UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DESARROLLO DE UNA INTERFAZ PARA LA CONSULTA Y VISUALIZACIÓN DE DATOS DE SALUD PÚBLICA DE LOS RESÚMENES ESTADÍSTICOS MENSUALES DEL MINSAL

Por: Maximiliano Agustín Araya Morales

Informe de Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al grado académico de Ingeniero/a Civil Biomédica

Agosto 2024
Concepción, Chile

Profesora Guía
Pamela Guevara Alvez
Supervisor externo
Jaime Jiménez Ruiz
Comisión
Rosa Figueroa Iturrieta

Resumen

El objetivo de esta memoria es desarrollar una interfaz para la consulta y visualización de los datos de salud pública contenidos en los Resúmenes Estadísticos Mensuales (REM) emitidos por el Ministerio de Salud de Chile (MINSAL). Esta memoria está patrocinada por la empresa privada HealthTracker.

El proyecto se enmarca en la necesidad de mejorar la gestión y análisis de los datos de salud, los cuales se encuentran dispersos y en distintos formatos, dificultando su utilización. Para abordar este desafío, se plantearon varias etapas que incluyen la descarga y consolidación de archivos REM, el análisis de los diccionarios de datos, la generación de un formato decodificador en JSON, y la implementación de una interfaz para la consulta de estos datos, la cual busca mejorar la accesibilidad y manejo de los datos de la Serie A de los REM, que abarcan desde 2009 hasta 2023.

Durante el desarrollo, se utilizaron técnicas como el método de chunks para manejar grandes volúmenes de datos, lo que permitió procesar múltiples cortes que conformaban hasta 500 MB cada uno, sin agotar la memoria del sistema. Además, se emplearon herramientas avanzadas de LLM como Chat GPT 4-O para la generación de descriptores y junto a métodos de cruce para la normalización de datos, lo que agilizó el proceso de decodificación.

Los resultados obtenidos incluyen una base de datos consolidada de los archivos REM de la Serie A, un formato decodificador en JSON que facilita la consulta y análisis de los datos a través de los años, y una interfaz interactiva desarrollada con Streamlit, que permite realizar consultas SQL.

Se lograron los objetivos, como la integración de datos de diferentes años y formatos, y la implementación de filtros avanzados para asegurar consultas. Significando un desarrollo en la accesibilidad de los datos abiertos de salud pública en Chile.

Abstract

The objective of this thesis is to develop an interface for querying and visualizing public health data contained in the Monthly Statistical Summaries (REM) issued by the Ministry of Health of Chile (MINSAL). This thesis is sponsored by the private company HealthTracker.

The project addresses the need to improve the management and analysis of health data, which are currently dispersed and in various formats, making their use challenging. To tackle this challenge, several stages were outlined, including the downloading and consolidation of REM files, the analysis of data dictionaries, the generation of a JSON decoding format, and the implementation of an interface for querying this data. The aim is to enhance the accessibility and management of the Series A REM data, covering the years 2009 to 2023.

During the development, techniques such as the chunk method were used to handle large volumes of data, allowing the processing of multiple segments, each up to 500 MB, without exhausting system memory. Additionally, advanced LLM tools like Chat GPT 4-O were employed for descriptor generation, along with cross-referencing methods for data normalization, which streamlined the decoding process.

The results include a consolidated database of Series A REM files, a JSON decoding format that facilitates the querying and analysis of data over the years, and an interactive interface developed with Streamlit that allows for SQL queries.

The objectives were achieved, including the integration of data from different years and formats, and the implementation of advanced filters to ensure precise queries, marking a significant advancement in the accessibility of public health open data in Chile.

Índice General

Índice de Tablas

Índice de Figuras

Capítulo 1

Introducción.

1.1. Introducción General.

La siguiente memoria aborda la problemática de la búsqueda y decodificación de los archivos REM (Resúmenes Estadísticos Mensuales) (1), documentos clave en la salud pública que se dividen en cinco series; A, P, BS, BM y D. La Serie A documenta controles y atenciones sanitarias en diferentes etapas de la vida, incluyendo temas como la salud sexual y reproductiva, controles por ciclo vital y evaluaciones específicas; mientras que la Serie P se enfoca en la atención sanitaria en centros de atención primaria, abarcando la atención a mujeres, niños, adolescentes y ancianos. Estos archivos contienen datos y diccionarios para la decodificación y el uso en la planificación y gestión en el sector de la salud. La Serie BM abarca prestaciones de apoyo diagnóstico y terapéutico, incluyendo exámenes de diagnóstico, procedimientos clínicos, y atenciones odontológicas. La Serie BS se enfoca en la atención abierta y cerrada, registrando actividades de apoyo diagnóstico y terapéutico en modalidades ambulatorias y de hospitalización, así como intervenciones quirúrgicas y atenciones de salud mental. La Serie D incluye estadísticas misceláneas y específicas, cubriendo programas y actividades de rehabilitación y otros aspectos no categorizados en las series anteriores, contribuyendo a la estandarización y monitoreo de la información de salud a nivel nacional (2).

El proceso de manejo de los archivos REM consiste en múltiples pasos y requiere considerar archivos adicionales como listados de establecimientos y diccionarios, lo

que complica la extracción de información específica, debido a que, la información contenida en la base de datos de los REM está codificada y por esto mismo son necesarios archivos adicionales para entender mejor estos valores, como también, no hay manera de considerar varios parámetros para la búsqueda. Originados para fortalecer el principio estadístico de que *"la actividad realizada se registra donde se hace"* (2), los archivos REM son fuente de indicadores de gestión, cumplimiento de programas y toma de decisiones en salud.

Aunque el Ministerio de Salud (MINSAL) de Chile ofrece herramientas de filtrado, estas son limitadas, permitiendo solo acceso a reportes Serie A de 2017-2020 con filtros acotados y poco flexibles (3).

En este contexto, este proyecto propone el desarrollo de una interfaz para la consulta y visualización de los archivos REM de la Serie A (2009-2023), destinada al uso de HealthTracker (4), una empresa privada. La interfaz ofrecerá una herramienta de búsqueda que permitirá una extracción de datos, contribuyendo así a mejorar la gestión de la información en la empresa.

1.2. Objetivo General.

- Desarrollar herramienta de consumo mediante una interfaz para consultar datos codificados de salud pública, específicamente del Resumen Estadístico Mensual (REM) emitido por el MINSAL.

1.3. Objetivos Específicos.

- Exploración de datos de archivos REM obtenidos de la base de datos del MINSAL.
- Definición del formato de decodificación de su diccionario.
- Aplicación de estándar definido utilizando el formato decodificador al diccionario REM.
- Generación de una base de datos del consolidado histórico del REM.
- Implementación de una interfaz para realizar consultas a la base de datos.

1.4. Alcances y limitaciones.

- La base de datos estará basada en archivos REM solo de Series A en el período 2009-2023.
- Las consultas estarán limitadas en base a parámetros contenidos en el formato decodificador.
- La empresa HealthTracker (4), cuenta con acceso a servicios en la nube de Google, y herramientas para desarrollo de Bots, que se utilizarán como insumo para esta memoria de título.

1.5. Metodología.

- **Descarga y consolidación de archivos REM:** Desarrollo en Python para concatenar los archivos REM Series A (2009-2023), normalizando la información en un formato uniforme, con el fin de generar una base de datos que facilite la manipulación y análisis de los datos.
- **Estudio de diccionarios REM Series A y P:** Estudio de los diccionarios de datos, estructura y contenido de las tablas.
- **Generación del formato decodificador Serie A:** Implementación de script en Python para extraer datos de interés de los diccionarios REM y normalización de esta información en un formato estándar JSON. Esto incluye la generación de datos faltantes para los JSON de la Serie A mediante LLM.
- **Implementación interfaz para consulta:** Desarrollo de script de prueba para extraer información de los JSON y realizar cruces con el consolidado de los archivos REM Serie A.

1.6. Temario.

- **Capítulo 1:** Introducción en forma general del trabajo, indicando los objetivos y alcances del proyecto.
- **Capítulo 2:** En esta sección se entrega contexto sobre el tema y las tecnologías usadas en el área. Además, se describe la información más relevante del análisis bibliográfico.
- **Capítulo 3:** En esta sección se detalla la decodificación de manera manual a los archivos REM Series A.
- **Capítulo 4:** En esta sección se detallan la implementación de interfaz para la consulta y visualización de los datos.
- **Capítulo 5:** Presentación de resultados.
- **Capítulo 6:** Conclusión del trabajo realizado, como también, mejoras a futuro del proyecto.

Capítulo 2

Marco Teórico.

2.1. Introducción.

Se aborda la relevancia y el manejo de los Resúmenes Estadísticos Mensuales (REM) proporcionados por el Ministerio de Salud de Chile, indicando su función en temas como monitorización y planificación de los servicios de salud. Se hace enfasís en la Serie A de los REM, que contienen datos sobre la atención sanitaria brindada a nivel nacional.

La implementación de una interfaz para consulta y el uso de tecnologías como Streamlit pueden mitigar las limitaciones presentes en las herramientas de filtrado existentes.

2.2. Archivos REM (Resúmenes Estadísticos Mensuales).

Los archivos REM son compilaciones de datos que cada establecimiento de salud en Chile debe enviar mensualmente al Ministerio de Salud (MINSAL). Estos reportes consolidan información detallada sobre las actividades y servicios brindados por los centros de salud, desde hospitales hasta clínicas locales, incluyendo atención primaria y especializada (1).

- **Recolección de datos:** Los archivos REM recogen un amplio espectro de datos, incluyendo números de consultas médicas, procedimientos realizados,

casos de enfermedades específicas, utilización de servicios, y más. Cada serie dentro del REM se enfoca en diferentes aspectos de la atención sanitaria, como salud reproductiva, enfermedades crónicas, urgencias, entre otros.

- **Monitoreo:** La información recogida permite a las autoridades sanitarias monitorizar la eficiencia y eficacia de los servicios de salud. Ayuda a identificar patrones de enfermedad, demanda de servicios de salud, y efectividad de programas públicos de salud.
- **Gestión y Planificación:** Con estos datos, el MINSAL puede planificar mejor la distribución de recursos, diseñar políticas de salud basadas en evidencia, y ajustar programas para abordar necesidades específicas o emergentes en la población. También facilita la evaluación del cumplimiento de metas sectoriales como parte de las políticas de salud pública.
- **Cumplimientos de Objetivos:** Los REM son esenciales para evaluar el progreso hacia metas nacionales de salud y para asegurar el cumplimiento de convenios y estándares de calidad en los servicios de salud.

  Los REM no solo son registros administrativos, son herramientas estratégicas que permiten a los gestores y decisores en el sistema de salud hacer seguimiento y tomar decisiones informadas para mejorar la salud de la población. Facilitan una gestión basada en datos, lo que es crucial para la administración efectiva del sistema de salud en cualquier país.

2.3. Archivos REM Serie A.

La Serie A se enfoca en la documentación, análisis de controles y atenciones sanitarias realizados en los distintos establecimientos de salud en las diferentes etapas de la vida de los pacientes. Detallando el contenido que se puede encontrar en esta serie se encuentran (1),

1. **Controles de salud sexual y reproductiva:**
   - Monitoreo de la salud reproductiva: Incluye datos sobre consultas prenatales, controles ginecológicos, y otros servicios relacionados con la salud reproductiva.
   - Prevención y atención: Registra intervenciones como la aplicación

de vacunas y exámenes específicos destinados a la prevención de enfermedades transmitidas sexualmente y otros problemas de salud reproductiva.

2. **Controles según ciclo vital:**
   - Infancia: Incluye registros de vacunación, controles de desarrollo infantil, y seguimiento nutricional.
   - Adolescencia: Atenciones relacionadas con la educación sexual, salud mental, y detección temprana de problemas de salud comunes en esta etapa.
   - Adultez y vejez: Registros de exámenes preventivos como controles de presión arterial, diabetes, y evaluaciones de salud cardiovascular, entre otros.
3. **Evaluaciones Específicas de Condiciones de Salud o Programas:**
   - Salud cardiovascular: Incluye datos sobre consultas y tratamientos para enfermedades del corazón, evaluaciones de riesgo cardiovascular, y seguimiento de pacientes con condiciones crónicas.
   - Diabetes: Registra controles de glucemia, educación para el manejo de la diabetes, y seguimiento de complicaciones relacionadas.
   - Otros programas específicos: Incluye programas de manejo de asma, obesidad, y cáncer, entre otros, con datos sobre detecciones, intervenciones y seguimientos.
4. **Atenciones de Urgencia y Consultas Especializadas:**
   - Urgencias: Datos sobre la frecuencia y tipo de emergencias atendidas, incluyendo accidentes, urgencias obstétricas, y otras condiciones agudas que requieren atención inmediata.
   - Consultas especializadas: Incluye referencias a especialistas, tratamientos específicos y seguimientos en áreas como oncología, cardiología, endocrinología, entre otras.

La Serie A proporciona una visión del estado de salud y las necesidades de atención de la población en diferentes etapas de la vida, permitiendo a los gestores de salud

pública y a los establecimientos ajustar y mejorar la oferta de servicios según las tendencias y resultados observados en los datos recopilados.

Esta serie es una herramienta para el seguimiento continuo y la evaluación de la efectividad de los programas de salud implementados, contribuyendo así a una mejor planificación y gestión de recursos en el sistema de salud chileno (2).

2.4. Datos y diccionarios Serie A.

Los archivos REM vienen estructurados en 57 columnas en los cuales se encuentran:

- **Datos:** Es un consolidado con todos los valores numéricos de los procedimientos realizados en los establecimientos de salud en Chile presentados en la tabla I.

Tabla I (5)
Campos del Archivo REM.

Mes	IdServicio	Año	IdEstablecimiento	CodigoPrestacion	IdRegion	IdComuna	Col01	...	Col50

Este tipo de archivos a través de los años han sido emitidos en formato TXT, como también, CSV o valores separados por coma.

Tabla II
Descripción de las variables del REM.

(5)

Variable	**Descripción**
Mes	Indica el mes cuando se realizó el procedimiento.
Código de prestación	Contenido en el diccionario respectivo a su serie, indica el procedimiento realizado.
Id de región	Indica la región en la que se realizó el procedimiento.
Id de comuna	Indica la comuna en la región respectiva que se realizó el procedimiento, está indicada en el listado de establecimientos emitido por el DEIS.
Col01-50	Contenido en el diccionario respectivo de cada serie, indica el valor cuantitativo del procedimiento realizado.
Id de establecimiento	Contenido en el listado de establecimientos emitido por el DEIS, indica tipo de establecimiento, dependencia, nombre oficial del establecimiento, vía y número de calle, teléfono y coordenadas del establecimiento.

La interpretación manual de la columna 'Mes' e 'IdRegion' se puede realizar como se muestran en las tablas III y IV respectivamente.

Tabla III
Meses del año.

Mes	Descripción
1	Enero
2	Febrero
3	Marzo
4	Abril
5	Mayo
6	Junio
7	Julio
8	Agosto
9	Septiembre
10	Octubre
11	Noviembre
12	Diciembre

Tabla IV
Regiones de Chile.

Id de región	Descripción
1	Tarapacá
2	Antofagasta
3	Atacama
4	Coquimbo
5	Valparaíso
6	Libertador General Bernardo O'Higgins
7	Maule
8	Bío-Bío
9	Araucanía
10	Los Lagos
11	Aysén
12	Región de Magallanes y de la Antártica Chilena
13	Región Metropolitana
14	Los Ríos
15	Arica y Parinacota
16	Ñuble

2.5. Diccionarios.

Contenidos dentro de la carpeta con el mismo nombre, es decir, diccionario, en ellos se encuentran los detalles de los procedimientos realizados, como el tipo de examen (entre más), sección, códigos de prestación y descripción de la columna con los valores *ColXX*. Los cuales se detallan en la tabla V

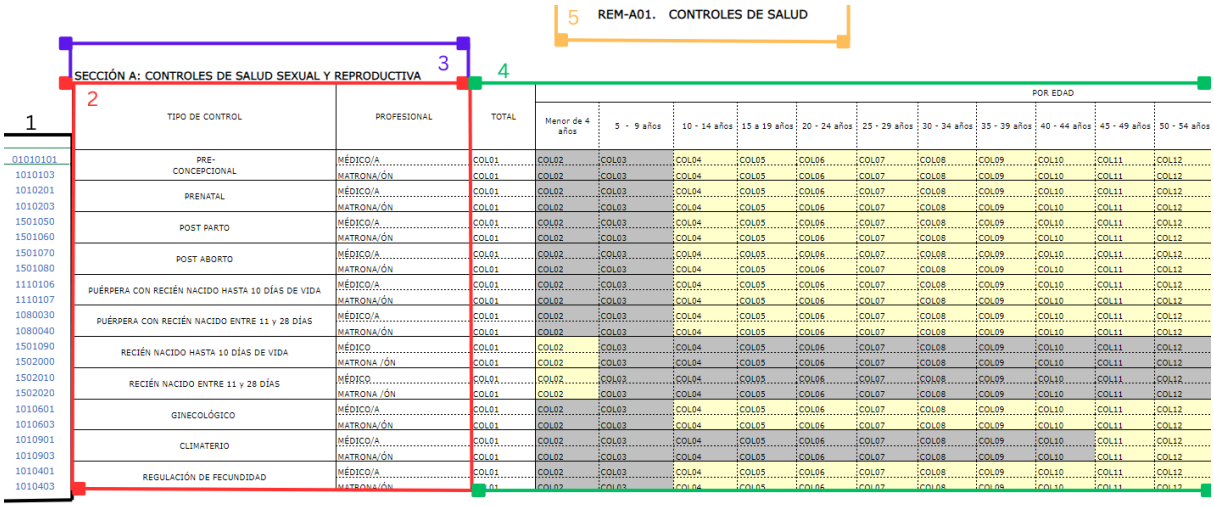


Figura 2.5.1
Formato de los diccionarios Serie A (5).

Tabla V
Descripción de la tabla de controles de salud

(5).

Punto	Descripción
1: Código de prestación	Se ubican los códigos de prestación del procedimiento que se realizó. En esta sección, se encuentran los códigos específicos asociados a cada procedimiento realizado. Estos códigos están en la primera columna a la izquierda de la tabla, y ejemplos de ellos son 01010101, 1010103, 1010201, entre otros. Cada código representa un tipo específico de prestación o servicio dentro del control de salud.
2: Identificador	Son los detalles del procedimiento que se realizó, ya sea, qué tipo de control, examen y por quién fue ejercido. Esta parte de la tabla proporciona información sobre los procedimientos realizados. Bajo la columna 'TIPO DE CONTROL', se especifica el tipo de control o examen realizado, como 'PRE-CONCEPCIONAL', 'PRENATAL', 'POST PARTO', etc. La columna 'PROFESIONAL' indica quién realizó el procedimiento, como 'MÉDICO/A' o 'MATRONA/ÓN'.
3: Sección	Es el encabezado de la tabla, especificando la sección. Este encabezado indica a qué sección pertenece la tabla y proporciona un contexto general de los datos que se presentan. En la Fig.2.5.1, el encabezado dice 'SECCIÓN A: CONTROLES DE SALUD SEXUAL Y REPRODUCTIVA', lo que significa que los datos detallados en la tabla se relacionan con los controles y exámenes de salud sexual y reproductiva.
4: Descriptores	Son las descripciones de cada valor 'Col', en ellos se pueden encontrar total, grupos etareos, total por sexo, entre más dependiendo de cada tabla. Las columnas bajo el encabezado 'POR EDAD' están etiquetadas como 'COL01', 'COL02', etc., y cada una representa un grupo etario específico o una categoría demográfica. Por ejemplo, 'Menor de 4 años', '5 - 9 años', '10 - 14 años', etc. Estos valores 'Col' permiten desglosar los datos totales en segmentos específicos, mostrando cómo se distribuyen los procedimientos a través de diferentes edades y sexos, según la tabla en cuestión.
5: Contexto	Se ubica en el encabezado de cada hoja en donde se detalla el contexto de las tablas que se mostrarán en la hoja. Este encabezado se encuentra en la parte superior derecha de la Fig. 2.5.1 y dice 'REM-A01. CONTROLES DE SALUD'. Este encabezado proporciona el contexto general de las tablas presentadas en la hoja, indicando que los datos relacionados se refieren a controles de salud específicos catalogados bajo 'CONTROLES DE SALUD'. Esto ayuda a los usuarios a identificar el propósito y el contenido de las tablas en la hoja.

2.6. Modelos de Lenguaje a Gran Escala (LLM).

Son sistemas de inteligencia artificial diseñados para comprender y generar texto de manera coherente y relevante. Estos modelos se entrenan con grandes volúmenes de texto para aprender patrones de lenguaje, estructura gramatical, y contextos diversos. La capacidad de estos modelos para generar texto, responder preguntas, y realizar tareas de procesamiento de lenguaje natural los hace extremadamente útiles en múltiples aplicaciones (6).

- **GPT (Generative Pre-trained Transformer):** GPT es un tipo de LLM desarrollado por OpenAI que utiliza la arquitectura de transformer para predecir la siguiente palabra en una secuencia de texto, basándose en las palabras anteriores. A través de su entrenamiento en un diverso conjunto de textos de internet, GPT es capaz de realizar tareas como traducción de idiomas, resumen de textos, generación de contenido, y más, mostrando una comprensión del lenguaje humano (7).

2.7. Streamlit.

Streamlit es una plataforma de código abierto en Python que permite la creación de aplicaciones web interactivas para la visualización y análisis de datos de manera sencilla y eficiente. Streamlit facilita a los desarrolladores la construcción de interfaces de usuario amigables sin necesidad de conocimientos profundos de desarrollo web, gracias a su diseño intuitivo que se basa en componentes predefinidos y la integración directa con pandas, y otras librerías de Python para el análisis de datos. Esto lo convierte en una herramienta ideal para proyectos de análisis de datos y machine learning, permitiendo a los usuarios visualizar resultados en tiempo real, modificar parámetros y explorar diferentes escenarios de análisis (8).

En el contexto del desarrollo de una interfaz para la consulta y visualización de datos de salud pública, Streamlit resulta particularmente útil debido a su capacidad para manejar datos y ofrecer una interfaz interactiva que facilita la exploración y análisis de dichos datos (9).

2.8. Métodos usados.

2.8.1. Método de chunking.

El método de chunking es una técnica utilizada para manejar y procesar grandes volúmenes de datos sin sobrecargar la memoria del sistema. Consiste en dividir el conjunto de datos en porciones más pequeñas, o 'chunks', que pueden ser procesadas individualmente. Esto permite que el sistema lea y manipule datos en segmentos manejables, en lugar de intentar cargar todo el conjunto de datos en la memoria a la vez. Esta técnica es particularmente útil cuando se trabaja con archivos de gran tamaño que exceden la capacidad de memoria disponible (10).

2.8.2. Método de cruce.

El método de cruces es una técnica empleada para combinar y verificar datos provenientes de diferentes fuentes o tablas, con el fin de asegurar la coherencia de la información. Este método consiste en la comparación y asociación de campos comunes entre distintos conjuntos de datos, lo cual permite identificar y corregir discrepancias, además de enriquecer la información disponible mediante la integración de datos complementarios (11). En la práctica, el cruce de datos se lleva a cabo mediante operaciones como 'join' en bases de datos o mediante scripts personalizados en lenguajes de programación como Python (12).

2.9. Formato JSON.

JSON (JavaScript Object Notation) es un formato de texto ligero para el intercambio de datos. Su estructura es fácil de leer y escribir. Utiliza un formato basado en texto que es completamente independiente del lenguaje de programación, pero utiliza convenciones familiares para los programadores de lenguajes de la familia C, incluyendo C, C++, C#, Java, JavaScript, Python y muchos otros (13).

Un archivo JSON contiene:

- **Objetos**: Son conjuntos de pares nombre/valor. Se representan entre llaves {}.

- **Arrays**: Son listas ordenadas de valores. Se representan entre corchetes.
- **Valores**: Pueden ser una cadena, un número, un objeto, un array, un valor booleano o `null`.

Ejemplo:

```
{
  "nombre": "Max",
  "edad": 24,
  "ciudad": "Concepción",
  "hobbies": ["Dibujar", "viajar", "cantar"]
}
```

Los archivos JSON son útiles en este tipo de proyectos por varias razones:

- **Intercambio de Datos**: JSON es ideal para el intercambio de datos entre un servidor y una aplicación web, como en aplicaciones de visualización de datos.
- **Compatibilidad**: Es compatible con casi todos los lenguajes de programación y es fácil de integrar.
- **Eficiencia**: Permite el manejo eficiente de grandes volúmenes de datos, lo cual es crucial para proyectos que involucran datos masivos, como los REM.
- **Flexibilidad**: La estructura de JSON facilita la adición de nuevos campos y la modificación de la estructura de datos sin afectar la funcionalidad existente.

2.10. Consultas SQL (Structured Query Language)

Las consultas SQL son instrucciones utilizadas para interactuar con bases de datos relacionales. Estas permiten realizar operaciones como la recuperación, inserción, actualización y eliminación de datos, así como la creación y modificación de la estructura de las bases de datos (14).

- **Manejo de datos:** Las consultas SQL permiten recuperar información específica de grandes bases de datos sin necesidad de revisar manualmente todos los registros. Esto es especialmente útil en proyectos que implican

la gestión de datos de salud, donde se requiere acceder rápidamente a información relevante (15).

- **Automatización y Estandarización:** La automatización en la generación de consultas SQL permite estandarizar el proceso de obtención de datos.
- **Flexibilidad y Personalización:** SQL permite crear consultas personalizadas que se ajustan a los requisitos específicos del proyecto. Por ejemplo, una interfaz que facilita la generación de consultas SQL basadas en parámetros seleccionados por el usuario.
- **Visualización y Análisis de Datos:** A través de SQL, es posible extraer conjuntos de datos específicos que luego pueden ser visualizados.

2.11. Discusión.

El desarrollo de una interfaz para la consulta y visualización de los archivos REM es beneficioso en la gestión de datos de salud en Chile. En esta sección se plantearon los datos contenidos en la Serie A de los REM, que son utilizados para la monitorización y planificación de los servicios de salud. Estos datos son esenciales para entender y mejorar la atención sanitaria brindada en diversas etapas de la vida y en distintos establecimientos de salud. No se incluyó el período 2024, debido a que, el MINSAL emite estos reportes cada trimestre del año correspondiente.

La adopción de tecnologías como los modelos de lenguaje a gran escala, como Chat GPT 4-O (7), pueden utilizarse para generar información para las consultas, agilizando el proceso de decodificación de los datos. Esto es relevante en contextos donde métodos tradicionales, como la implementación en Python, no han resultado efectivos debido a la complejidad de los datos en múltiples formatos de tablas, dificultando la extracción coherente de la información.

Herramientas como Streamlit (9) permiten la creación de interfaces de usuario intuitivas para la visualización y análisis de datos. La combinación de estas tecnologías puede superar las limitaciones actuales que se detallan en el Capítulo 3 en el manejo de los datos de salud, permitiendo una mejor integración y análisis de la información contenida en los REM. Esto no solo ayudaría en la accesibilidad de los datos de salud pública al estar dispuestos de manera que su decodificación requiere de muchos parámetros a considerar, sino que también podría ser utilizado

tanto para la gestión de salud pública como particular, como es el caso de este proyecto realizado para la empresa privada HealthTracker (4) los cuales trabajan en el rubro recopilando datos de enfermedades crónicas, entre otras.

Capítulo 3

Búsqueda manual para decodificación de archivos REM.

3.1. Introducción.

El proceso de decodificación de los archivos REM presenta varias limitaciones y dificultades, especialmente si se desean considerar múltiples parámetros de interés, como la búsqueda por establecimiento, región, código y/o códigos de prestación. Esto convierte la búsqueda en un proceso engorroso para el usuario. En el presente capítulo se aborda la búsqueda manual de información utilizando los recursos disponibles del MINSAL para su decodificación. Además, se presenta un diagrama de flujo 3.1.1, que ilustra las tareas involucradas en este proceso.

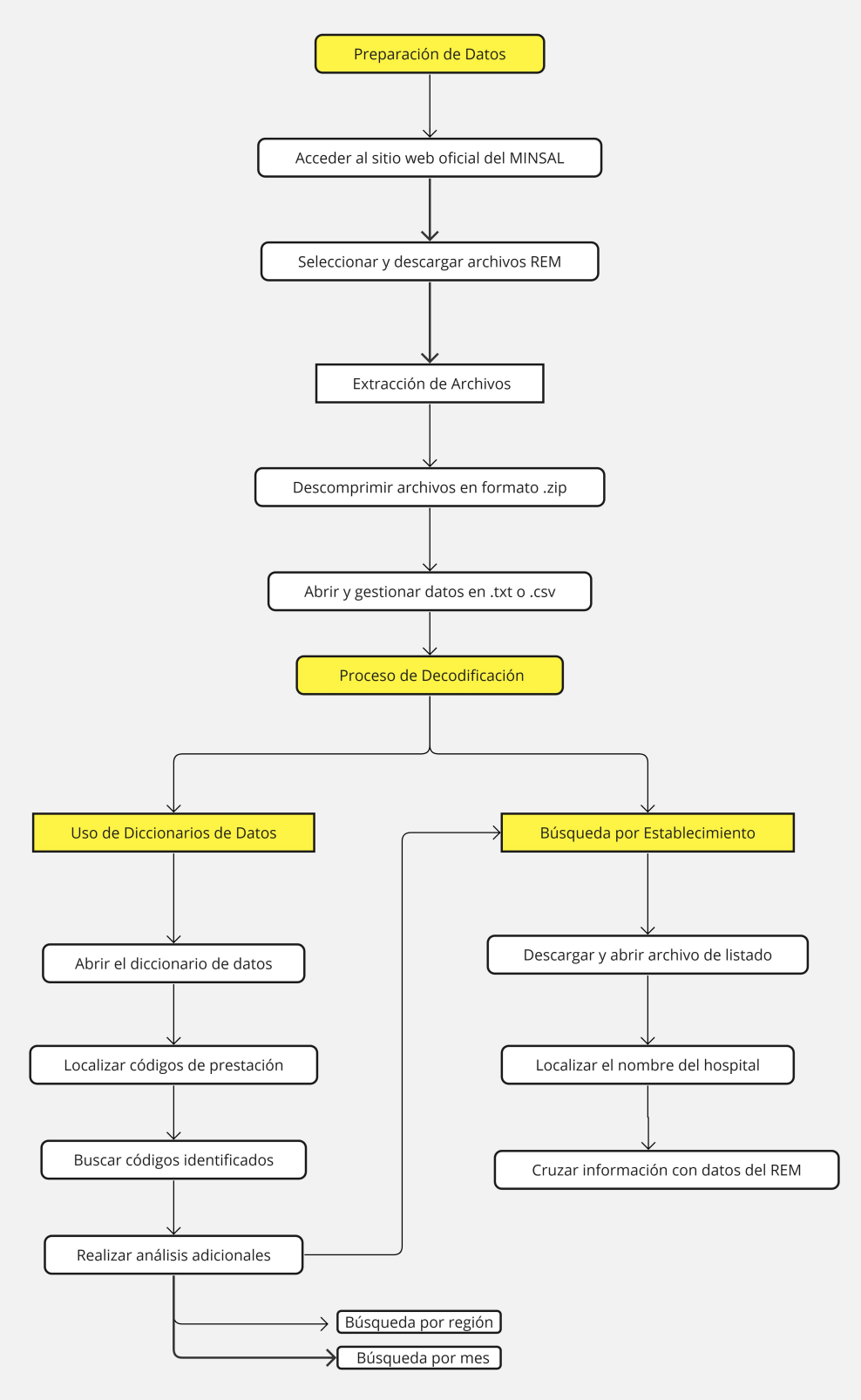


Figura 3.1.1
Diagrama de flujo para búsqueda manual.

3.2. Preparación de Datos.

- **Descarga de Archivos REM:** Los archivos REM son descargados directamente desde el sitio web oficial del MINSAL (16). Estos archivos están organizados y etiquetados según el año de reporte. El proceso de descarga implica seleccionar los archivos correspondientes al período de estudio deseado.
- **Extracción de Archivos:** Una vez descargados, los archivos REM se encuentran comprimidos en formato ZIP. Al descomprimir estos archivos, los datos están en formatos TXT o CSV, dependiendo del año. Estos formatos no requieren un preprocesamiento; pueden ser manejados directamente mediante aplicaciones comunes como bloc de notas o Microsoft Excel.

3.3. Proceso de Decodificación.

El uso de diccionarios de datos es necesario para decodificar los códigos encontrados en los archivos REM, convirtiendo los códigos encriptados en descripciones con información para el análisis. El proceso inicia con la apertura del diccionario de datos, que está organizado por hojas específicas, cada una categorizando diferentes tipos de información. En estas hojas, los códigos de prestación se encuentran en la primera columna la cual no está etiquetada, acompañados por detalles adicionales en columnas adyacentes que describen los procedimientos y valores asociados a esos códigos, descritos como valores COL, que corresponden a los datos cuantitativos en los archivos del REM.

Una vez que se identifica un código de prestación en el diccionario, se procede a buscarlo en los datos consolidados usando herramientas de búsqueda. No obstante, identificar el código es solo el primer paso, ya que a menudo se requiere realizar otro tipos de análisis que implican buscar datos adicionales como por establecimiento, mes o región para obtener una comprensión general de los datos de salud. Adicionalmente, ciertos identificadores, como 'IdEstablecimiento', requieren la descarga de información complementaria.

3.3.1. Búsqueda por establecimiento.

Si se requiere buscar por establecimiento esto implica utilizar el listado de establecimientos proporcionada por el MINSAL. Una vez descargado este archivo, se localiza el nombre del hospital en la columna 'Nombre Oficial'. Importante es notar que los códigos de establecimiento pueden variar año con año, por lo que el archivo incluye tanto código antiguo establecimiento como código nuevo establecimiento. Tras identificar el código, este se busca en los archivos consolidados del REM para obtener los datos del establecimiento.

Sin embargo, este método presenta problemas, especialmente cuando se necesita analizar datos en un contexto temporal o geográfico más amplio. En tales casos, es necesario cruzar la información del establecimiento con datos extraídos del diccionario de datos del REM, así como con otros indicadores temporales como el mes y el año. Este cruce de datos complica el proceso de análisis y búsqueda, ya que requiere manejar múltiples variables y asegurar la coherencia entre los diferentes formatos y códigos a lo largo de los años, dificultando la consulta directa y eficiente de información por periodos.

La disposición de información contenida en el listado de establecimientos se presenta como se muestra en la tabla VI.

Tabla VI (17)
Formato del listado de establecimientos.

Columnas	**Valores**
Código Antiguo Establecimiento	01-010
Código nuevo Establecimiento	101010
Código Establecimiento Madre	
Código nuevo Establecimiento Madre	
Código Región	15
Región	De Arica Parinacota
SEREMI / Servicio de Salud	Servicio de Salud Arica
Perteneciente	Perteneciente
Tipo Establecimiento	Dirección Servicio de Salud
Tipo Estrategia	
Certificación	
Dependencia	Servicio de Salud
Nivel de Atención	No Aplica
Nombre Oficial	Actividades Gestionadas por la Dirección del Servicio para apoyo de la Red (S.S de Arica)
Alias	
Código Comuna	15101
Nombre Comuna	Arica
Vía	Calle

Continúa en la siguiente página

Tabla VI
Formato del listado de establecimientos (continuación).

Columnas	**Valores**
Número	305
Dirección	Arturo Prat
Teléfono	
Fecha Vigencia (Desde)	01/01/2009
Fecha Cierre (Hasta)	
Fecha Reapertura	
Código Servicio de Salud/ Seremi	1
Clasificación SAPU	
Fecha Cambio	
Observación	
LONGITUD [Grados decimales]	-70.320689
LATITUD [Grados decimales]	-18.477644

3.4. Otras opciones de búsqueda.

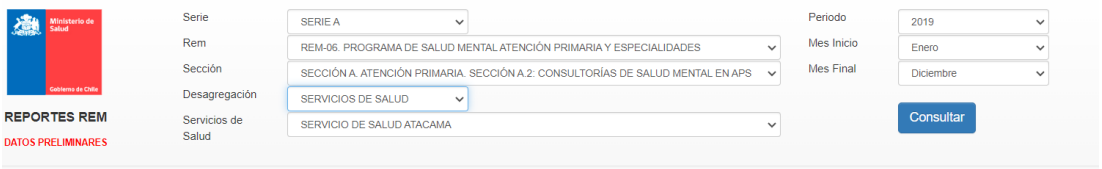
3.4.1. Consulta de reportes REM en página del MINSAL.

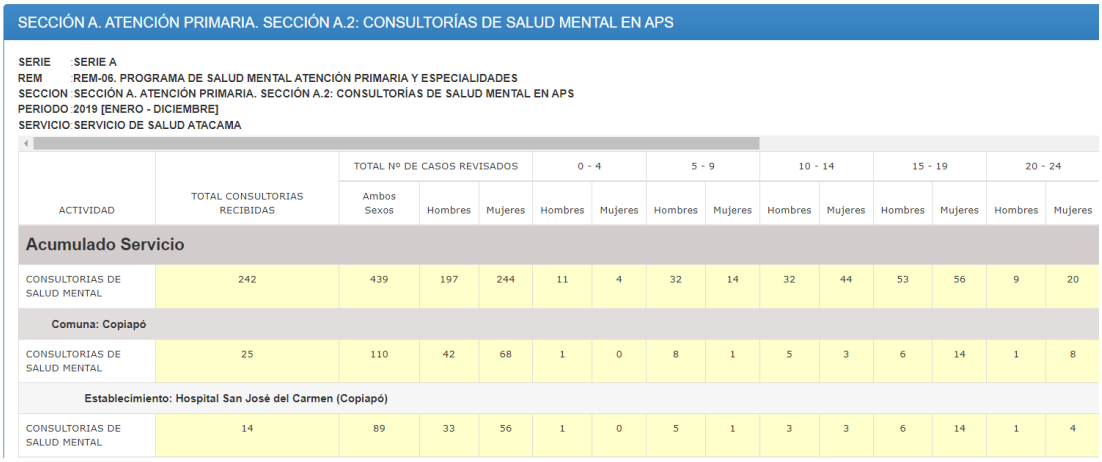
Una alternativa para la obtención de datos es utilizar directamente el sitio web del MINSAL que se muestran en las Fig. 3.4.1 y 3.4.2, el cual permite acceder a información ya decodificada. Sin embargo, esta opción presenta limitaciones. Primero, el sistema extrae automáticamente toda la información disponible en la tabla, sin ofrecer la opción de enfocarse en un dato específico, es decir, no se puede buscar específicamente por solo una región, código de prestación, establecimiento, descriptor ni meses. Esto resulta en una sobrecarga de información y dificulta el

análisis focalizado.

Por otra parte, el rango de datos disponibles se limita a los años 2017-2020, lo que puede resultar en una falta de actualización. La interfaz sólo permite seleccionar un año a la vez, lo cual restringe las comparativas que podrían ser necesarias para estudiar tendencias o evoluciones a lo largo del tiempo.

Otra restricción es que, al seleccionar datos por meses o por todas las regiones, el sistema solo proporciona datos acumulados, ya sea para el período seleccionado o para todas las regiones en conjunto. Esto complica un análisis comparativo, especialmente si el objetivo es evaluar la evolución de un establecimiento a lo largo de los años.





SECCIÓN A. ATENCIÓN PRIMARIA. SECCIÓN A.2: CONSULTORÍAS DE SALUD MENTAL EN APS

SERIE :SERIE A
REM :REM-06. PROGRAMA DE SALUD MENTAL ATENCIÓN PRIMARIA Y ESPECIALIDADES
SECCION :SECCIÓN A. ATENCIÓN PRIMARIA. SECCIÓN A.2: CONSULTORÍAS DE SALUD MENTAL EN APS
PERIODO :2019 [ENERO - DICIEMBRE]
SERVICIO :SERVICIO DE SALUD ATACAMA

ACTIVIDAD	TOTAL CONSULTORIAS RECIBIDAS	TOTAL Nº DE CASOS REVISADOS			0 - 4		5 - 9		10 - 14		15 - 19		20 - 24	
		Ambos Sexos	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres
Acumulado Servicio														
CONSULTORIAS DE SALUD MENTAL	242	439	197	244	11	4	32	14	32	44	53	56	9	20
Comuna: Copiapó														
CONSULTORIAS DE SALUD MENTAL	25	110	42	68	1	0	8	1	5	3	6	14	1	8
Establecimiento: Hospital San José del Carmen (Copiapó)														
CONSULTORIAS DE SALUD MENTAL	14	89	33	56	1	0	5	1	3	3	6	14	1	4

Figura 3.4.1
Página oficial del MINSAL para la realización de consultas al consolidado Serie A filtrado por establecimiento y región (3).

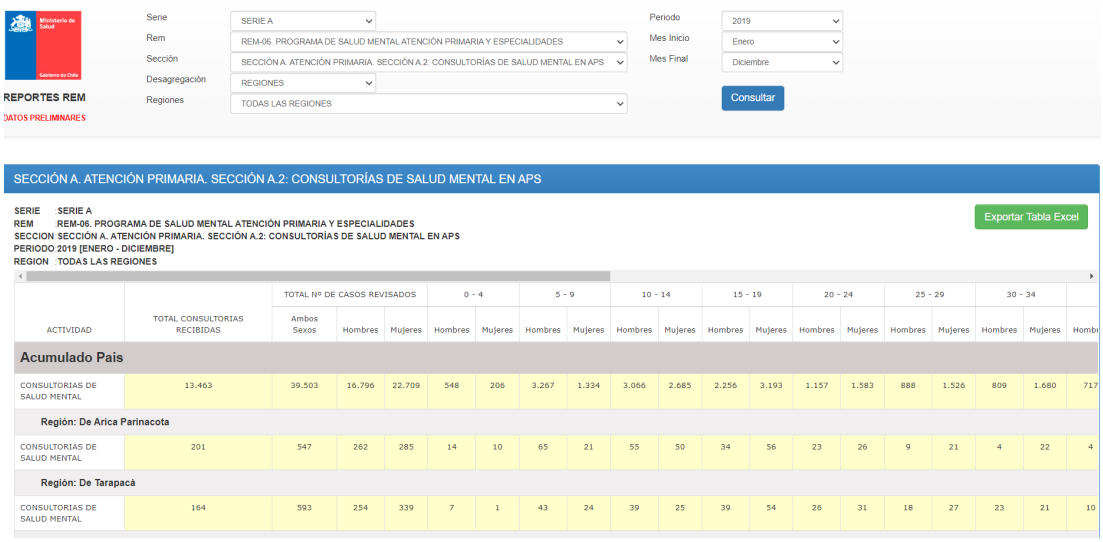


Figura 3.4.2

Página oficial del MINSAL para la realización de consultas al consolidado Serie A filtrado por todas las regiones (3).

3.5. Discusión.

El proceso de decodificación de los archivos REM hasta la fecha sigue sin tener solución. Actualmente, no existe una herramienta integral que permita realizar búsquedas considerando todos los parámetros deseados por el usuario, lo que representa una limitación para quienes desean consultar la evolución de ciertos procedimientos en un período específico, ya sea por años o meses, y en establecimientos de interés particular. Las herramientas actuales, como Excel, presenta restricciones debido a las limitaciones en el manejo de grandes volúmenes de datos y la dependencia de las habilidades del usuario para manejar estos datos proporcionados por el MINSAL. La problemática es la ausencia de una herramienta que integre todos los parámetros contenidos en la base de datos de los REM para una extracción y consulta. Herramientas como Excel presentan restricciones en términos de capacidad de procesamiento y funcionalidades de búsqueda, impidiendo una decodificación rápida de la información. Además, la eficacia de la decodificación y análisis de datos depende en gran medida de las habilidades del usuario para manipular las herramientas disponibles y los datos proporcionados, lo que puede conducir a errores humanos y a una interpretación inexacta de los datos, afectando la calidad y utilidad de la información extraída. Por lo tanto, se ha propuesto el uso de tecnologías avanzadas como Streamlit que faciliten la interacción del usuario con los datos y mejoren la accesibilidad de la información.

Capítulo 4

Desarrollo.

4.1. Introducción.

A continuación, se presentan las implementaciones realizadas para la búsqueda en el archivo REM Serie A describiendo los pasos que fueron necesarios para la consolidación de la base de datos, extracción y generación del formato decodificador, y prototipo de consulta al consolidado generado.

4.2. Consolidación de archivos REM Serie A.

El proceso de consolidación de los archivos REM de 2009 a 2023 se realizó mediante un script en Python, en donde se usó el método de chunking para manejar volúmenes de datos sin sobrecargar la memoria del sistema. Este método facilitó el procesar y cargar los datos de manera segmentada, facilitando la lectura y la concatenación de los archivos de datos directamente desde su formato descomprimido. Durante este proceso, se estandarizaron los nombres de las columnas, como *Mes*, *IdServicio*, *Año*, y otros, adaptándolos a un formato uniforme para asegurar la consistencia entre los diferentes años, dado que algunos presentaban variaciones en el formato de columnas como *COL1* frente a *Col01* difiriendo en la escritura de estas.

El script ejecutó la concatenación de estos archivos ajustando los nombres de columnas y exportando los resultados en segmentos al directorio designado para evitar la saturación del entorno.

4.3. Generación JSONs.

El empleo del formato JSON para estructurar los datos extraídos de los diccionarios REM es fundamental debido a la diversidad y complejidad de las tablas contenidas en estos archivos, que a menudo varían en su formato de interpretación. Al convertir estos datos a un formato JSON, se logra un fácil acceso a estos datos que es necesaria para la manipulación y el análisis posterior. Este formato permite que la información pueda ser accesible y extraíble, independientemente de la estructura de cada tabla original.

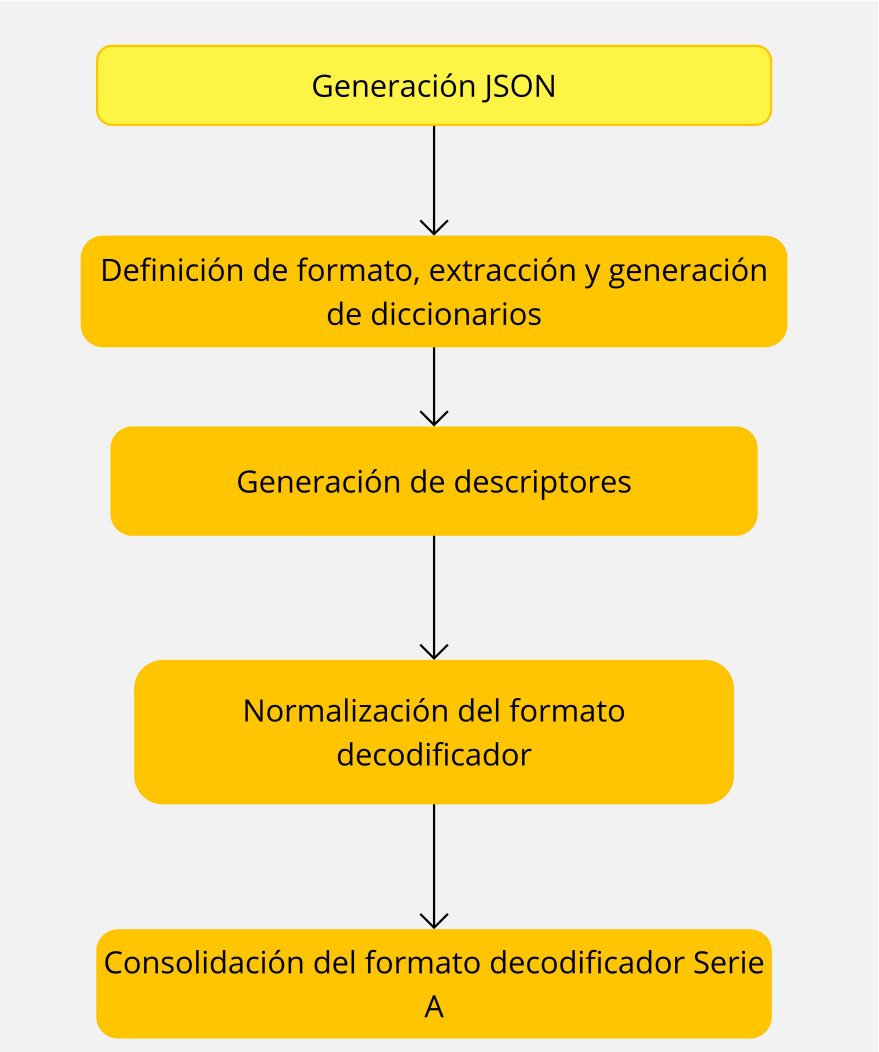


Figura 4.3.1
Diagrama flujo para generación del formato decodificador.

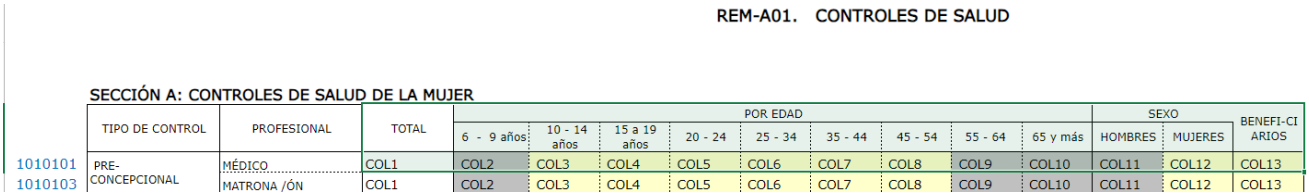
4.3.1. Definición de formato, extracción y generación de diccionarios.

El proceso de generación de JSONs, ejecutado a través de un script de Python, transforma datos estructurados del diccionario REM en entradas de JSON definidas que incluyen campos como *código*, *serie*, *hoja*, *contexto*, *año*, *sección* e *identificador*. Cada campo refleja un aspecto específico de los datos del diccionario, estructurando la información de manera que mantiene la integridad y el contexto necesario para su utilización. Por ejemplo, el script establece valores predeterminados en campos como *serie*, *año*, y *hoja*, que pueden modificarse según las necesidades, mientras que, los datos como *sección* e *identificador* se extraen directamente desde archivo Excel original de los diccionarios. Este método estandarizado no solo reduce el riesgo de errores humanos a la entrada manual de datos, sino que también acelera el proceso de manejo de datos al simplificar y automatizar la transformación y extracción de la información necesaria.

A través del script antes mencionado, se pueden extraer todas las tablas de cada hoja directamente del diccionario el cual se puede apreciar en la Fig. 4.3.2 en formato JSON que poseen la información de interés siendo estas el *código* que sería en este caso las columnas que contienen los valores como *1010101*, para *sección* el encabezado de cada tabla, *contexto* sería la descripcion de la fila *REM-A01.* e *identificador* son las descripciones que no involucran a ninguna columna con los valores *ColXX*, y de esta manera se puede consolidar un diccionario JSON de todo un año para cada hoja como se ve en la Fig. 4.3.2, destacando que el proceso es automatizado y dejando el campo de descriptores vacío.


```
{
    "01010101": {
        "codigo": "01010101",
        "serie": "A",
        "hoja": "A01",
        "contexto": "CONTROLES DE SALUD",
        "año": 2009,
        "seccion": "SECCIÓN A: CONTROLES DE SALUD DE LA MUJER",
        "identificador": "PRE-\nCONCEPCIONAL, MÉDICO",
        "descriptores": {}
    },
```


Figura 4.3.2
Ejemplo de JSON definido y generado para cada hoja del diccionario.



REM-A01. CONTROLES DE SALUD

SECCIÓN A: CONTROLES DE SALUD DE LA MUJER

	TIPO DE CONTROL	PROFESIONAL	TOTAL	POR EDAD									SEXO		BENEFI-CIARIOS
				6 - 9 años	10 - 14 años	15 a 19 años	20 - 24	25 - 34	35 - 44	45 - 54	55 - 64	65 y más	HOMBRES	MUJERES	
1010101	PRE-CONCEPCIONAL	MÉDICO	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11	COL12	COL13
1010103		MATRONA /ÓN	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11	COL12	COL13

Figura 4.3.3
Referencia de tabla del diccionario REM Serie A de donde se extrae los campos del formato JSON
(5).

4.3.2. Generación de descriptores.

Para la generación de este campo se siguió el siguiente diagrama 4.3.4, donde se realizó la implementación de un script en Python que exporta la información desde los diccionarios, los campos de secciones etiquetando cada hoja de donde fue extraída aquella sección y dejando una columna vacía para descriptores, dando como resultado una plantilla Excel con el formato presentado en el cuadro VII.

Tal que, mediante vía LLM se utilizan las instrucciones descritas en el cuadro VIII y junto al uso de Chat-GPT 4-o se fueron generando los formatos JSON de las entradas de cada tabla, por ejemplo, como se aprecia en la Fig. 4.3.3 se selecciaron los campos de interés que serían las descripciones de columnas referentes a los valores *ColXX* para la generación de descriptores dando como producto de la consulta lo que se ve generado en la Fig. 4.3.5.

Para completar los valores de la columna descriptores de la plantilla Excel se utilizó la información generada por el LLM mencionado en donde se aseguró su correcta asignación a cada fila esto de manera manual y mediante un script en Python se realizó un método de cruce con la plantilla Excel y los archivos JSON generados para cada hoja de cada año, relacionando los campos *hoja*, *seccion* y *descriptores* para su correcta asignación de manera automatizada y dando como resultado un listado de archivos JSON con el campo de descriptores correspondiente mostrado en la Fig. 4.3.6.

Tabla VII
Formato plantilla Excel con columnas hoja, sección y descriptores.

hoja	**sección**	**descriptores**
A01	SECCIÓN A: CONTROLES DE SALUD SEXUAL Y REPRODUCTIVA	
	...	

Tabla VIII
Estructura del prompt utilizado para consulta a LLM.

Estructura del prompt	**Prompt ajustado**
¿Podrías generar en formato JSON las siguientes entradas <Entrada 1>, <Entrada 2>, ..., <Entrada XX>?	¿Podrías generar en formato JSON las siguientes entradas "TOTAL POR EDAD SEXO BENEFICIARIOS 6 - 9 años 10 - 14 años 15 - 19 años 20 - 24 25 - 34 35 - 44 45 - 54 55 - 64 65 y más HOMBRES MUJERES COL1 COL2 COL3 COL4 COL5 COL6 COL7 COL8 COL9 COL10 COL11 COL12 COL13"

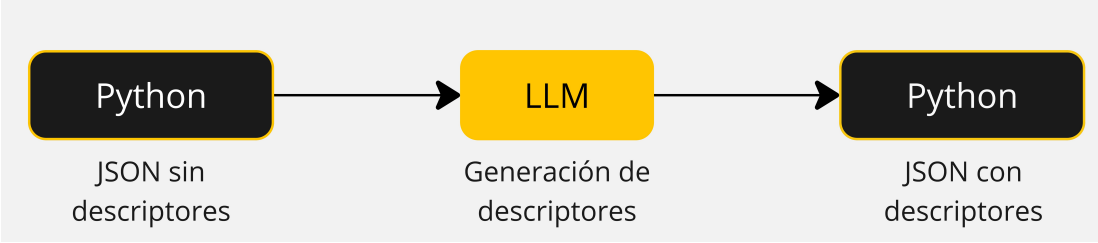


Figura 4.3.4
Diagrama de flujo para generación de descriptores.

Entrada 1:

json Copy code

```
{
  "TOTAL": "COL1",
  "POR_EDAD": {
    "6_9_anos": "COL2",
    "10_14_anos": "COL3",
    "15_19_anos": "COL4",
    "20_24": "COL5",
    "25_34": "COL6",
    "35_44": "COL7",
    "45_54": "COL8",
    "55_64": "COL9",
    "65_y_mas": "COL10"
  },
  "SEXO": {
    "Hombres": "COL11",
    "Mujeres": "COL12"
  },
  "BENEFICIARIOS": "COL13"
}
```

Figura 4.3.5
Resultado consulta vía LLM para descriptor sección A del diccionario Serie A.

```
"01010101": {
    "codigo": "01010101",
    "serie": "A",
    "hoja": "A01",
    "contexto": "CONTROLES DE SALUD",
    "año": 2009,
    "seccion": "SECCIÓN A: CONTROLES DE SALUD DE LA MUJER",
    "identificador": "PRE-\nCONCEPCIONAL, MÉDICO",
    "descriptores": {
        "Total": "COL1",
        "desglose por edad": {
            "6 - 9 años": "COL2",
            "10 - 14 años": "COL3",
            "15 a 19 años": "COL4",
            "20 - 24 años": "COL5",
            "25 - 34 años": "COL6",
            "35 - 44 años": "COL7",
            "45 - 54 años": "COL8",
            "55 - 64 años": "COL9",
            "65 y más": "COL10"
        },
        "sexo": {
            "Hombres": "COL11",
            "Mujeres": "COL12"
        },
        "BENEFICIARIOS": "COL13"
    }
},
```


Figura 4.3.6
Resultado del cruce del archivo JSON con la plantilla Excel para descriptores.

4.3.3. Normalización del formato decodificador.

Para la normalización del formato decodificador se consideraron los siguientes aspectos, normalizar las secciones, contexto, identificadores y descriptores, esto por la razón de unificar información que puede significar lo mismo, por dar ejemplo, se puede tener escrito **0-4** o **0A4 años** que en ambos casos significaría **0 - 4 años** pero que por razones de fuente original y/o generación por parte del LLM están escritas de manera distinta, tal que, detallando algunas:

- **Normalización de descriptores** Usando una plantilla Excel generada mediante un script de Python, se realizó una extracción del campo de descriptores del consolidado JSON de Serie A de todos sus años. La extracción se separó en plantillas diferentes según el nivel de desglose definido por la anidación dentro de descriptores, es decir, `descriptores{{}}`, donde la primera llave indica un descriptor sin desglose, y por consecuencia, la siguiente llave es el primer nivel de desglose, y así sucesivamente. Una vez se extrae el campo de descriptores asegurando su extracción en valores únicos, se usó un script en Python que permitió normalizar la mayoría de los valores y asignarlos en la columna *Descriptores normalizados*, aunque algunos tuvieron que ser revisados y modificados manualmente por discrepancias de formato.

Tabla IX
Formato de extracción y asignación de descriptores normalizados en plantilla Excel.

Descriptores	**Descriptores normalizados**
0 - 4	0 - 4 años
0 - 4 AÑOS	0 - 4 años
0-4	0 - 4 años
0-9	0 - 9 años
0-4 años	0 - 4 años
0-20 Min	0 - 20 min
05 - 0 años	0 - 5 años
	...

- **Normalización de identificadores** Con una lógica similar a la de descriptores, es decir, mediante una plantilla Excel pero sin considerar anidaciones, ya que, el campo de identificador solo tiene una única anidación, se extrajeron todos los valores únicos para aquel campo contenidos en el consolidado JSON Serie A, y a través de un script en Python se normalizó la mayoría de los valores y fueron asignados a la columna *Identificadores normalizados*, pero algunos tuvieron que ser revisados de manera manual por discrepancia de formato.

Tabla X

Extracción y asignación de identificadores normalizados en plantilla Excel.

Identificador	**Identificadores normalizados**
1 a 4 Años	1 - 4 años
1 - 4 AÑOS	1 - 4 años
10A14 años	10 - 14 años
10A19	10 - 19 años
10 a 24 años	10 - 24 años
12-23 HORAS	12 - 23 horas
	...

Una vez completadas cada plantilla Excel se implementó un script en Python que realizó un cruce con el consolidado JSON original para asignar los nuevos valores a cada campo específicado como se puede apreciar en la Fig. 4.3.7.


```
{
    "01010101": {
        "codigo": "01010101",
        "serie": "A",
        "hoja": "A01",
        "contexto": "CONTROLES DE SALUD",
        "año": 2009,
        "seccion": "SECCIÓN A: CONTROLES DE SALUD DE LA MUJER",
        "identificador": "pre-\nconcepcional atendido por medico",
        "descriptores": {
            "total": "COL1",
            "Grupo etario": {
                "6 - 9 años": "COL2",
                "10 - 14 años": "COL3",
                "15 - 19 años": "COL4",
                "20 - 24 años": "COL5",
                "25 - 34 años": "COL6",
                "35 - 44 años": "COL7",
                "45 - 54 años": "COL8",
                "55 - 64 años": "COL9",
                "65 años y más": "COL10"
            },
            "Por sexo": {
                "hombres": "COL11",
                "mujeres": "COL12"
            },
            "beneficiarios": "COL13"
        }
    },
```


Figura 4.3.7
Resultado del cruce del archivo JSON con las plantillas Excel para la normalización de los campos específicados.

4.4. Implementación de interfaz para consulta y visualización de datos.

Para la implementación de la interfaz se hizo un script en Python donde se usó la librería de Streamlit debido a sus herramientas que permiten la visualización como selección de datos que son necesarios para un buen manejo por parte del usuario, detallando los aspectos mostrados en el diagrama de flujo de la Fig. 4.4.1 que se consideraron para la implementación de consulta.

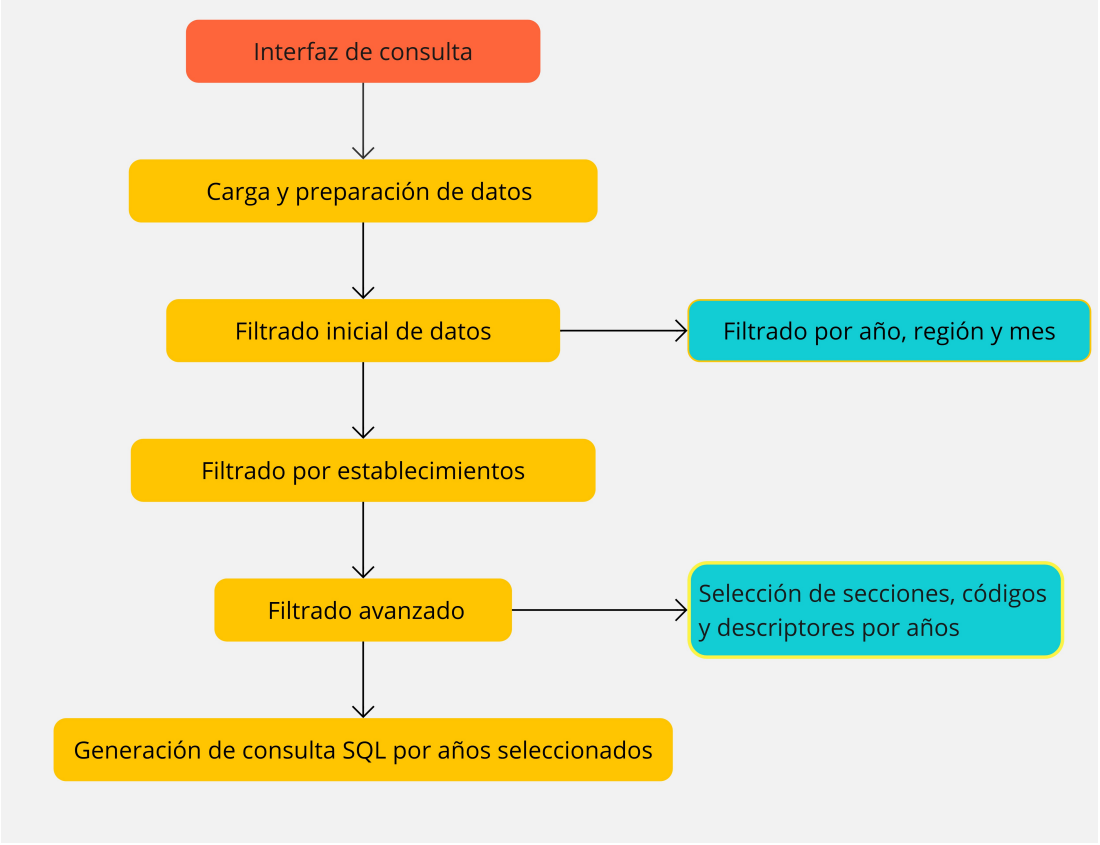


Figura 4.4.1
Diagrama flujo de interfaz para consulta.

4.4.1. Carga y preparación de datos.

Los archivos necesarios para el correcto funcionamiento de la plataforma consistieron en el consolidado del formato decodificador Serie A, los listados de establecimientos emitidos por el MINSAL y un archivo CSV que contiene todas las combinaciones únicas para las columnas *CodigoPrestacion* e *IdEstablecimiento* extraídas directamente del consolidado de la base de datos para la Serie A, como también, las siguientes librerías que se detallan en la tabla XI.

Tabla XI
Descripción de librerías utilizadas.

Librerías	**Descripción**
Streamlit	Esta librería se utiliza para crear una interfaz de usuario interactiva y manejar el estado de la aplicación web. Permite mostrar elementos interactivos como formularios, selectores de múltiples opciones y casillas de verificación, y presentar resultados y mensajes al usuario.
Pandas	Esta librería es para el análisis y manipulación de datos. Se usa para cargar datos desde archivos Excel y CSV, renombrar columnas, y realizar diversas operaciones de filtrado y selección de datos en DataFrames.
JSON	Esta librería se emplea para leer y manipular datos en formato JSON. Se utiliza para cargar un diccionario desde un archivo JSON, el cual es procesado y filtrado según los requisitos de la aplicación.

4.4.2. Filtro inicial.

El filtro inicial consitió en un listado con todos los años, meses y regiones disponibles a consultar.

Al seleccionar *Aplicar filtros iniciales* estos valores quedan guardados para su posterior consulta SQL.

Figura 4.4.2
Visualización plataforma Streamlit con filtros iniciales.

4.4.3. Filtro por establecimientos según región.

El siguiente filtro funciona en base a la información extraída desde los listados de establecimiento donde se realiza una normalización de las columnas *Código Antiguo Establecimiento* y *Código Nuevo Establecimiento* y se extrae el *Nombre Oficial* del establecimiento para su visualización en el selector mostrados en la Fig. 4.4.3 y 4.4.4.

Por último, en la implementación de la interfaz se presentaron algunos problemas con respecto al filtrado de establecimientos. Durante las consultas SQL de prueba, se observó que ciertos establecimientos no tenían los códigos de prestación consultados. Para solucionar esta situación, se extrajo un archivo CSV que contenía todas las combinaciones únicas de *IdEstablecimiento* y *CodigoPrestacion* directamente de la base de datos generada. Esto permitió asegurar consultas SQL correctas al extraer únicamente información válida según los parámetros deseados.

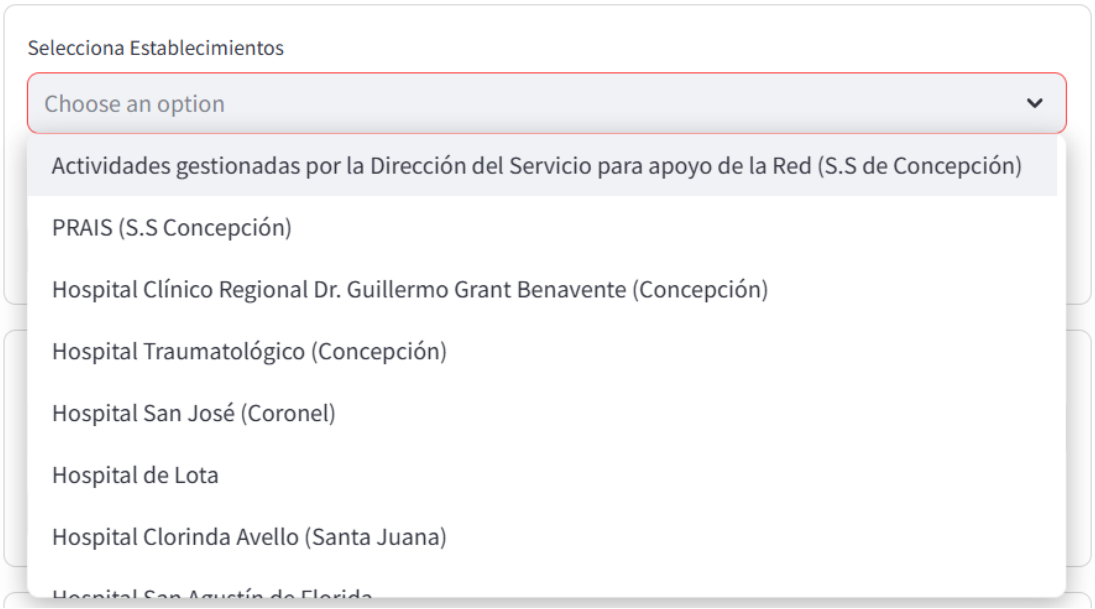


Figura 4.4.3
Visualización plataforma Streamlit con selector por establecimiento.

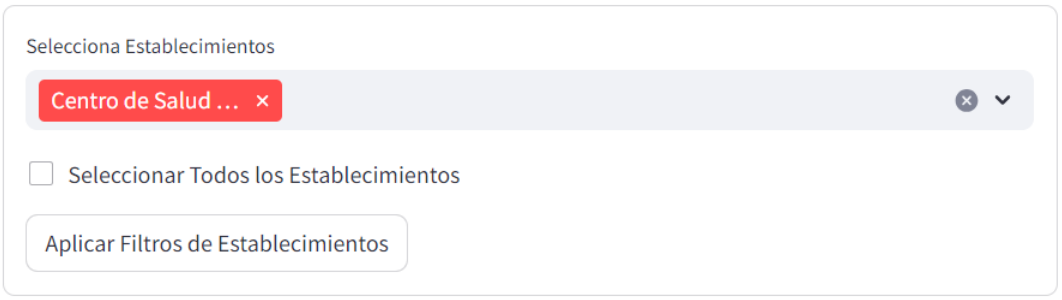


Figura 4.4.4
Visualización plataforma Streamlit con filtros por establecimiento.

4.4.4. Filtro avanzado de descriptores por sección.

Para este filtro se extrae la información desde el diccionario decodificador de Serie A y mediante un cruce con los valores contenidos en *codigo* para cada entrada se identifican que secciones, códigos de prestación junto a su identificador y en consecuencia descriptores estarán disponibles para el o los establecimientos que se quieren consultar gracias al archivo CSV que tiene las combinaciones únicas que incluyen *IdEstablecimiento* y *CodigoPrestacion*.

Se aseguró que para sección esta pueda ser seleccionada como un único valor debido a que cada una significa una tabla distinta y por ende contiene códigos junto a identificadores como descriptores diferentes y que pueden ser visualizados como se muestra en la Fig. 4.4.5.

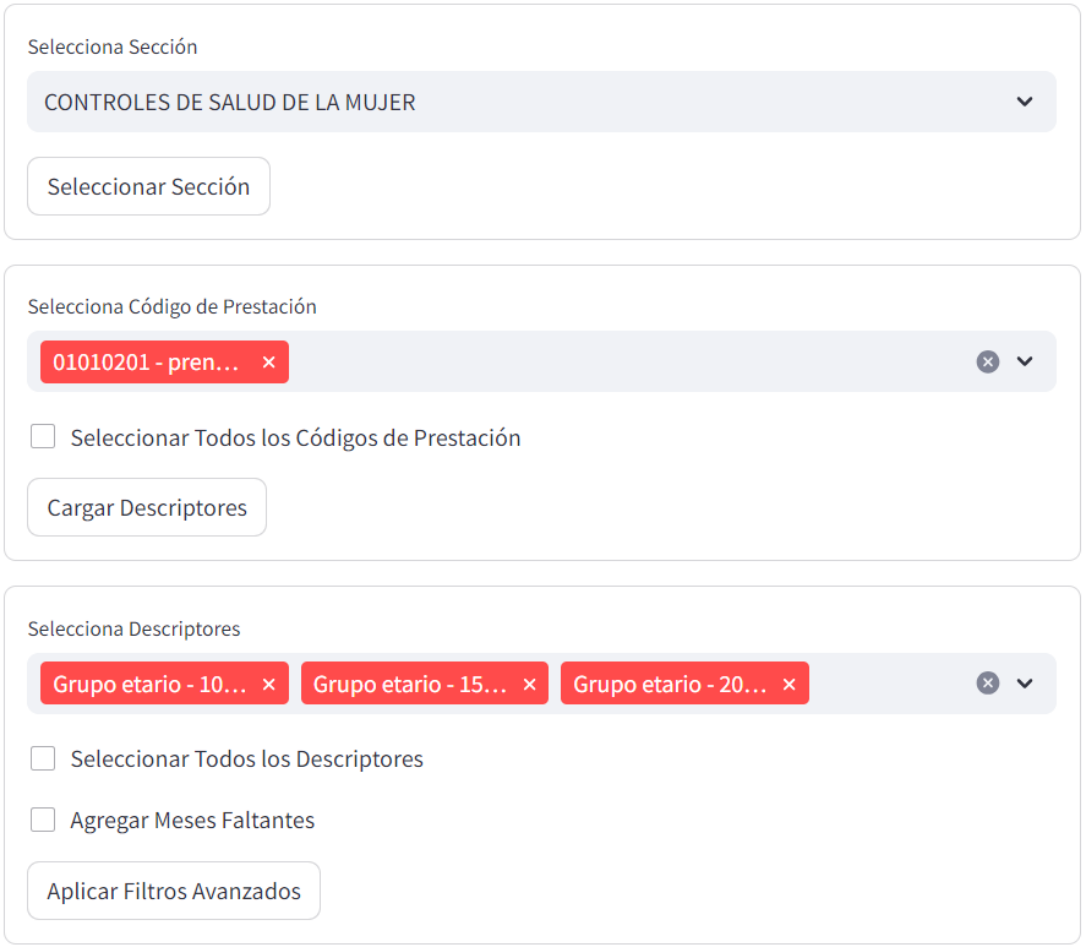


Figura 4.4.5
Visualización de filtro avanzado en plataforma Streamlit.

4.4.5. Consulta SQL.

Al seleccionar *Hacer consulta* mostrada en la Fig. **??** y 4.4.7 se generan consultas SQL independientes para cada año que fue consultado asegurando la integración de cada campo que fue seleccionado en filtros anteriores. Destacando que por motivos de visualización no se puede apreciar completamente la consulta SQL al tener tantos carácteres por lo que a continuación se detallan estas mismas en donde la consulta es la misma diferenciandose solo por el año consultado, es decir, que todos los campos se mantienen igual, mientras que, varía el año específicado.

La consulta está definida de la siguiente manera '**SELECT Ano, Mes, IdRegion, IdEstablecimiento, CodigoPrestacion, Col03, Col04, Col05 FROM 'copia de base de datos' WHERE Ano = '2009' AND Mes IN ('1', '2', '3',**

'4', '5', '7', '6') AND IdRegion IN ('8') AND IdEstablecimiento IN ('28-311', '128311') AND CodigoPrestacion IN ('01010201')'. Donde son detalladas las funciones en la tabla XII.

Tabla XII
Descripción de consulta SQL.

Comando	**Función**	**Ejemplo y Descripción**
SELECT	Especifica las columnas que desean extraer de la base de datos.	Ejemplo: `SELECT Ano, Mes, IdRegion, IdEstablecimiento, CodigoPrestacion, Col03, Col04, Col05` Esto indica que deseas obtener las columnas `Ano`, `Mes`, `IdRegion`, `IdEstablecimiento`, `CodigoPrestacion`, `Col03`, `Col04` y `Col05`.
FROM	Especifica la tabla de la cual se desea extraer los datos.	Ejemplo: `FROM 'copia de base de datos'` Esto indica que los datos deben ser recuperados de la tabla llamada `'copia de base de datos'`.
WHERE	Filtra los registros que cumplen con ciertas condiciones.	Ejemplo: `WHERE Ano = '2009'` Esto filtra los registros para que solo se incluyan aquellos donde el año (`Ano`) es `2009`.
AND	Combina múltiples condiciones en la cláusula `WHERE`.	Ejemplo: `AND Mes IN ('1', '2', '3', '4', '5', '7', '6')` Esto añade una condición adicional que solo incluye los registros donde el mes (`Mes`) es uno de los valores listados (`1`, `2`, `3`, `4`, `5`, `7`, `6`).

Continúa en la siguiente página

Tabla XII
Descripción de consulta SQL (continuación).

Comando	**Función**	**Ejemplo y Descripción**
IN	Especifica múltiples valores posibles para una columna.	Ejemplo: `AND IdRegion IN ('8')` Esto añade una condición que solo incluye los registros donde la región (`IdRegion`) es `8`. Otro Ejemplo: `AND IdEstablecimiento IN ('28-311', '128311')` Esto añade una condición que solo incluye los registros donde el ID del establecimiento (`IdEstablecimiento`) es `28-311` o `128311`. Otro Ejemplo: `AND CodigoPrestacion IN ('01010201')` Esto añade una condición que solo incluye los registros donde el código de prestación (`CodigoPrestacion`) es `01010201`.

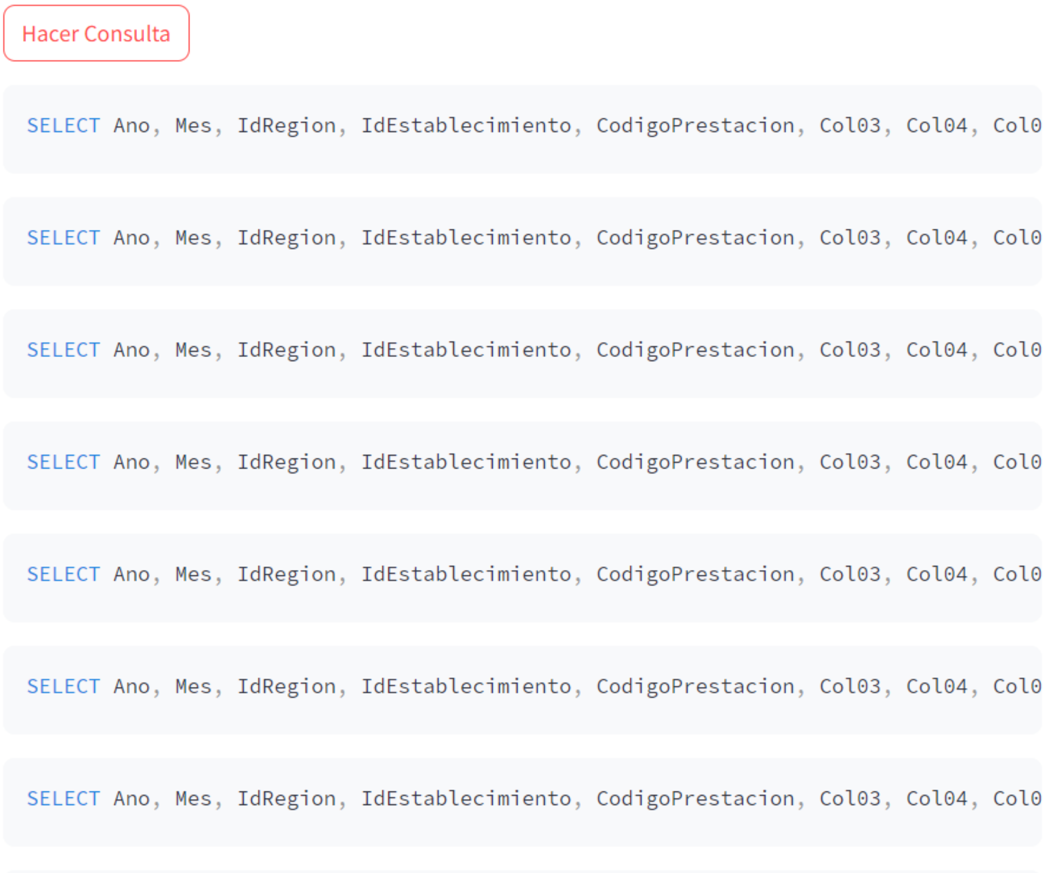


Figura 4.4.6
Generación de consulta SQL en plataforma Streamlit 1ra parte.

```
 WHERE Ano = '2009' AND Mes IN ('1', '2', '4', '3', '5', '6') AND IdRegion IN ('8')
```

```
WHERE Ano = '2010' AND Mes IN ('1', '2', '4', '3', '5', '6') AND IdRegion IN ('8')
```

```
a WHERE Ano = '2011' AND Mes IN ('1', '2', '4', '3', '5', '6') AND IdRegion IN ('8')
```

```
WHERE Ano = '2012' AND Mes IN ('1', '2', '4', '3', '5', '6') AND IdRegion IN ('8')
```

```
) AND IdEstablecimiento IN ('28-311', '128311') AND CodigoPrestacion IN ('01010201')
```

```
) AND IdEstablecimiento IN ('28-311', '128311') AND CodigoPrestacion IN ('01010201')
```

```
) AND IdEstablecimiento IN ('28-311', '128311') AND CodigoPrestacion IN ('01010201')
```

```
) AND IdEstablecimiento IN ('28-311', '128311') AND CodigoPrestacion IN ('01010201')
```

```
) AND IdEstablecimiento IN ('28-311', '128311') AND CodigoPrestacion IN ('01010201')
```

Figura 4.4.7
Generación de consulta SQL en plataforma Streamlit 2da parte.

4.5. Discusión.

En el desarrollo de este trabajo se desarrolló una base de datos con un formato específicado con el fin de facilitar el uso con el formato decodificador generado haciendo uso de herramientas de visualización para la consulta SQL en base a archivos dependientes para una decodificación de la información consultada. El primer paso fue consolidar la base de datos, lo cual presentó un problema inicial debido a que los archivos de texto se leían de manera diferente diferenciandose por su delimitador siendo para TXT el delimitador `";"`, mientras que, el delimitador para CSV era `","`. Sin embargo, la principal complicación se debió al tamaño de los archivos de la Serie A, los cuales oscilaban entre los 200 y 550 MB de manera individual. Debido a este tamaño, el entorno de trabajo no podía concatenar todos los archivos de los años 2009 a 2023 en uno solo sin agotar la memoria disponible. Para resolver este problema, se implementó el uso del método de chunking, que facilitó el proceso al recopilar los primeros 10,000 datos de cada archivo y exportarlos directamente a un directorio designado para luego eliminar los 10,000 datos procesados y seguir con los siguientes.

El segundo y más importante desafío fue la generación de un formato decodificador. Esta tarea resultó considerablemente ardua debido a la necesidad de unificar todos los diccionarios de cada año, los cuales presentaban una disposición de tablas variable a lo largo del tiempo. Se probaron varios métodos para estructurar la información, y la extracción de los campos mencionados pero el método usado en la sección 4.2.1 resultó ser la más efectiva, ya que agilizó el proceso de generación al extraer la información de manera automatizada directamente desde los diccionarios de la Serie A. En cuanto a la generación de descriptores, la reciente versión de Chat GPT 4-O demostró ser útil. En la generación de descriptores, no se optó por utilizar un script en Python debido a la complejidad de las tablas que presentaban los diccionarios de la Serie A. A lo largo de los años, estos diccionarios mostraron una disposición variable de las tablas, lo que habría requerido la creación de múltiples condicionales para cada posible variación. Esto habría dificultado enormemente la automatización del proceso de extracción de descriptores. En contraste de otros campos en donde aquellas informaciones como *sección*, *código de prestación*, *contexto* e *identificadores* mantenían un patrón constante en la disposición de columnas a lo largo del tiempo, lo que hacía viable su automatización.

Por último, se implementó un mecanismo en la generación del SQL para el campo de establecimiento que asignaba tanto el código numérico antiguo como el nuevo al realizar consultas para el mismo establecimiento. Esto se hizo para garantizar que, en caso de que no se encontrara información con el código antiguo, se pudiera localizar con el nuevo, ya que la información podía estar registrada de manera diferente según el año. Todo esto resultó en una interfaz capaz de manejar la información deseada para una generación de consultas SQL.

Capítulo 5

Resultados.

5.1. Introducción.

En este trabajo, se logró consolidar y normalizar una base de datos de 7GB a partir de los archivos individuales REM de la Serie A, abarcando el período desde 2009 hasta 2023. Además, se desarrolló un formato decodificador en JSON para facilitar la interpretación de estos datos. Finalmente, se implementó una interfaz que permite la consulta de la información consolidada.

5.2. Base de datos.

Se logró una base de datos resultado de la concatenación de archivos individuales REM Serie A que datan del año 2009 hasta 2023 donde se aseguró una normalización de sus columnas debido a la discrepancia de formatos a través de los años, ya sea por la etiquetación de sus columnas, y de esta manera logrando una unificación en un formato CSV. La concatenación y unificación de estos archivos dió como resultado una base datos para Serie A.

5.3. Formato decodificador.

Con respecto al formato decodificador este consistió en una unificación de archivos en formato JSON en los cuales está extraída y normalizada su información anteriormente dicha que constan de los diccionarios Serie A de los años 2009 hasta 2023.

5.4. Implementación de interfaz para consulta y visualización de datos.

La interfaz resultó ser una herramienta que permite la consulta de campos como los años, meses, regiones, establecimientos, secciones, códigos de prestación y descriptores para la generación de consulta SQL y posterior extracción y visualización de datos consultados para la decodificación de aquellos campos que de manera manual hubiesen sido de muy complejo acceso y obtención como se presentó en la Fig. 3.1.1, y de esta manera relacionando los archivos dependientes para la generación de esta consulta.

5.5. Visualización de consulta SQL.

Las consultas SQL se fueron probando en un prototipo de script en Python en donde se podía consultar de manera independiente una consulta SQL la cual es mostrada en la tabla XIII directamente a una base de datos local acotada a solo la Sección A: Controles de salud sexual de la mujer, dando como resultado un dataframe con la información solicitada. Estos datos solo fueron usados como ejemplo no implica que sean lo más importantes para la base de datos de la Serie A.

Donde la interpretación de la consulta visualizada en las Fig. 5.5.1 y 5.5.2, es controles de salud realizados en el Centro de Salud Familiar Lebu Norte, ubicado en la región del Bío-Bío, durante el primer semestre de los años 2009 a 2023. El enfoque está en los controles de salud de la mujer, específicamente los prenatales, atendidos por un médico, para el grupo etario de entre 10 y 24 años. Esta consulta también abarca los otros años, es decir, 2010 hasta 2023.

Tabla XIII
Consulta SQL de ejemplo.

Consulta SQL realizada
SELECT Ano, Mes, IdRegion, IdEstablecimiento, CodigoPrestacion, Col03, Col04, Col05 FROM 'copia de base datos' WHERE Ano = '2009' AND Mes IN ('1', '2', '3', '4', '5', '7', '6') AND IdRegion IN ('8') AND IdEstablecimiento IN ('28-311', '128311') AND CodigoPrestacion IN ('01010201')

10 - 14 años	15 - 19 años	20 - 24 años	Seccion	Contexto
12	5	4	SECCIÓN A: CONTROLES DE SALUD DE LA MUJER	CONTROLES DE SALUD
5	4	9	SECCIÓN A: CONTROLES DE SALUD DE LA MUJER	CONTROLES DE SALUD
None	2	6	SECCIÓN A: CONTROLES DE SALUD DE LA MUJER	CONTROLES DE SALUD
5	4	4	SECCIÓN A: CONTROLES DE SALUD DE LA MUJER	CONTROLES DE SALUD
2	2	6	SECCIÓN A: CONTROLES DE SALUD DE LA MUJER	CONTROLES DE SALUD

Figura 5.5.1
Visualización de base de datos filtrada con Pandas parte 1.

CodigoPrestacion	Mes	AÃ±o	RegionName	NOMBRE_ESTABLECIMIENTO
01010201 - prenatal atendido por medico	Abril	2,009	Bío-Bío	Centro de Salud Familiar Lebu Norte
01010201 - prenatal atendido por medico	Julio	2,009	Bío-Bío	Centro de Salud Familiar Lebu Norte
01010201 - prenatal atendido por medico	Mayo	2,009	Bío-Bío	Centro de Salud Familiar Lebu Norte
01010201 - prenatal atendido por medico	Enero	2,009	Bío-Bío	Centro de Salud Familiar Lebu Norte
01010201 - prenatal atendido por medico	Marzo	2,009	Bío-Bío	Centro de Salud Familiar Lebu Norte

Figura 5.5.2
Visualización de base de datos filtrada con Pandas parte 2.

5.6. Discusión.

La consolidación de los archivos REM en una base de datos unificada y normalizada representó la importancia de la estandarización de información en la gestión de datos abiertos de salud pública.

El desarrollo del formato decodificador en JSON se hizo con el fin de estructurar los diccionarios REM Serie A y facilitar la decodificación de la base de datos. Este formato no solo facilita la integración con sistemas de consulta, sino que también permite que la información se interprete con lo extraído desde el formato establecido en JSON, reduciendo así el riesgo de errores en la interpretación de estos campos.

La implementación de una interfaz que permite la generación de consultas SQL se desarrolló para poder extraer información específica desde el formato decodificador, realizar un cruce con archivos dependientes para garantizar la coherencia con lo que se quería consultar y presentar una visualización para el usuario. Esta herramienta permite a los usuarios realizar consultas SQL y obtener resultados de manera rápida que de otra forma implicaría una búsqueda manual extensa. La capacidad de consultar por diversos campos, como años, meses, regiones y establecimientos, y de decodificar información como descriptores, mejora la accesibilidad e interpretación de los datos.

La consolidación y normalización de los archivos REM, junto con el desarrollo de herramientas de decodificación y consulta, representan la importancia de la estandarización y la automatización en el manejo de grandes volúmenes de datos.

Capítulo 6

Conclusión y Trabajo Futuro.

Este proyecto ha logrado consolidar y normalizar una base de datos extensa a partir de los archivos REM de la Serie A, cubriendo el período de 2009 a 2023. La unificación de estos archivos en un formato común ha logrado estructurar todos estos años en mismas columnas y etiquetando al año que corresponden. El desarrollo de un formato decodificador en JSON ha permitido estructurar la información de los diccionarios REM Serie A de manera que se puede extraer información directamente de este archivo, reduciendo el riesgo de errores y mejorando la calidad de extracción de información.

La implementación de una interfaz para la consulta de esta base de datos ha permitido a los usuarios realizar consultas SQL y estructurar la información según año, meses, regiones, establecimientos, códigos de prestación, secciones y descriptores de manera ordenada. Esto no solo ayuda al acceso a la información, sino que también a la estandarización y automatización en el manejo de grandes volúmenes de datos, mejorando la accesibilidad y utilidad de la información de datos abiertos en el sector de la salud pública.

Durante el desarrollo del proyecto, se abordaron varios procesos claves. La normalización de los archivos en formatos comunes permitió una mejor manipulación y coherencia en la extracción de información, facilitando la interpretación de los códigos de prestación. Se implementaron métodos como el procesamiento por chunks para manejar grandes volúmenes de datos y la generación de un formato decodificador en JSON para estructurar los datos. La interfaz implementada permitió realizar consultas SQL. La consolidación de

los archivos REM en una base de datos unificada y normalizada representó la importancia de la estandarización y la automatización en la gestión de datos de salud pública.

Aunque se ha avanzado en la consolidación y normalización de los datos, quedan varios aspectos por desarrollar para mejorar el potencial de esta herramienta:

Actualizar con años recientes: La base de datos debe actualizarse periódicamente para incluir datos de años recientes, comenzando con la inclusión de los datos de 2024.

Trabajar con otras Series: Además de la Serie A, existen otras series como P, BM, BS y D en los archivos REM. Incluir estas series en el proyecto aumentará la amplitud y profundidad de los datos disponibles, proporcionando una visión más completa de la salud pública.

Desarrollar lógica para análisis estadísticos: Implementar análisis estadísticos con la información extraída, como visualización de gráficos, cálculo de promedios, sumas y otras métricas relevantes. Esto permitirá no solo la consulta de datos, sino también la interpretación y análisis en profundidad, facilitando la toma de decisiones basadas en evidencia.

Implementación de API: Desarrollar una API para facilitar el acceso y consulta de la base de datos consolidada. Esta API permitirá que otras aplicaciones y sistemas se integren fácilmente con la base de datos, mejorando la accesibilidad y la interoperabilidad.

Implementación de API para consultas cruzadas con las bases de datos: Desarrollar una API que permita realizar consultas cruzadas entre las diferentes series de la base de datos. Esta funcionalidad permitirá a los usuarios obtener una visión más completa y detallada de los datos, facilitando el análisis comparativo.

Estos pasos futuros no solo mejorarán la funcionalidad de la herramienta actual, sino que también ampliarán su aplicabilidad y valor, asegurando que continúe siendo una herramienta crucial para la gestión y análisis de datos de salud pública, como también, para el uso privado, de empresas, como la auspiciadora de este proyecto HealthTracker (4).

Capítulo 7

Glosario.

REM	: Resúmenes Estadísticos Mensuales.
JSON	: JavaScript Object Notation.
CSV	: Comma-Separated Values.
API	: Application Programming Interface.
SQL	: Structured Query Language.
MINSAL	: Ministerio de Salud.
LLM	: Large Language Model.
GPT	: Generative Pre-trained Transformer.
TXT	: Plain Text File.
XLS	: Excel Spreadsheet.
ZIP	: Zipped.

Bibliografía

[1] Ministerio de Salud de Chile, "Manual Series REM 2023 - Departamento de Estadísticas e Información de Salud," https://deis.minsal.cl/#datosabiertos, [Online; accessed 1-May-2024].

[2] M. de Salud Gobierno de Chile, "Manual series rem 2024," 2024, accessed: 2024-07-20. [Online]. Available: https://estadistica.ssmso.cl/series-rem-2024/

[3] Ministerio de Salud de Chile, "Reportes Estadísticos Mensuales - REM," https://reportesrem.minsal.cl/?_token=zR6KI44R7ac4XmpOLoSE4qAWD5PdJjLLDai5BO5k&serie=1&rem=61&seccion_id=687&tipo=4&tipoReload=4&regiones=-1&regionesReload=15&servicios=0&serviciosReload=0&periodo=2019&mes_inicio=1&mes_final=12, [Online; accessed 2-May-2024].

[4] HealthTracker, "Healthtracker," https://healthtracker.ai/, 2024, accessed: 2024-08-02. [Online]. Available: https://healthtracker.ai/

[5] Departamento de Estadísticas e Información de Salud (DEIS), "Diccionario de la serie a 2009," https://deis.minsal.cl/#datosabiertos, 2009, accedido el: 06-May-2024.

[6] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.06196

[7] K. I. Roumeliotis and N. D. Tselikas, "Chatgpt and open-ai models: A preliminary review," 2023. [Online]. Available: https://www.mdpi.com/1999-5903/15/6/192

[8] "Display your application data with streamlit," 2023, accessed: 2024-07-01. [Online]. Available: https://www.redhat.com/sysadmin/streamlit-display-data

[9] "Streamlit • a faster way to build and share data apps," 2024, accessed: 2024-07-01. [Online]. Available: https://streamlit.io/

[10] "Unlock faster data processing in python: The one trick to supercharge your pandas code," 2024, accessed: 2024-06-20. [Online]. Available: https://acortar.link/FzIKQG

[11] "What is data consistency? definition, examples and best practice," 2024, accessed: 2024-06-20. [Online]. Available: https://www.decube.io/post/what-is-data-consistency-definition-examples-and-best-practice

[12] "The pandas merge method - professional pandas series," 2022, accessed: 2024-06-20. [Online]. Available: https://ponder.io/professional-pandas-the-pandas-merge-method/

[13] Turing, "Json: Introduction, benefits, applications, and drawbacks," 2024, accessed: 2024-07-02. [Online]. Available: https://www.turing.com/kb/what-is-json

[14] Stanford University, "Introduction to Database and Structured Query Language (SQL)," https://codethechange.stanford.edu/, [Online; accessed 1-May-2024].

[15] S. de F. Mendes Sampaio, C. Dong, and P. Sampaio, "Dq2s – a framework for data quality-aware information management," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8304–8326, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417415004522

[16] Ministerio de Salud de Chile, "Datos Abiertos - DEIS," https://deis.minsal.cl/#datosabiertos, [Online; accessed 2-May-2024].

[17] Departamento de Estadísticas e Información de Salud (DEIS), "Listado de establecimientos," https://deis.minsal.cl/#datosabiertos, 2024, accedido el: 06-May-2024.

Apéndice A

Anexos.

UNIVERSIDAD DE CONCEPCIÓN – FACULTAD DE INGENIERÍA
RESUMEN DE MEMORIA DE TÍTULO

Departamento : Departamento de Ingeniería
Carrera : Ingeniería Civil Biomédica
Nombre del memorista : Maximiliano Agustín Araya Morales
Título de la memoria : Desarrollo de una interfaz para la consulta y visualización de datos de salud pública del Resumen Estadístico Mensual del MINSAL.

Fecha de presentación oral: 28-09-2024
Profesor(es) Guía : **Pamela Guevara Alvez**
Comisión : **Rosa Figueroa Iturrieta**
Supervisor externo : **Jaime Jiménez Ruiz**
Concepto :
Calificación :

Resumen
El objetivo es desarrollar una interfaz para la consulta y visualización de los datos de salud pública contenidos en los REM emitidos por el MINSAL. Esta herramienta está destinada a la empresa privada HealthTracker y busca mejorar la accesibilidad y manejo de los datos de la Serie A de los REM, que abarcan desde 2009 hasta 2023.
Se aborda la necesidad de mejorar la gestión y análisis de los datos de salud, los cuales están dispersos y en distintos formatos. Para ello, se planificaron etapas que incluyen la descarga y consolidación de archivos REM, análisis de diccionarios de datos, creación de un formato decodificador en JSON, e implementación de una interfaz para consultar estos datos.
Durante el desarrollo, se utilizaron técnicas como el método de chunks para manejar grandes volúmenes de datos, procesando archivos de hasta 500 MB sin agotar la memoria del sistema. Además, se emplearon herramientas avanzadas de LLM para generar descriptores y el método de cruce para normalizar datos, agilizando la decodificación.
Los resultados incluyen una base de datos, un formato decodificador en JSON, y una interfaz con Streamlit que permite realizar consultas SQL. Este proyecto ayuda a la accesibilidad y gestión de los datos de salud pública, proporcionando una herramienta valiosa para HealthTracker y para la gestión de datos abiertos de salud en Chile.