibles causas que han llevado a que estos polígonos no sean objeto de encuestas. A través de la revisión minuciosa de características geográficas, demográficas y cualquier otro factor relevante, buscamos arrojar luz sobre las condiciones particulares que han llevado a la falta de atención en estos lugares específicos. Este enfoque detallado nos permitirá obtener una comprensión más profunda de los desafíos y circunstancias que rodean a estos polígonos, contribuyendo así a una evaluación más completa y contextualizada.

Dada la ubicación geográfica de los polígonos en la vasta extensión del departamento del Amazonas, se torna fundamental emprender un análisis visual y analítico que permita discernir patrones y características distintivas. Con un total de 160 polígonos en consideración, se torna pertinente realizar una segmentación que facilite la comprensión y el análisis pormenorizado de su distribución espacial. La siguiente tabla proporciona una representación detallada de la disposición y distribución geográfica de estos polígonos, ofreciendo así una visión clara de la configuración y concentración de las áreas de interés en el contexto geográfico del departamento del Amazonas. Esta segmentación contribuirá a una exploración más efectiva de las particularidades asociadas a cada grupo de polígonos, permitiendo una evaluación más exhaustiva y contextualizada.

Grupos	Número de polígonos	COD_MUNICIPIO	
1	7	91405	NA
2	150	91001	91540
3	3	91798	NA

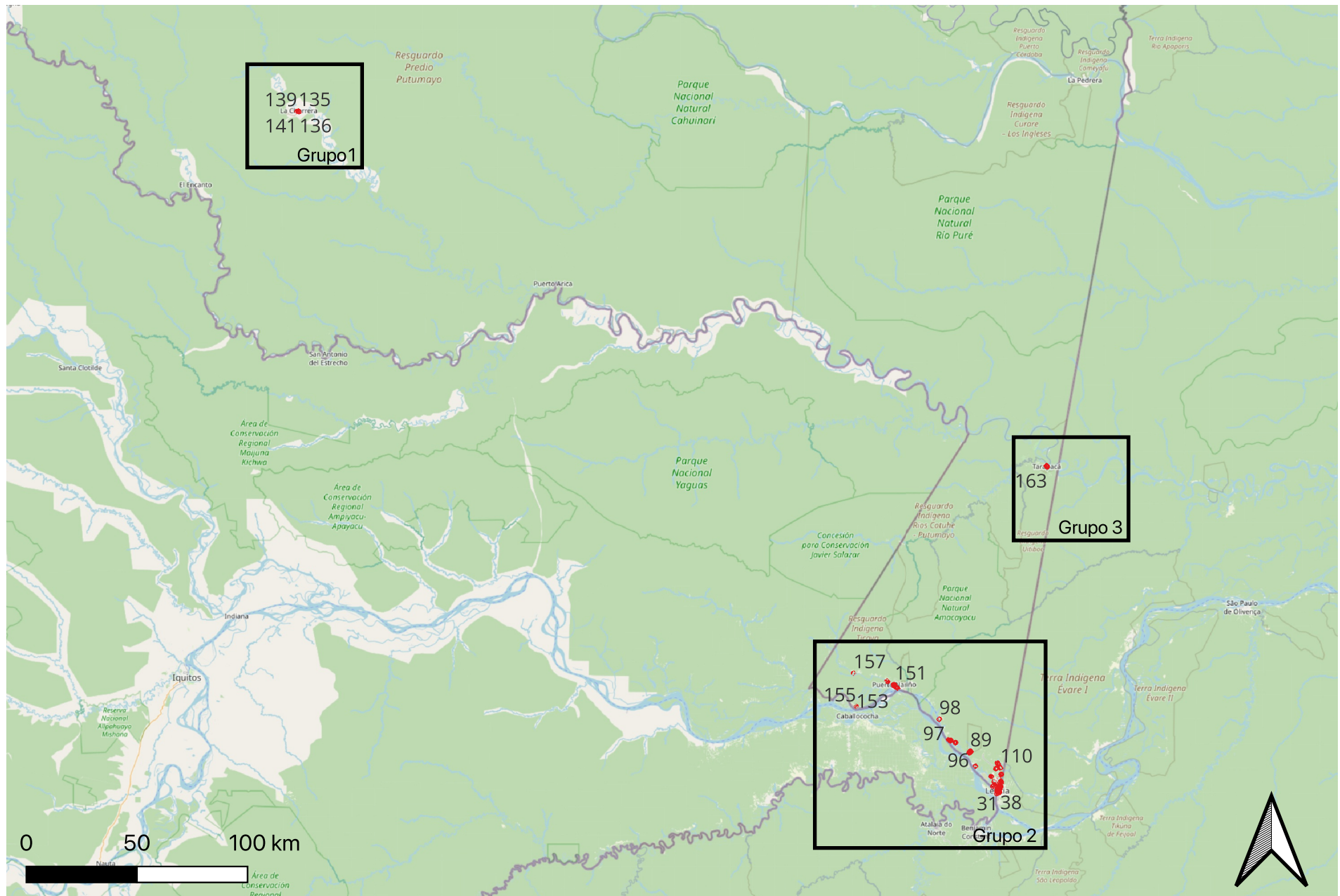
Se evidencia que el grupo 2 de polígonos está mayormente conformado por áreas de interés, y a su vez, se destaca que este grupo establece una conexión entre dos municipios. A continuación, se presentan mapas que ilustran la distribución geográfica de cada grupo, así como la segmentación utilizada para el análisis, sin embargo, debido a la cantidad de mapas generados en el grupo 2, se dejará el enlace local en los anexos, [Apéndice B](#) para visualizar los mapas.

6.1. Asociación de variables demográficas

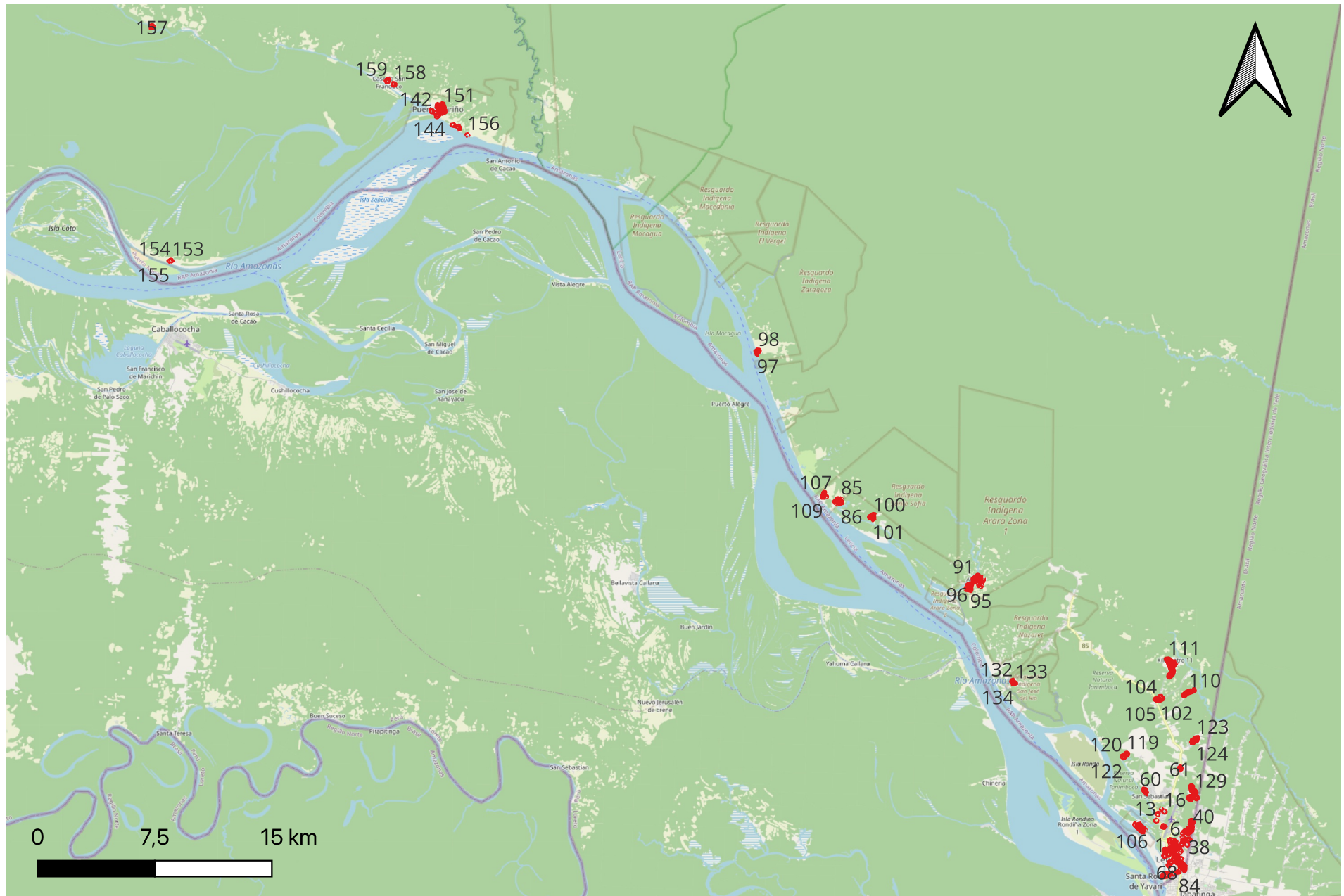
Considerando inicialmente la limitación de contar únicamente con dos variables vinculadas a los polígonos, a saber, el número total de viviendas y las viviendas sin respuestas, surgió la propuesta de enriquecer el DataFrame del archivo shape incorporando nuevas variables que proporcionaran una perspectiva más completa y rica en información sobre los polígonos. Dentro de las variables identificadas para esta ampliación se encontraba el índice de pobreza multidimensional (IPM).

Sin embargo, la inclusión del IPM no estuvo exenta de complejidades. Este índice aborda la pobreza desde cinco dimensiones distintas, las cuales se detallarán a continuación. Una de las complicaciones inherentes al IPM radica en que, cuando el polígono se sitúa en una zona periférica al casco urbano, su asignación se realiza a nivel municipal. Esto significa que el IPM asociado no refleja la situación específica de la zona, sino más bien la del municipio en su conjunto. Esta particularidad del IPM puede generar un efecto de baja incidencia como factor de no respuesta, ya que no captura con precisión la realidad de los polígonos ubicados en zonas periféricas.

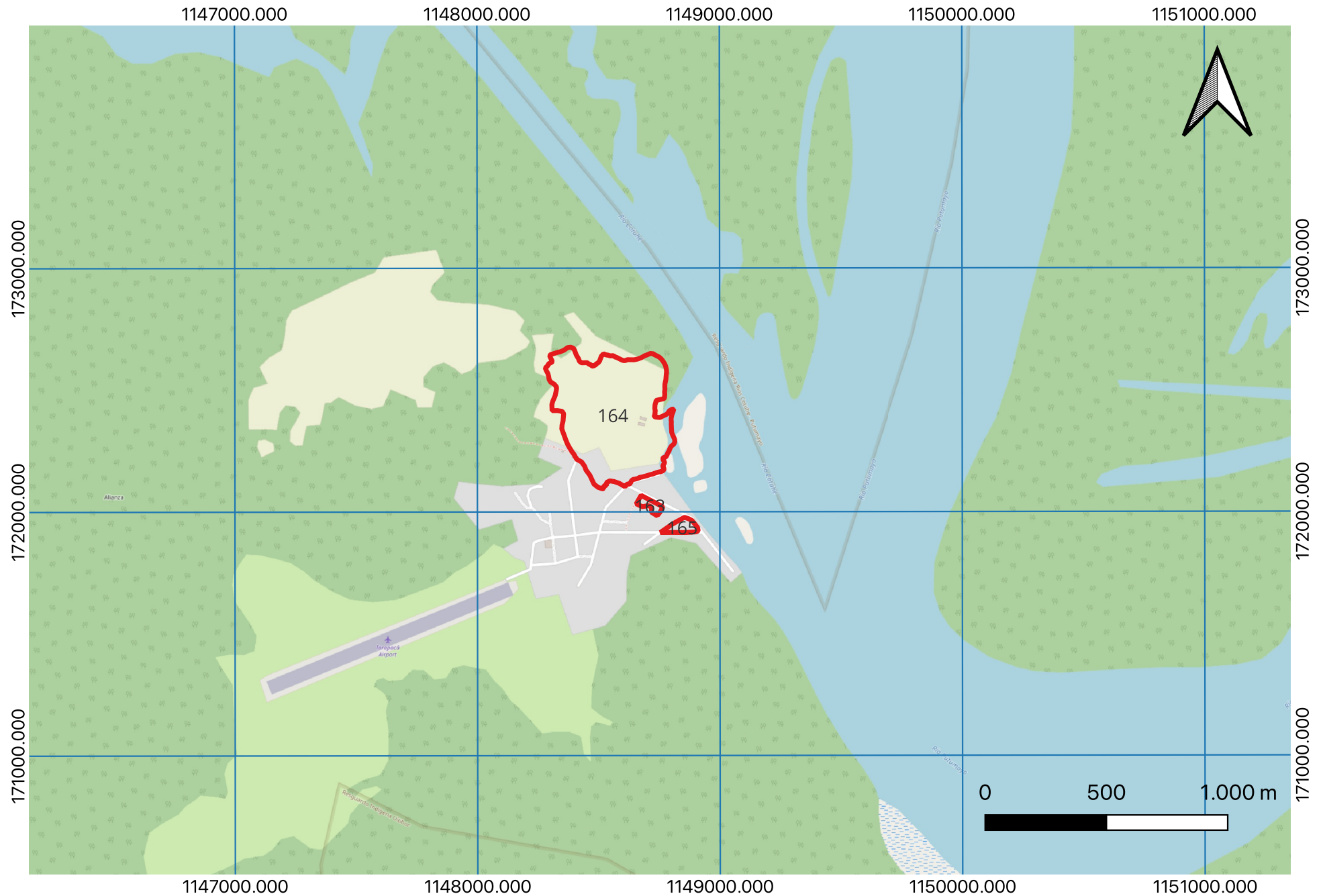
Mapa Completo de polígonos



Mapa G2: Todos los Polígonos



Mapa G3: Todos los Polígonos

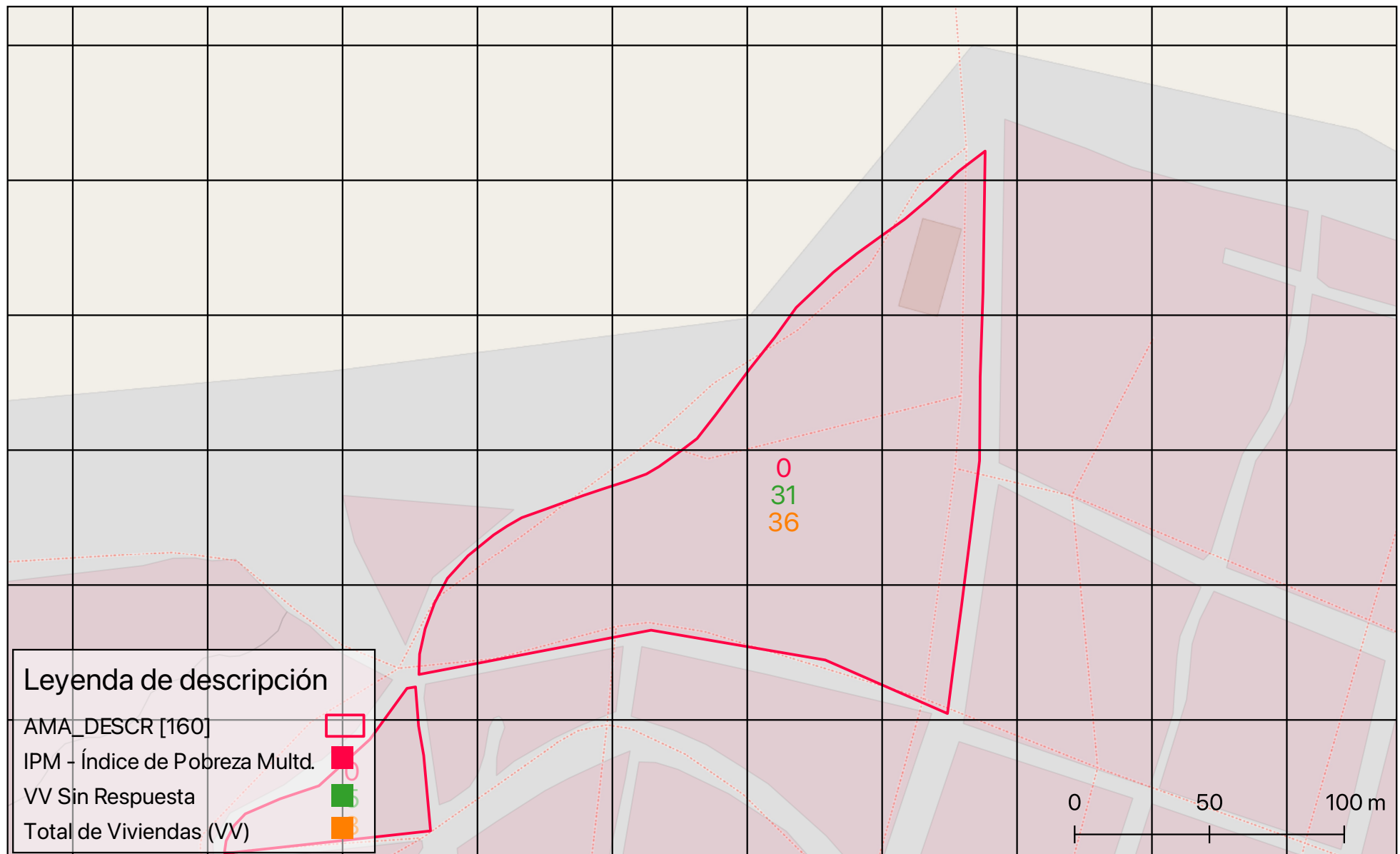


Segmentación de Mapas - Grupos

Grupo: 1

Número de Polígono: 130

Zona: 0

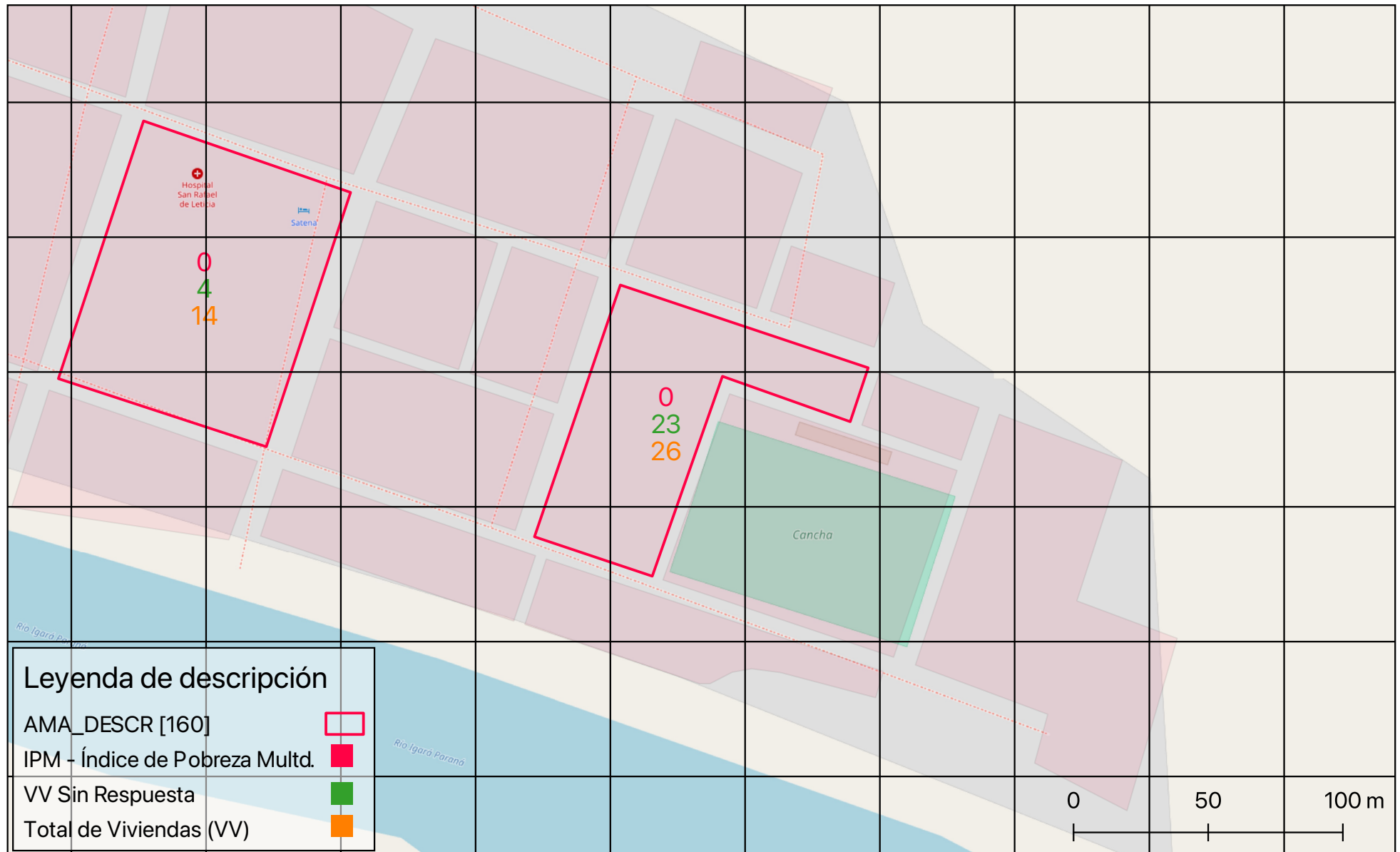


Segmentación de Mapas - Grupos

Grupo: 1

Número de Polígono: 131

Zona: 0

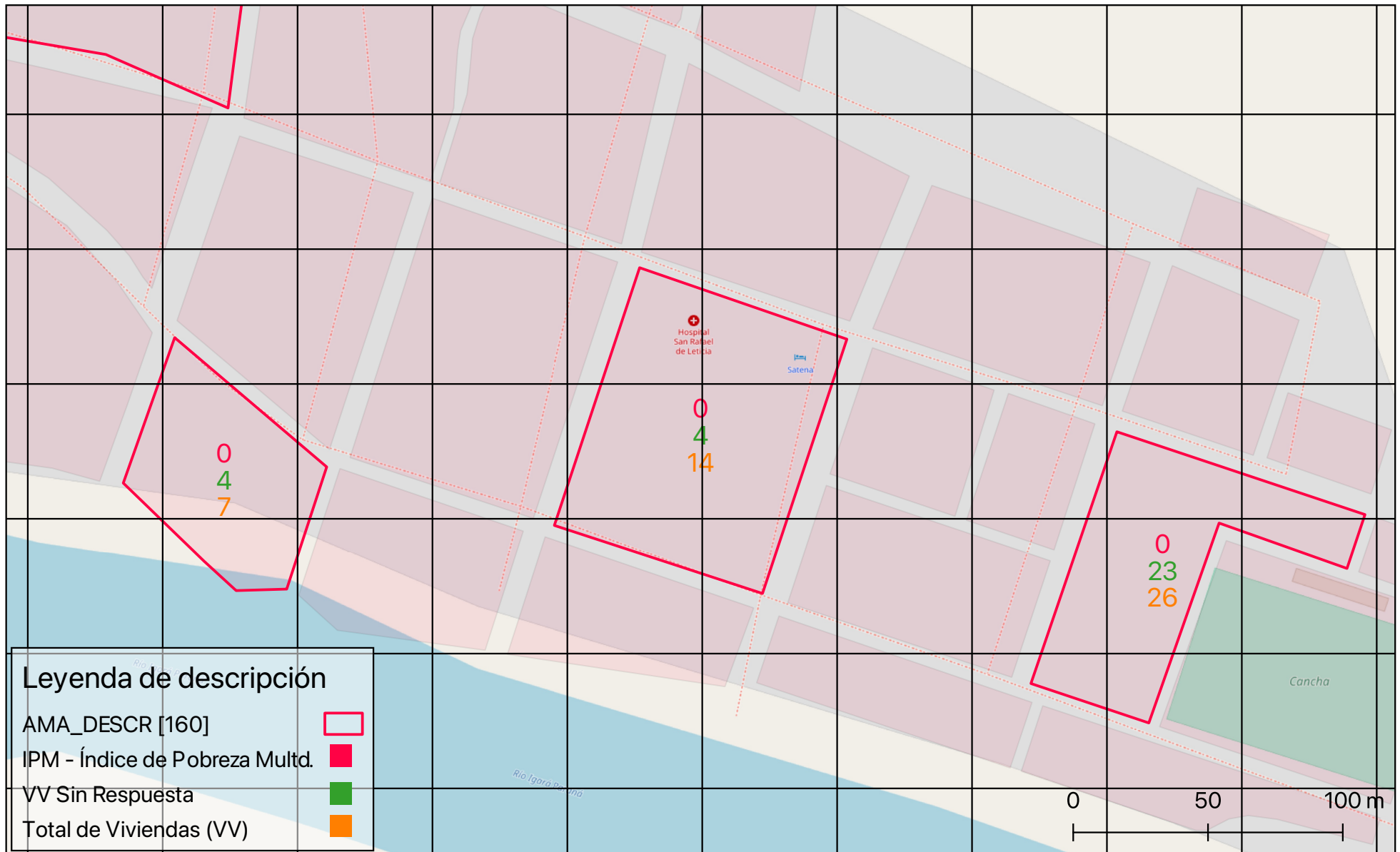


Segmentación de Mapas - Grupos

Grupo: 1

Número de Polígono: 132

Zona: 0

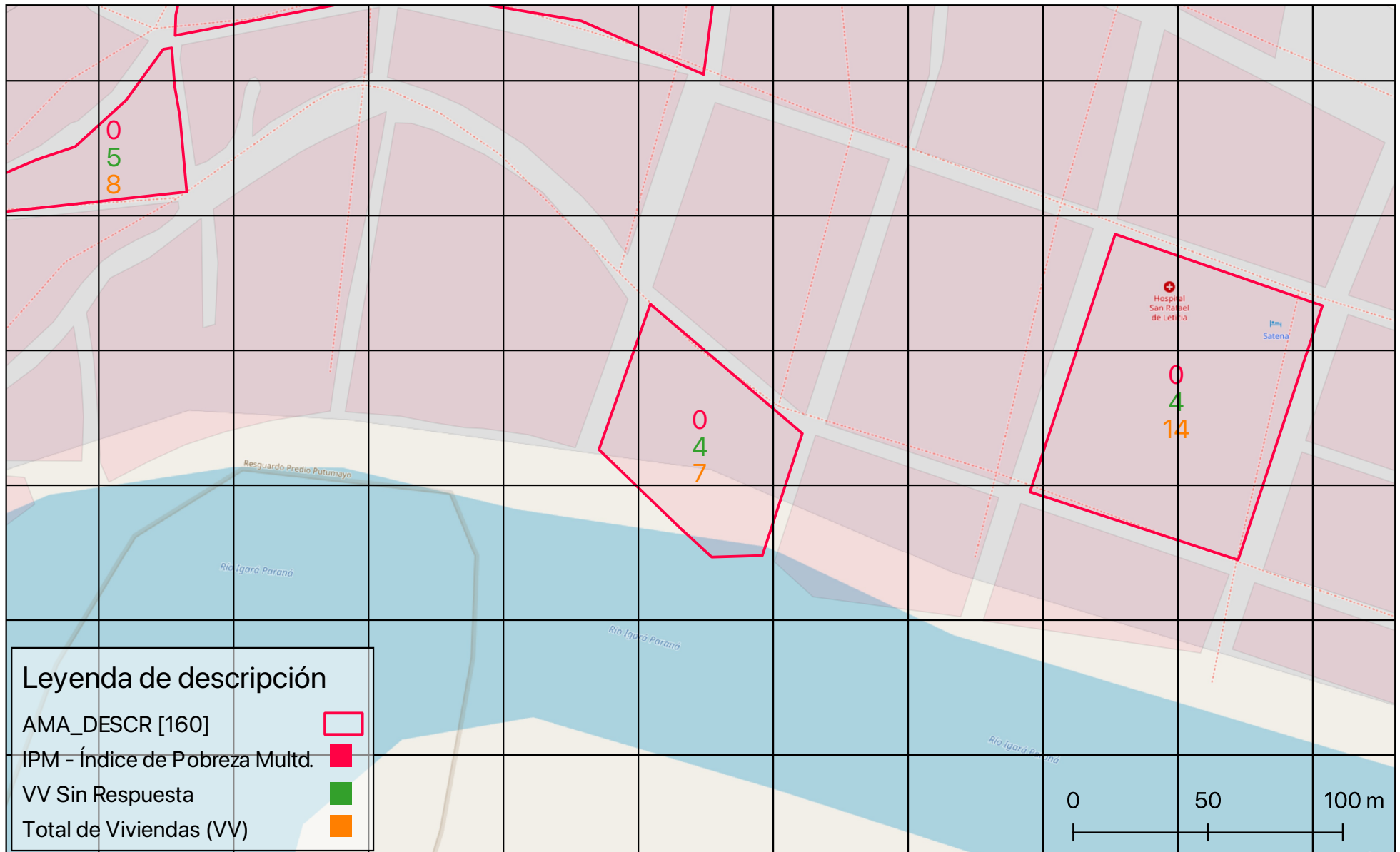


Segmentación de Mapas - Grupos

Grupo: 1

Número de Polígono: 133

Zona: 0

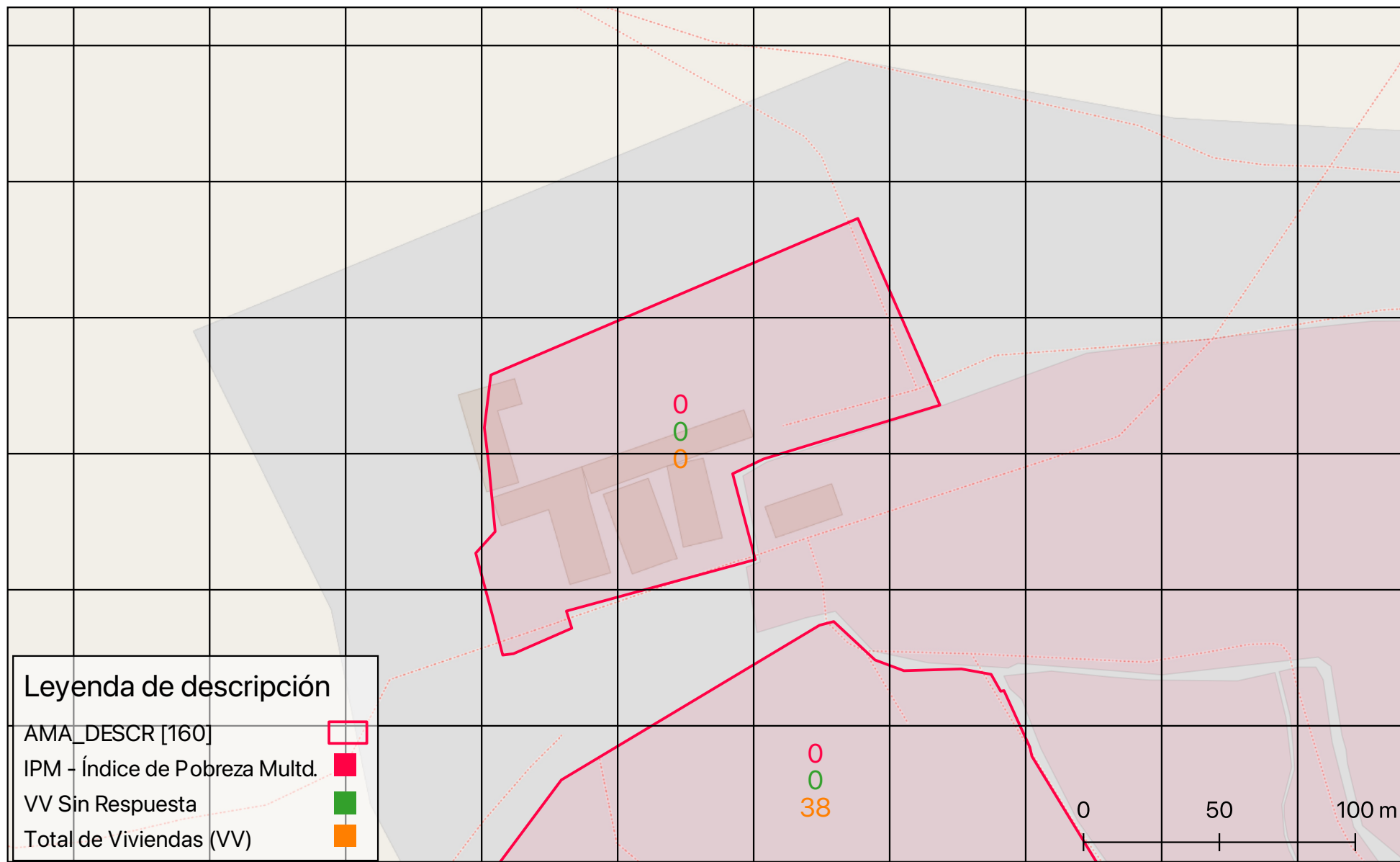


Segmentación de Mapas - Grupos

Grupo: 1

Número de Polígono: 134

Zona: 0

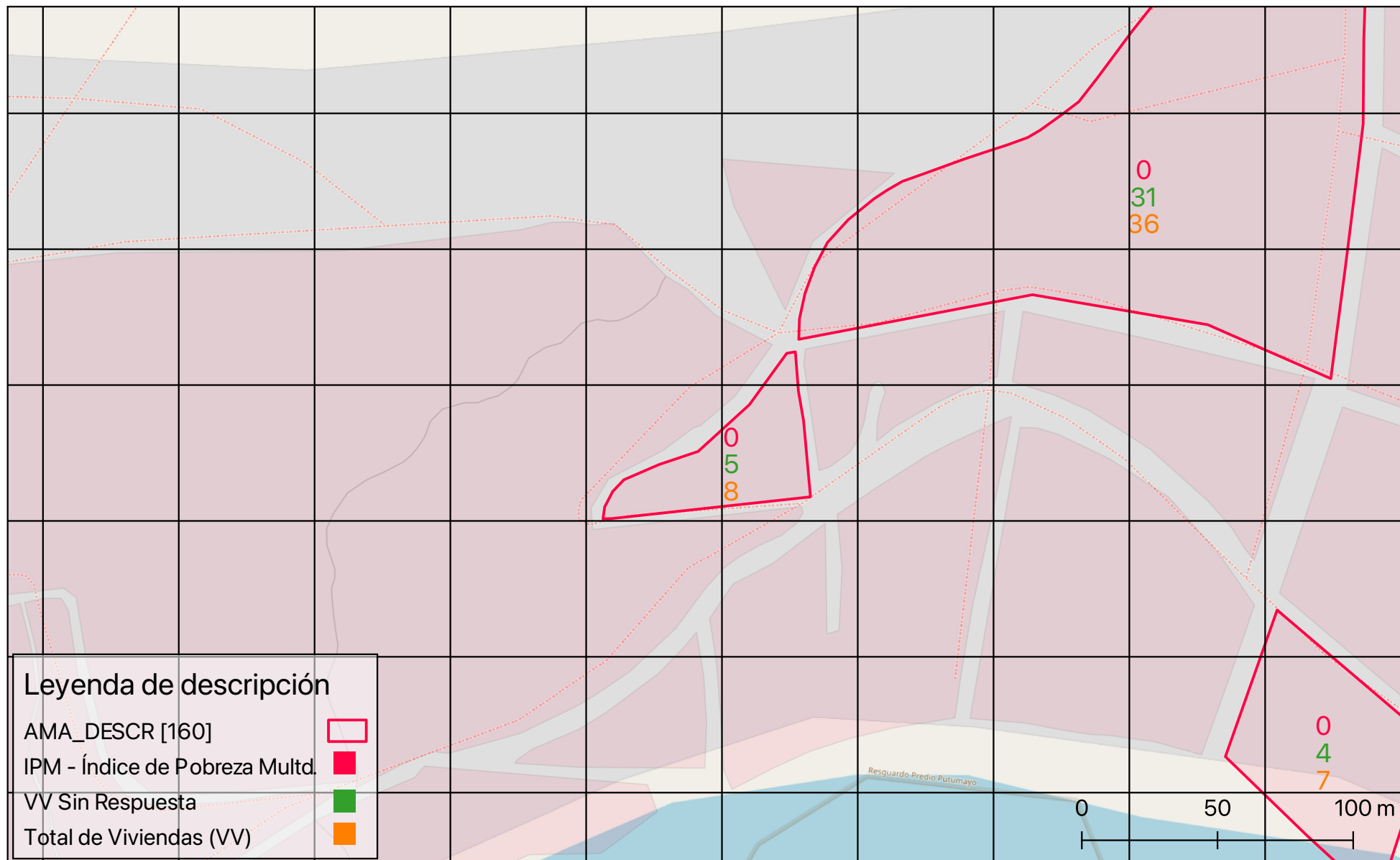


Segmentación de Mapas - Grupos

Grupo: 1

Número de Polígono: 135

Zona: 0

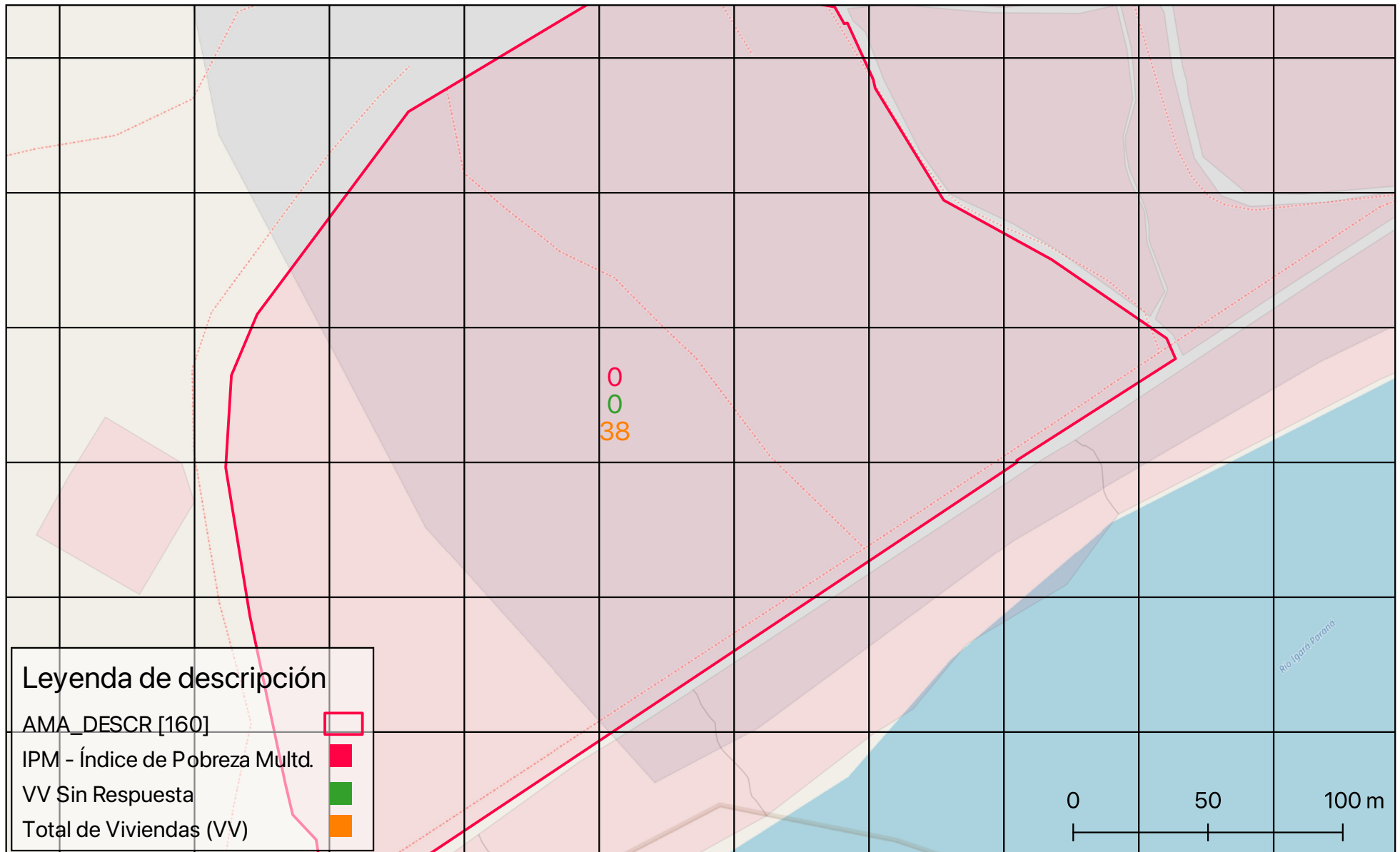


Segmentación de Mapas - Grupos

Grupo: 1

Número de Polígono: 136

Zona: 0

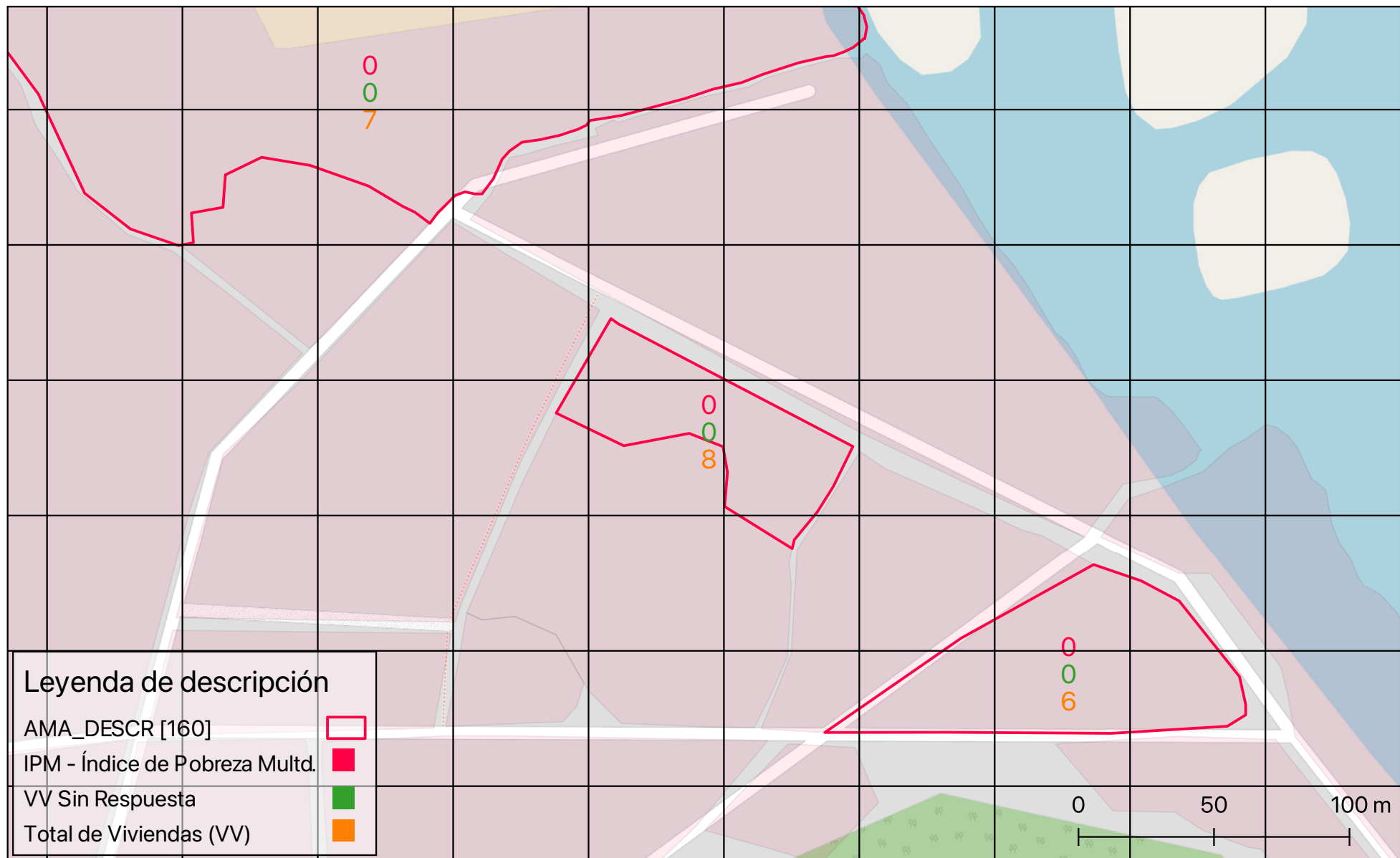


Segmentación de Mapas - Grupos

Grupo: 3

Número de Polígono: 158

Zona: 0

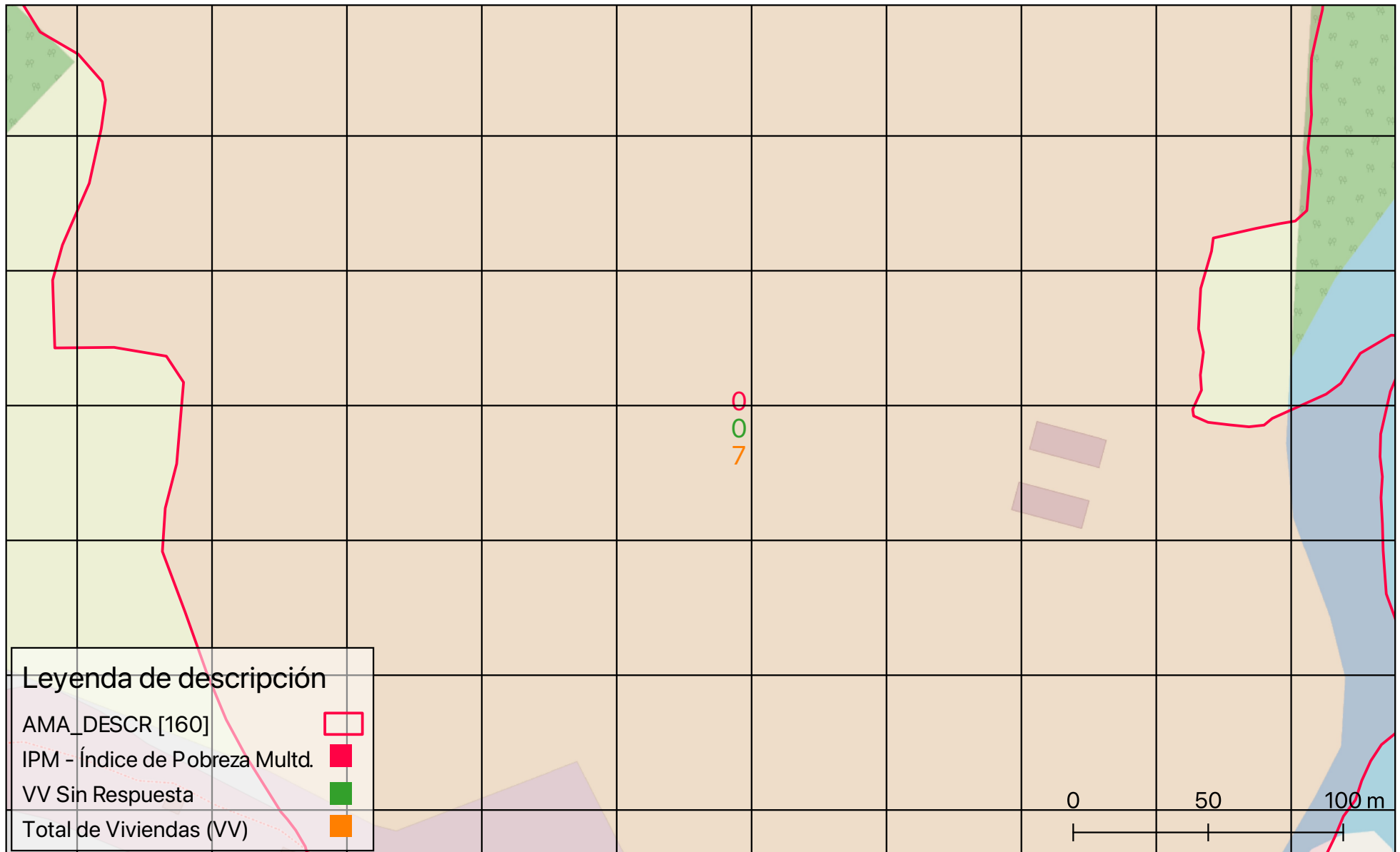


Segmentación de Mapas - Grupos

Grupo: 3

Número de Polígono: 159

Zona: 0

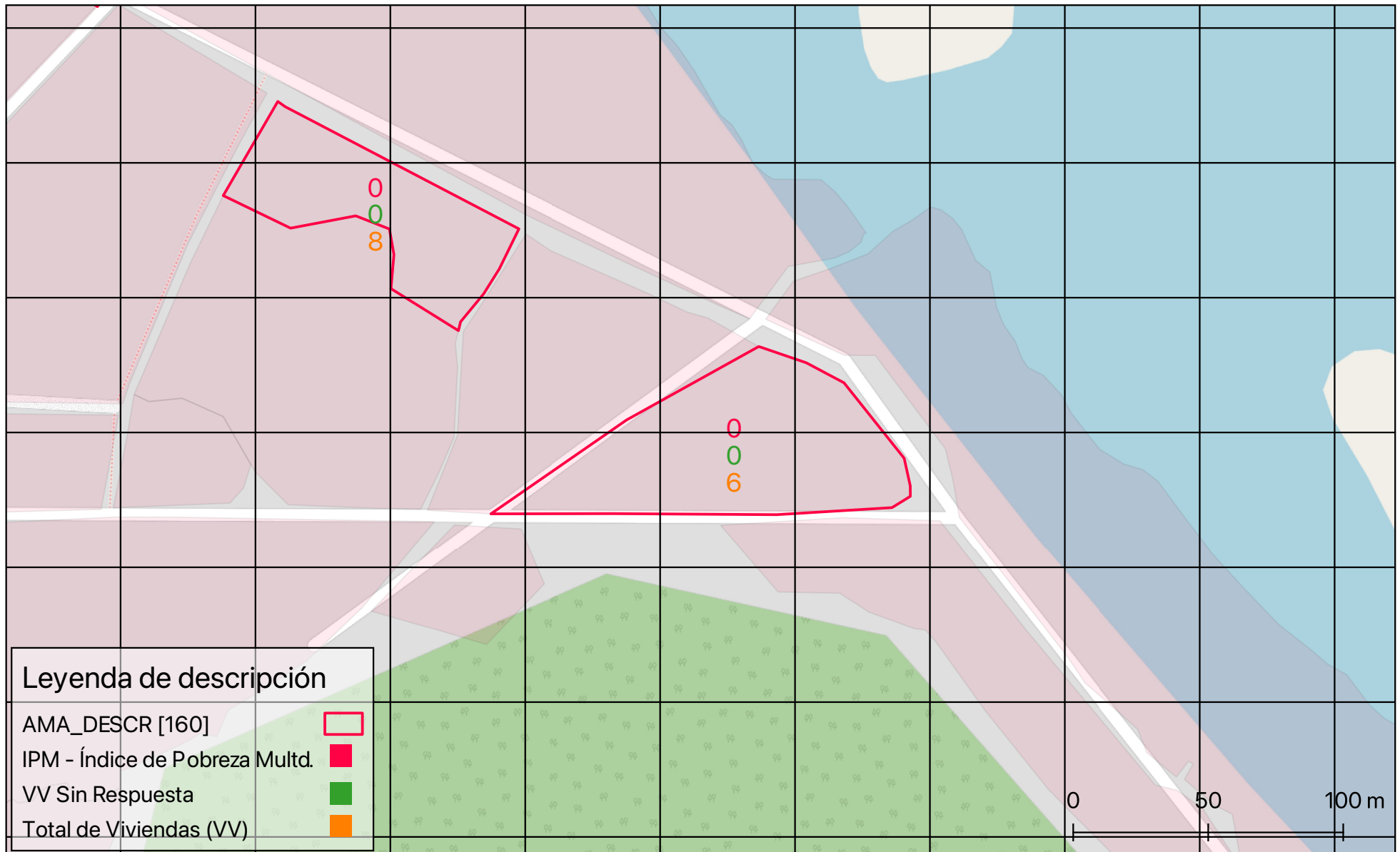


Segmentación de Mapas - Grupos

Grupo: 3

Número de Polígono: 160

Zona: 0



6.1.1. Índice de Pobreza Multidimensional

Descripción del IPM: El Índice de Pobreza Multidimensional (IPM) en Colombia es una medida que va más allá de la medición tradicional de la pobreza basada únicamente en el ingreso. En Colombia, el IPM se utiliza para evaluar la pobreza desde una perspectiva multidimensional, teniendo en cuenta una variedad de factores que afectan la calidad de vida de las personas. A través de esta métrica, se busca comprender la privación en áreas esenciales como la salud, la educación, el empleo, el acceso a servicios básicos y la vivienda. El IPM proporciona una visual sobre la pobreza en el país y permite diseñar políticas y programas más efectivos para mejorar las condiciones de vida de la población [1].

Dimensiones del IPM en Colombia:

1. **Salud:** Evalúa el acceso a servicios de atención médica y la prevalencia de enfermedades.
2. **Educación:** Analiza la asistencia escolar y el nivel educativo de los miembros del hogar.
3. **Empleo:** Considera la presencia de empleo en el hogar y la calidad de dicho empleo.
4. **Acceso a Servicios Básicos:** Incluye la disponibilidad de servicios como electricidad, agua potable, saneamiento básico y vivienda adecuada.
5. **Vivienda y Entorno Comunitario:** Evalúa las condiciones de la vivienda y su ubicación en áreas urbanas o rurales.
6. **Niñez y Juventud:** Mide la privación en términos de nutrición y acceso a la educación.
7. **Adultos Mayores:** Considera el bienestar de los adultos mayores en el hogar.

Estas dimensiones son fundamentales para comprender parcialmente la pobreza y permiten abordar las necesidades de las comunidades de manera más integral y eficiente.

6.1.2. Variables de hacinamiento y miseria

La visualización de los mapas segmentados por grupos revela una baja incidencia y, en algunos casos, la ausencia de respuesta en relación con la variable del Índice de Pobreza Multidimensional (IPM). Este hallazgo sugiere que el IPM puede no ser una variable que aporte información sustancialmente relevante para nuestro análisis. A pesar de esta limitación, se ha logrado establecer conexiones con otras variables de índole demográfica, las cuales tienen el potencial de ofrecer valiosas perspectivas relacionadas con la no respuesta.

Es crucial señalar que estas variables adicionales no están directamente vinculadas a los polígonos específicos, sino más bien a los municipios a los cuales corresponden dichos polígonos. Entre las variables que podrían enriquecer nuestro análisis descriptivo, se encuentran el Índice de Hacinamiento y el Índice de Miseria. Estas variables demográficas podrían proporcionar una comprensión más detallada de los factores asociados a la no respuesta, aportando así una dimensión más completa a nuestro análisis.

En los mapas, se destaca que el grupo 2 de polígonos exhibe la mayor ausencia de respuestas. A pesar de este hallazgo, aún carecemos de información relevante para formular hipótesis sobre las posibles razones detrás de la no respuesta en este grupo. A continuación, se presenta una tabla que contrasta, por municipio, las variables consideradas pertinentes para el análisis de esta variable asociada a la no respuesta.

COD_MPIO	MUNICIPIO	HACINAMIENTO (%)	MISERIA (%)	NUM. POLI	GRP
91001	LETICIA	14,31	6,27	129	2
91405	LA CHORRERA	14,67	9,78	7	1
91540	PUERTO NARINO	13,92	11,94	21	2
91798	TARAPACA	13,74	15,93	3	3

7. Conclusiones

En conclusión, el presente proyecto abordó el desafío empresarial de optimizar el tiempo dedicado a la generación de mapas con características cartográficas específicas. Inicialmente, se exploró la opción de diseñar un código en Python para la automatización, enfrentando desafíos relacionados con la consola de ArcMap, que limitaban la eficiencia y calidad de las exportaciones.

Posteriormente, se consideró la opción de utilizar la función Atlas en QGIS, una herramienta que ofrece una solución más integral y eficiente para la generación automatizada de series de mapas. A pesar de novedades en la exportación, en código en Python, como la dispersión de polígonos, esta opción Atlas se reveló como una estrategia más viable.

Además, se proporcionaron recursos adicionales, como video tutoriales, que detallan cada etapa del procesamiento de mapas mediante la función Atlas en QGIS. Se exploraron variables asociadas a los polígonos de muestra, como el índice de pobreza multidimensional (IPM), y se identificaron limitaciones, como la asignación municipal del IPM.

En resumen, el proyecto no solo ofreció soluciones prácticas y eficientes para la generación automatizada de mapas, sino que también destacó la importancia de explorar y comprender detalladamente las herramientas disponibles en el ámbito geoespacial para abordar retos específicos.

Referencias

- [1] DE MÁQUINAS, E. I. S. Índice de pobreza multidimensional.
- [2] LI, Z., CHENG, Y., ZHANG, L., AND YANG, Y. A review of gis-based urban evacuation models: Challenges and opportunities. *ISPRS International Journal of Geo-Information* 9, 1 (2020), 22.
- [3] MACHADO, J. A. T., PIRES, L. F., AND MARUJO, R. F. A gis-based approach for quantifying the impacts of land use changes on the frequency of urban flooding. *Journal of Hydrology* 566 (2018), 554–565.

A. Anexo A: Código en Python para la exportación de Mapas

A continuación encontrará el código inicial para la exportación de mapas desde Python. [Archivo Código](#) y [Readme](#).

```
# AUTOMATIZACIÓN EN ARCMAP-PYTHON
# EXTRACCIÓN DE MAPAS - CARACTERISTICAS ECO
# NOMBRE : EXTRACCIONES_DE_POLIGONOS
# Autor : Joel Fernando Morera Robles
# Fecha Creado : 160823
# Ultimo Cambio : 231023
# Por : Joel Fernando Morera Robles

"""
Descripcion:
La automatización es un proceso cuyo objetivo es la optimización de tiempo en el procesamiento de in
la automatización en programación Python se refiere al desarrollo de scripts o programas que ejecu
automáticamente sin intervención manual. Utilizando bibliotecas de Python, se accede a datos, se r
y se toman decisiones lógicas para automatizar tareas repetitivas. El proceso involucra identifica
escribir el código, probarlo y programar su ejecución periódica si es necesario. La automatización
de tiempo y mejora la eficiencia al eliminar tareas manuales y evitar errores humanos.
En particular, en este proyecto se busca diseñar un algoritmo en lenguaje Python con el objetivo d
manuales inherentes al proceso de exportación de mapas con características particulares, tanto de
los datos cruzados y geolocalizados, con el objetivo de optimización de recursos y tiempo, adición
contener protocolos de actualización del código para futuras tareas corporativas. Para el desarrol
se cuenta con un tiempo límite de tres meses.

Este archivo compila la versión final de Proyect_ArcMap.py

Exe_CODE: El código proporciona una herramienta eficaz para la generación de archivos PDF personal
geoespaciales almacenados en un archivo Shapefile. Utilizando una variedad de bibliotecas de Pytho
específicos en mapas geográficos al tiempo que permite un alto grado de personalización en término
tamaños y estilos de los polígonos. Además, se integra una funcionalidad que verifica si los mapas
proporcionando advertencias en caso de problemas de ajuste. Este código simplifica el proceso de e
resaltados en un formato PDF, brindando a los usuarios la capacidad de comunicar y compartir visua
de manera efectiva. El PDF generado por el código contiene una serie de páginas, una por cada polí
el archivo Shapefile de entrada. Cada página del PDF incluye:
1. Un mapa geoespacial: El polígono de interés se resalta en un mapa geográfico, permitiendo la v
de su ubicación en contexto.
2. Números de identificación: Cada polígono se etiqueta con un número de identificación que fac
3. Título: El título del PDF se muestra en la parte superior de cada página, proporcionando un
los mapas generados.

"""

from fpdf import FPDF
from fpdf.enums import XPos
from fpdf.enums import YPos
import matplotlib.pyplot as plt
import geopandas as gpd
import contextily as ctx
from tqdm import tqdm
```

```

from PIL import Image
from PIL import ImageOps

# Define las rutas de los archivos
shapefile_path = "/Users/joelmorera/Documents/Proyecto USTA-ECONO/etapa1/input/MGN_2021_COLOMBIA/URB
output_pdf_path = "/Users/joelmorera/Documents/PDFs_poligonos/poligonos_destacados.pdf"

def cargar_shapefile(shapefile_path):
    """
    Carga un shapefile y lo proyecta en el sistema de coordenadas adecuado.

    Args:
        shapefile_path (str): Ruta del archivo shapefile.

    Returns:
        gpd.GeoDataFrame: GeoDataFrame cargado desde el shapefile.
        str: Sistema de coordenadas proyectadas.
    """
    print("Iniciando la carga del shapefile...")

    try:
        shapefile = gpd.read_file(shapefile_path)
        proyecto_crs = 'EPSG:21897'

        # Cambiar el CRS del GeoDataFrame al sistema de coordenadas proyectadas adecuado
        shapefile = shapefile.to_crs(proyecto_crs)

        print("Carga del shapefile completada.")

    except Exception as e:
        print(f"Error al cargar el shapefile: {e}")
        shapefile = None
        proyecto_crs = None

    return shapefile, proyecto_crs

def generar_poligonos_en_pdf(shapefile, output_pdf_path, proyecto_crs):
    """
    Genera un PDF con mapas de polígonos resaltados.

    Args:
        shapefile (gpd.GeoDataFrame): GeoDataFrame con los datos de polígonos.
        output_pdf_path (str): Ruta de salida para el archivo PDF.
        proyecto_crs (str): Sistema de coordenadas proyectadas.

    Returns:
        None
    """
    if shapefile is None:
        print("No se pudo cargar el shapefile. No se generará el PDF.")
        return

```

```

# Obtén el número total de polígonos
total_poligonos = len(shapefile)

# Crear un objeto PDF
pdf = FPDF()
pdf.set_auto_page_break(auto=True, margin=15)

# Crear una barra de progreso para la generación del PDF
pdf_bar = tqdm(total=total_poligonos, desc="Generando PDF", unit=" página(s)")

# Título
titulo = "Automatización de procesos de extracción en Python"
pdf.set_font("Helvetica", size=16)

for i, polygon in enumerate(shapefile.geometry):
    if polygon.area > 0:
        pdf.add_page()

        # Crear una figura de Matplotlib con un tamaño personalizado
        fig, ax = plt.subplots(figsize=(8, 8))

        # Configurar los límites del mapa
        bounds = shapefile.total_bounds
        ax.set_xlim(bounds[0], bounds[2]) # Límites x (longitud)
        ax.set_ylim(bounds[1], bounds[3]) # Límites y (latitud)

        # Dibujar los polígonos
        for j, poly in enumerate(shapefile.geometry):
            color = 'red' if j == i else 'black'
            linewidth = 1.0 if j == i else 0.5
            gpd.GeoSeries(poly).plot(ax=ax, color='none', edgecolor=color, linewidth=linewidth,

            # Obtener el centro del polígono y agregar el número
            centro = poly.centroid
            label = str(j + 1)
            if j == i:
                ax.text(centro.x, centro.y, label, ha='left', va='center', fontsize=12, color='r')
            else:
                ax.text(centro.x, centro.y, label, ha='left', va='center', fontsize=12, color='b')

        # Agregar OpenStreetMap como fondo utilizando contextily
        ctx.add_basemap(ax, crs=proyecto_crs, source=ctx.providers.OpenStreetMap.Mapnik)

        # Eliminar ejes y etiquetas
        ax.set_axis_off()

        # Guardar la figura como una imagen temporal
        temp_image_filename = f"temp_image_{i}.png"
        fig.savefig(temp_image_filename, format='png', bbox_inches='tight', pad_inches=0, dpi=50)
        plt.close(fig)

        # Obtén el ancho y alto de la página
        pdf_w = pdf.w

```

```
pdf_h = pdf.h

# Título de la página en mayúsculas
titulo = "Automatización de procesos de extracción en Python"
pdf.set_font("Helvetica", size=16)
pdf.cell(200, 10, text=titulo, align="C")
pdf.ln()

numero_poligono = f"NÚMERO DE POLÍGONO: {str(i + 1).zfill(4)}.upper()"
pdf.set_font("Helvetica", size=14, style="B")
pdf.set_xy(10, pdf.get_y()) # Alinea a la izquierda
pdf.cell(200, 10, text=numero_poligono, align="L")
pdf.ln()

# Calcular la posición x para centrar horizontalmente la imagen
img_w = pdf_w * 0.65
x = (pdf_w - img_w) / 2

# Calcular la posición y para centrar verticalmente en la línea del medio
img_h = img_w # Suponiendo que la imagen tenga las mismas dimensiones de ancho y alto
y = 30

# Verificar si el mapa se sale de la página
if x < 0 or x + img_w > pdf_w or y < 0 or y + img_h > pdf_h:
    print("Advertencia: El mapa no cabe en la página. Ajusta el tamaño del mapa o la pág")
    return # Detiene la generación del PDF y sale de la función

# Insertar la imagen en el PDF
pdf.image(temp_image_filename, x=x, y=y, w=img_w)

# Actualizar la barra de progreso de la generación del PDF
pdf_bar.update(1)

# Cerrar la barra de progreso de la generación del PDF
pdf_bar.close()

# Guardar el PDF
pdf.output(output_pdf_path)

print(f"Archivo PDF generado exitosamente con polígonos destacados en {output_pdf_path}")

# Cargar el shapefile y obtener el CRS
shapefile, proyecto_crs = cargar_shapefile(shapefile_path)

# PARAMETROS PARA MUESTRA - Pruebas operativas
# Cargar el shapefile y obtener el CRS
shapefile_sample = shapefile.iloc[500021:500031]
# Generar polígonos en el PDF
generar_poligonos_en_pdf(shapefile_sample, output_pdf_path, proyecto_crs)
```

B. Anexo B: Resultados Adicionales

A continuación enlisto las exportaciones finales, generadas en QGIS LTR con la función Atlas, como los enlaces de los PDF con los mapas generados para el análisis geoespacial de descripción.

1. [Ejemplo de Exportación](#)
2. [Segmentación de mapa para Grupo 1](#)
3. [Segmentación de mapa para Grupo 2](#)
4. [Segmentación de mapa para Grupo 3](#)