UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

INSTITUTO DE INGENIERÍA

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA

“GESTIÓN DE DATOS HETEROGÉNEOS ASOCIADOS AL COVID-19 EN MÉXICO”

TESIS PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

PRESENTA:

CARLOS ALONSO ESPINOZA GARCÍA

DIRECTOR DE TESIS:

DR. GABRIEL LÓPEZ MORTEO

CODIRECTOR DE TESIS:

DR. FRANCISCO MUÑOZ-ARRIOLA

MEXICALI, B.C. **AGOSTO 2021**

TESIS DEFENDIDA POR

Carlos Alonso Espinoza García

Y aprobada por el siguiente comité:

_____

Dra. Brenda Leticia Flores Ríos

Presidente del Comité

_____ _____

Dr. Félix Fernando González Navarro

Miembro del Comité

Dr. Gabriel Alejandro López Morteo

Miembro del Comité

_____

Dr. Jesús Caro Gutiérrez

Miembro del Comité

Agosto del 2021

I. Resumen

La presente investigación describe el sistema núcleo COVID-19 México y su API, es un software desarrollado mayormente en Python con el propósito de almacenar los datos referentes a COVID-19 en México que expone la federación por medio del portal de la secretaría de salud. Se realizan procesos de descarga automática diaria, extracción, manipulación y transformación de datos. La arquitectura que se implementó está basada en el funcionamiento de plugin, lo cual permite que sea empleada por terceros. Se utilizó una arquitectura en capas para el funcionamiento interno del software. El sistema es tolerante a cambios o actualizaciones. Se utiliza PostgreSQL en la capa de persistencia debido a que los datos obtenidos son de origen relacional. Para el desarrollo de la capa de persistencia se implementaron herramientas de data warehouse y transformación. Esto quiere decir que se emplean entidades de datos, tablas que tienen contexto específicos para el caso de análisis que se requiera. En la documentación se describen los procesos para ETL que se llevaron a cabo. También se creó un servicio que realiza las consultas a la base de datos, esto con el fin de ser utilizado por terceros con perfil técnico en computación. El objetivo principal es hacer llegar set de datos al investigador que no cuenta con perfil de Ciencias de la computación y éste requiere analizar dicha información. Aunque la plataforma se encarga de recolectar datos de COVID-19 en México se pueden integrar distintas fuentes de datos para facilitar el análisis. Ambos sistemas fueron validados con respecto al almacenamiento, gestión y conexión. Los resultados fueron comparados con distintas plataformas de información que actualmente se utilizan para corroborar números de COVID-19 en México.

Palabras clave: API, arquitectura, COVID-19, data warehouse, ETL, PostgreSQL, plugin, Python, servicio, software.

II. Abstract

This research describes the core system COVID-19 Mexico and its API, it is a software developed mainly in Python with the purpose of storing data referring to COVID-19 in Mexico that the federation exposes through the portal of the health ministry. Daily automatic downloading, data extraction, manipulation and transformation processes are carried out. The architecture that was implemented is based on the operation of the plugin, which allows it to be used by third parties. A layered architecture was used for the internal workings of the software. The system is tolerant to changes or updates. PostgreSQL is used in the persistence layer because the data obtained is of relational origin. For the development of the persistence layer, data warehouse and transformation tools were implemented. This means that data entities are used, tables that have specific context for the required analysis case. The documentation describes the processes for ETL that were carried out. A service was also created that performs the queries to the database, this in order to be used by third parties with a technical profile in computing. The main objective is to send a set of data to the researcher who does not have a Computer Science profile and he / she needs to analyze said information. Although the platform is responsible for collecting COVID-19 data in Mexico, different data sources can be integrated to facilitate analysis. Both systems were validated with respect to storage, management and connection. The results were compared with different information platforms that are currently used to corroborate COVID-19 numbers in Mexico.

Keywords: API, architecture, COVID-19, data warehouse, ETL, PostgreSQL, plugin, Python, service, software.

Agradecimientos

Quiero agradecer a mi familia que siempre estuvo a mi lado apoyándome en cada una de las etapas de este proceso. Mi esposa Lucia que sin su comprensión y apoyo nada de esto hubiera sido posible.

Le tengo un agradecimiento profundo a mi director de tesis, el Dr. Gabriel López Morteo, ya que es parte fundamental de este trabajo. Sus enseñanzas y dedicación hicieron posible todo esto y además encontré un gran amigo en él. A mi codirector de tesis, el Dr. Francisco Muñoz Arriola, gracias a sus amplios conocimientos y comentarios que dieron frutos al presente trabajo. La paciencia que tuvo en estos tiempos de COVID-19 fue algo que atesoraré siempre.

También un agradecimiento especial al M.C Rosendo Sosa Canales, quien me presentó con el programa de posgrado.

A los miembros de mi comité de tesis, por su disposición y retroalimentación durante los avances de investigación. Al Dr. Félix Fernando González Navarro que con sus conocimientos me inspiró a investigar sobre el tema de datos.

A mis compañeros de posgrado, Emanuel, Karla, Paola. Gracias a ellos la estancia fue placentera y llena de nuevos conocimientos.

Al CONACyT, por el apoyo económico otorgado a través de la beca nacional.

Mexicali, Baja California, México. Carlos Alonso Espinoza García

Agosto del 2021.

Índice

CAPÍTULO 1. INTRODUCCIÓN 1

1.1 DEFINICIÓN DEL PROBLEMA 4
1.2 JUSTIFICACIÓN 6
1.3 PREGUNTAS DE INVESTIGACIÓN 7
1.4 OBJETIVOS 7
1.4.1 OBJETIVO GENERAL 7
1.4.2 OBJETIVOS ESPECÍFICOS 7
1.5 ESTRUCTURA DEL DOCUMENTO 8

CAPÍTULO 2. MARCO TEÓRICO 9

2.1 DATOS ABIERTOS EN SECTOR SALUD NACIONAL 9
2.2 FUENTES DE DATOS 10
2.2.1 DATOS PUNTUALES 10
2.2.2 DATOS ESPACIALES 10
2.2.3 DATOS DE SERIES DE TIEMPO 11
2.2.4 FUENTES HISTÓRICAS 11
2.2.5 FUENTES EN TIEMPO REAL 11
2.2.6 FUENTES DE DATOS EPIDEMIOLÓGICAS NACIONALES 11
2.3 VARIABLES EPIDEMIOLÓGICAS 13
2.4 DEPENDENCIAS DE INFORMACIÓN 14
2.4.1 ORGANIZACIÓN MUNDIAL DE LA SALUD 14
2.4.2 SECRETARÍA DE SALUD DEL GOBIERNO DE MÉXICO 14
2.5 LA EPIDEMIA COVID-19 15
2.5.1 PANORAMA MUNDIAL 15
2.5.2 PANORAMA NACIONAL 16
2.6 BASES DE DATOS HETEROGÉNEAS 16
2.6.1 BASES DE DATOS RELACIONALES 18
2.6.2 BASES DE DATOS NO RELACIONALES 20
2.7 DATA WAREHOUSE 22
2.7.1 APLICACIONES 23
2.7.2 PREPROCESAMIENTO DE DATOS 24
2.7.3 MANIPULACIÓN DE SET DE DATOS 24
2.8 ARQUITECTURAS 24
2.8.1 ARQUITECTURAS DE PLUGIN 25
2.8.2 MIDDLEWARE 26
2.8.3 SERVICIOS 27
2.9 INTERFAZ DE PROGRAMACIÓN DE APLICACIONES (API) 28
2.10 FORMATOS DE SET DE DATOS 28
2.10.1 DOCUMENTOS XML 28
2.10.2 FORMATO JSON 28
2.10.3 ARCHIVOS CSV 29

2.11 PLATAFORMAS DE INFORMACIÓN 29
2.11.1 PLATAFORMAS INFORMATIVAS 29
2.11.2 PLATAFORMAS EXPLORATORIAS 30
2.11.3 PLATAFORMAS COMO REPOSITORIOS 30

CAPÍTULO 3. METODOLOGÍA 31

3.1 METODOLOGÍA DEL DESARROLLO 33

CAPÍTULO 4. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN DE SOFTWARE 37

4.1 ANÁLISIS 37
4.2 DISEÑO 37
4.3 DESCRIPCIÓN DE PROCESOS DEL SISTEMA NÚCLEO COVID-19 MÉXICO 40
4.4 FASES DE FUNCIONALIDAD 40
4.5 DESCRIPCIÓN FASE 1. SERVICIO DE DESCARGA DE DATOS 42
4.5.1 SERVICIO DE DESCARGA 42
4.5.2 PROCESO PARA DESCOMPRIMIR EL RECURSO 44
4.6 DESCRIPCIÓN FASE 2. PREPROCESAMIENTO DE LOS DATOS 44
4.6.1 DESDE DATOS DESCOMPRIMIDOS 45
4.6.2 DECODIFICACIÓN DE ARCHIVO 46
4.6.3 MANIPULACIÓN DE DATOS 46
4.7 DESCRIPCIÓN FASE 3 Y 4. PERSISTENCIA Y DATA WAREHOUSE 48
4.7.1 BASE DE DATOS EN POSTGRESQL 49
4.7.2 FUNCIONES EN PERSISTENCIA Y DATA WAREHOUSE 49
4.8 DESCRIPCIÓN FASE 5. FINALIZACIÓN 54
4.9 DESCRIPCIÓN DE PROCESOS DEL SERVICIO 54
4.9.1 ENDPOINTS O URLS 56
4.10 HERRAMIENTAS 57
4.11 VALIDACIÓN 58
4.11.1 CASO DE PRUEBA 1. EXTRACCIÓN DESDE EL SERVICIO Y ALMACENAMIENTO 59
4.11.2 CASO DE PRUEBA 2. COMPARACIÓN DE VARIABLES CON UNA MUESTRA MENSUAL 60
4.11.3 CASO DE PRUEBA 3. VERIFICAR DATOS DE ARCHIVO NO EXPUESTO 62
4.11.4 CASO DE PRUEBA 4. ENTIDAD DE CAMBIOS 63
4.11.5 CASO DE PRUEBA 5. COMPARACIÓN CON PLATAFORMAS INFORMATIVAS 64
4.11.6 CASO DE PRUEBA 6. SERVICIO Y BASE DE DATOS 67

CAPÍTULO 5. DISCUSIONES 69

CAPÍTULO 6. CONCLUSIONES Y TRABAJO FUTURO 74

6.1 TRABAJO FUTURO 76
6.1.1 DISEÑO 76
6.1.2 SEGURIDAD 77
6.1.3 ANÁLISIS DE DATOS 77

Lista de Figuras

Figura 1 Proceso de muestras en población. 13
Figura 2. Evolución de bases de datos.....18
Figura 3 . Ejemplo de base de datos no relacional por documento. 21
Figura 4. Middleware orientado a mensajes o datos. Extraído de middleware (weizmanc.blogspot.com) 27
Figura 5. Modelo de cascada....33
Figura 6. Diagrama de arquitectura Plugin. 35
Figura 7. Modelo arquitectónico. Elaboración propia.....39
Figura 8. Diagrama de funciones del sistema.....41
Figura 9. Diagrama de fase del servicio. Elaboración propia.....44
Figura 10. Proceso de preprocesamiento del archivo de datos. Elaboración propia....45
Figura 11 Ciclo de preprocesamiento de datos. Elaboración propia.....46
Figura 12. Orientación de los datos. Elaboración propia 47
Figura 13. Diagrama de interacción entre persistencia y data warehouse. Elaboración propia. 48
Figura 14. Ejemplo de catálogo de sectores. Elaboración propia. 49
Figura 15. Proceso de extracción de fechas por medio de iteraciones. Elaboración propia.....51
Figura 16. Descripción del ciclo diario. Elaboración propia.....52
Figura 17. Diagrama API REST 55
Figura 18. Gráficas de documentos vs base de datos. Elaboración propia 62
Figura 19. Gráfica de defunciones. Elaboración propia....62
Figura 20. Información de plataforma CONACyT al 31 de marzo del 2021. 66
Figura 21. Información de plataforma UNAM al 31 de marzo del 2021.....67
Figura 22. Comparativos entre plataformas y bases de datos. 67

Lista de tablas

Tabla 1. Puntos opcionales al tomar en cuenta para realizar la normalización en bases de datos. 20
Tabla 2. Datos que se manipulan en el proceso de escritura a la base de datos. 50
Tabla 3. Valores de aceptación en entidad de cambios. 52
Tabla 4. Herramientas utilizadas para API 58

Anexos

Anexo A. Bitácora de versiones de sistema núcleo COVID-19 México.....85
Anexo B. Bitácora de versiones API.....86
Anexo C. API endpoints.....86
Anexo D. Ejemplo de respuesta de API.....88
Anexo E. Tabla de estados y casos positivos.....89

III. Acrónimos

API: Application Programming Interface

BTree: Binary Tree

CPU: Central Processing Unit

CSS: Cascading Style Sheets

CSV: Comma Separated Values

DB: Database

DBMS: Database Management System

ETL: Extract, Transform, Load

EU: Estados Unidos

GB: Gigabyte

GIS: Geographic Information System

GNU: GNU's Not Unix

GPL: GNU General Public License

HTML: Hypertext Markup Language

JavaEE: Java Platform, Enterprise Edition

JPEG: Joint Photographic Experts Group

JSON: JavaScript Object Notation

KB: Kilobyte

MB: Megabyte**OGD:** Open Government Data

OLAP: Online Analytical Processing

OMS: Organización mundial de la salud

PB: Petabyte

PDF: Portable Document Format

PNG: Portable Network Graphics

PSQL: PostgreSQL

RAM: Random Access Memory

REST: Representational State Transfer

SEP: Secretaría de Educación Pública

SDK: Software Development Kit

SOAP: Simple Object Access Protocol

SQL: Structured Query Language

TB: Terabyte

UABC: Universidad Autónoma de Baja California

UNL: University of Nebraska-Lincoln

URL: Uniform Resource Locator

WHO: World Health Organization

XML: Extensible Markup Language

Capítulo 1. Introducción

En la actualidad, los conceptos de ciencias de datos o científicos de datos resultan de lo más usual, esto se debe al cambio global referente cantidad, velocidad y variedad de datos que se intercambian segundo a segundo. Esto ha generado que las entidades tanto privadas como públicas busquen favorecerse de dicho fenómeno.

Las empresas tecnológicas o aquellas que sus actividades recaen en el uso de herramientas tecnológicas buscan personal altamente calificado en ciencia de datos para extraer información, transformar o predecir comportamientos. Debido a que la economía y la industria son ecosistemas de constante fluctuación, se requieren personas que aporten valor a su equipo de trabajo. A partir del estudio de los datos, las plataformas digitales y reclutadores han realizado esfuerzos para crear contenido de aprendizaje el cual instruye a las personas en el campo de la ciencia de datos, sin embargo, en el camino para adquirir las habilidades y conocimientos necesarios no es tan simple como tomar algún curso o certificación. Por otra parte, no es suficiente conocer las herramientas y algoritmos para la manipulación de datos; es necesario contar con el conocimiento del contexto de los datos, es decir, reconocer la narrativa que pueden brindar los datos sobre cierto tema. Las técnicas que se emplean requieren que el profesional atraviese una curva de aprendizaje de extensión cambiante, según las aptitudes o perfil. Existen distintas herramientas tecnológicas que van desde lenguajes de programación para extraer, cargar o transformar datos, gestores de bases de datos y prácticamente cualquier sistema o producto de software necesario. Se da por entendido que, para el perfil de científico de datos no solo se requiere ser experto en un área, debe conocer ampliamente varias herramientas tales como gestores de bases de datos, sistemas operativos, redes de comunicación y todo aquello que conlleve a la gestión de datos y su análisis.

Para obtener el mayor conocimiento que proporcionan los datos, es importante tener en cuenta que es lo que se busca, realizar preguntas que previamente hayan sido analizadas cuidadosamente y que estas generen información. Se debe comenzar con interrogantes que proporcionen respuestas claras. Por esto, las compañías invierten capital en busca de estas, ya que generan grandes cantidades de dinero. Desde el punto de vista empresarial el conocer el valor de los datos, aproxima una mirada al futuro y a la correcta toma de decisiones.

Otro de los ámbitos importantes que se debe mencionar, es el entorno educativo. Los datos brindan la capacidad de conocer temas inexplorados o predecir comportamientos que tendrán

impacto en pequeñas y grandes áreas. Un ejemplo claro de esta proyección, es el clima. Cada día cientos de sensores en todo el mundo capturan información que se almacena en forma de datos. Se analizan variables que expertos en el área leen para predecir el clima con alta exactitud. Este proceso requiere varias etapas antes de entregar una decisión. Los datos van transformándose, transportándose y extrayéndose en fragmentos para su interpretación (Hernández, 2018) .

Es importante mencionar que, además de existir obstáculos sobre diversidad de herramientas se tiene que tomar en cuenta la capacidad de cómputo para leer datos. En años pasados el tema de velocidad de transferencia o capacidad de almacenamiento provocó que sistemas no pudieran ser utilizados, ya que no había tal tecnología para realizar los procedimientos que estos proponían. El crecimiento de los datos requirió que los sistemas evolucionaran, hoy en día ya no es factible contenerlos en un solo lugar. Debido a esto, comenzaron a surgir tecnologías de almacenamiento en la nube. Dicha tecnología permite almacenamiento de información desde cualquier lugar donde se tiene un punto de conexión, por lo que no es necesario trabajar en el mismo lugar donde se encuentran los datos. Comenzaron a sobresalir protocolos de comunicación de soporte para el intercambio de datos, además de lenguajes universales de comunicación, en otras palabras, se tiende a la interoperabilidad. En consecuencia, la heterogeneidad de los sistemas ya no es un obstáculo, es decir, los sistemas pueden mantener su lenguaje y forma de realizar sus procesos intactos, en caso de ser requerido establecer comunicación con dicho sistema, se utiliza un software conector que realice dicha tarea.

El presente trabajo tiene como parte de su sujeto de estudio un fenómeno que sigue impactando a nivel mundial. En 2020 se decretó estado de pandemia global emitido por la Organización Mundial de la Salud (OMS, o sus siglas en ingles WHO), debido al virus SARS-CoV-2, el cual pertenece a la familia de los coronavirus, aunque la identificación de dicho virus fue realizada en 2019 (NIH, 2021).

En efecto de materia de estudio, esto representó un intercambio masivo de información respecto a descripción de síntomas, tiempos de contagio, entre otras variables. El contexto de la información pertenece a la epidemiología. Por lo que, cuando se obtuvo más información acerca de la enfermedad, los expertos en datos comenzaron a estudiar los comportamientos por medio de las variables epidemiológicas, con la finalidad de concebir predicciones en las curvas de contagio y reconocer patrones hacia la población.

El gobierno federal de México optó por campañas de cuidado hacia la población en general, tales como sana distancia, un programa el cual enfatiza las precauciones que se deben seguir ante la crisis COVID-19. Se utilizó el semáforo epidemiológico, el cual da una pauta para la movilidad económica y social en los estados. El control del color está sujeto a el índice de contagio que existe en el estado (entidad federativa) y es controlado por el representante de la secretaría de salud de cada estado. Los cambios de estatus del semáforo epidemiológico son transmitidos a la población los días viernes conforme a la semana epidemiológica. Por otra parte, el sector salud comenzó a recopilar información de los pacientes, bajo la idea de contar con una base de datos con registros de variables epidemiológicas de carácter relevante.

Los datos han sido proporcionados por el gobierno mexicano por medio archivos separados por comas desde el 12 de abril del 2020 por medio del portal de datos abiertos dirección general de epidemiología bajo el esquema de datos abiertos en Latinoamérica (GobMX-a, 2021). Estos datos han sido tomados de forma oficial por su proveniencia gubernamental, pero resulta necesario pensar en escenarios donde la fuente no sea expuesta o carezca de credibilidad. Lo anterior representa una oportunidad en el área de ciencia de datos y técnicas de gestión de datos.

Si bien existen portales que concentran la información de forma resumida donde resaltan las variables más importantes, estas carecen de documentación de la metodología aplicada y no permiten la descarga total de la información sino segmentos de datos previamente construidos por terceros. Lo anterior no significa que carezca de importancia, sin embargo, la investigación sobre el tema debe ser construida partiendo de la totalidad de información existente. En consecuencia, resulta complicado para el investigador sin perfil de computación el recopilar la información acerca del COVID-19 México.

Ya se ha descrito la importancia de los datos abiertos, además de los problemas que conlleva realizar su estudio. Se han definido los conceptos de ciencias de datos, estadística, variables epidemiológicas y los factores que se requieren para elaborar un correcto análisis. Contar con la información sin duda es prioridad para el investigador y estimula el desarrollo de la investigación, sin embargo, no se está exento de problemas. En la siguiente sección se describe la problemática a tratar y como se pretende solucionar.

1.1 **Definición del problema**

El recopilar información de forma que un sector en específico lo requiera, supone gran dificultad al momento de consultar o resguardar dicha información en formatos adecuados con el fin de ser estudiados. Los datos o set de datos son segmentos seleccionados para brindar información con un contexto predeterminado (DCC, 2014). Estos se encuentran originalmente en su forma almacenada la cual puede significar que tienen que ser transformados para su estudio o análisis. Convertir dichos datos en un nuevo elemento de estudio, no es tarea fácil para cualquiera, la cantidad de datos puede incrementar en varios terabytes o incluso llegar a tamaño de petabyte por lo que, además de la dificultad de extraer información, no es fácil de obtener con cualquier capacidad de cómputo. Es habitual que se consulte a un experto en la materia o alguien con experiencia en extracción, carga y transformación de datos (ETL) y gestores de bases de datos. Desde el punto de vista del rol que se dedica a recabar dicha información, existen varias limitantes con las que lidiar, por ejemplo: volumen, veracidad, velocidad y variedad (Elgendy-Nada, 2014). Estas características también son conocidas con las cuatro 'v' del Big data, pero en realidad no se limitan a este concepto solamente. Estas también pertenecen a características de ETL.

Actualmente, las dependencias y gobiernos del mundo han adoptado el término de open data (datos abiertos), el cual consiste en brindar transparencia de información en determinados temas que conciernen en mayor medida a grupos de analistas e investigadores (D'Agostino, 2020). Estos requieren dicha información para la toma de decisiones o pronósticos de comportamientos específicos. Los temas suelen ser fenómenos que interactúan directamente con sociedades o impactos mundiales, por ejemplo: datos geográficos, climáticos, educativos, poblacionales, epidemiológicos, etc. Además de contar con los datos que serán procesados o manipulados, es necesario aplicar métodos o modelos matemáticos para el análisis de dicha información. Partiendo de esto, los expertos tienen diferencia de opinión con los límites de la ciencia de datos y otras ciencias, como la estadística. Por una parte, está la ciencia que se encarga de los fenómenos aleatorios siendo a su vez parte de las matemáticas y la Ciencia de datos que más que una ciencia, se le es atribuye la definición de técnica para implementar los distintos métodos a datos con el fin de encontrar relación entre ellos o patrones que nos lleven a la toma de decisiones (Brodie, 2019). La disciplina emergente de la Ciencia de datos ha revolucionado nuestro mundo y cambiado nuestra forma de vivir drásticamente (Daniel Lemus-Delgado, 2020).

La Ciencia de datos al ser un punto de encuentro entre varias disciplinas ha incrementado su uso de tal manera se han simplificado los tópicos de estudio y se ha apoyado de herramientas de software que contienen algoritmos de fácil aplicación, dejando de lado la curva de aprendizaje que recae en el conocimiento de los algoritmos matemáticos. En la mayoría de los casos la ciencia de datos ha sido implementada como estudio junto con los perfiles computacionales o investigadores en torno al análisis matemático permitiendo que el profesionista adquiera conocimientos y habilidades para resolver problemas en torno a los datos.

Cuando se dispone de datos se busca extraer información, sin embargo, existen varios obstáculos. Como se mencionó, el perfil resulta de suma importancia, este es la diferencia en conocer previamente herramientas tecnológicas que facilitan el trabajo sobre los datos. También existe complejidad en la distintas fuentes de bases de datos que van desde el diseño y la arquitectura con las que son creadas. Se requiere conocer protocolos de comunicación para el intercambio de datos. Lo anterior refleja una base sólida de conocimientos previos a la implementación de cualquier análisis científico.

Como caso de estudio, la base de datos que el gobierno de México expone para dar seguimiento de COVID-19 en México, se actualizan diariamente o cada determinado tiempo, es decir, el análisis que se pudiera aplicar en un determinado tiempo varía con los datos que saldrán en otro momento (México, 2021). Desde este punto de vista es posible analizar los datos históricos y evaluar su comportamiento o también, dar un seguimiento diario con cantidades actualizadas según las variables consideradas.

La Secretaria de salud del Gobierno de México (GobMx) pone a disposición los diccionarios de los sets de datos de COVID-19 en México, estos cambian según las variables que se hayan agregado con el tiempo al set de datos. Se ha separado la información de las variables y se han creado nuevas variables que no se contemplaban en el contexto de la investigación. Este tipo de comportamientos hace que sea una tarea compleja para los sistemas de información, ya que requiere que cambien su estructura constantemente. Al momento de requerir la información esta solo se encuentra en un solo formato y con los datos actualizados por día. El documento contiene el total de los datos existentes y carece de variables que expresen los cambios que han atravesado las variables en el tiempo.

El problema radica, en la estructura y metodología explicativa por parte de la entidad que entrega los datos. Carecer de dichos elementos provoca inestabilidad en la gestión de datos. Al

no tener una metodología clara, supone que los datos deben ser tomados como auténticos y que pertenecen al día en que el documento fue expuesto. Pasan por alto perfiles de investigación que están interesados en los procesos de exploración y preparación de datos. Dejan en duda las complicaciones técnicas que pueden existir y la posibilidad de interpretar en costo, el uso de herramientas que permitan mayor computo de datos. También, se ven limitados los perfiles profesionales al no poder acceder a los datos por segmentos de estudio o en distintos formatos.

Ahora que se exponen las bases de complejidad del problema, es posible generar objetivos con el fin de utilizar la tecnología en la resolución del problema. Además de definir por lo menos de forma general las necesidades partiendo de proto-usuarios.

1.2 **Justificación**

Una vez que ha sido descrito el problema y las limitaciones que se encuentran dentro del contexto de investigación referente a los grupos de investigadores que requieren dicha información de COVID-19 en México, es importante mencionar la solución que propone el presente trabajo de investigación.

Es necesario crear un sistema de software capaz de procesar los archivos de datos que expone la federación con tema COVID-19 en México y a su vez, brindar el correcto tratamiento para el almacenamiento de la información. Esto tiene como finalidad brindar acceso con información correcta y concisa en forma de servicio, así, los investigadores que no cuentan con el perfil de computación para manipular la cantidad o forma de datos, pueden apoyarse con alguien que tiene conocimientos de gestores de bases de datos y desarrollo de software para lograr el objetivo deseado.

Cabe mencionar que para lograr lo anterior es necesario definir correctamente la metodología a seguir y proporcionar solamente la interfaz la cual, servirá de puente para el acceso a la información. Los datos resguardados serán aquellos publicados solamente por la entidad federativa ya que son considerados como oficiales, en este caso la Secretaría de salud. Estos serán considerados como datos actuales (actualización diaria automática, datos históricos y datos por fecha desde el comienzo de su exposición).

Como resultado se obtendrá la gestión de datos COVID-19 México utilizando como herramienta tópicos selectos en el área de las bases de datos y obtención de datos.

1.3 **Preguntas de investigación**

Las preguntas que se presentan a continuación pretenden dar sentido a los resultados del trabajo de investigación:

1. ¿Cuáles plataformas existen para la consulta de datos abiertos con tema de COVID-19 en México? ¿Hacia qué tema están enfocados? ¿Qué estructuras utilizan? (servicios, manejadores de bases de datos, archivos)
2. ¿La información que proporcionan las plataformas de información es correcta? (integridad y estandarización) ¿Cuál es la mejor opción para exponer los datos epidemiológicos con respecto a COVID-19 en México? ¿De qué forma apoyan al investigador sin perfil computacional?

3. ¿Cuál esquema resulta ser el más adecuado para el almacenamiento de datos heterogéneos? ¿Cuáles son sus características?

1.4 **Objetivos**

1.4.1 **Objetivo general**

Gestión y obtención de datasets de registros COVID-19 México, eliminando procesos intermedios a través de plugin Sistema núcleo COVID-19 México en periodo de abril del 2020 hasta mayo del 2021.

1.4.2 **Objetivos específicos**

1. Desarrollar un sistema de software basado en plugin que sea capaz de consultar, extraer, transformar y almacenar el archivo de datos COVID-19 en México diariamente.
2. Formalizar la estructura de bases de datos para el correcto depósito de datos. Con el fin de agilizar las consultas a gran escala.
3. Desarrollar un servicio el cual exponga la información de COVID-19 en México tal como se da a conocer por las entidades de salud en México. Está a su vez con la capacidad de crecer según las consultas requeridas.

1.5 **Estructura del documento**

- Como primera aparición se encuentra el capítulo 1 de**marco teórico**, aquí se explican los temas que toman relevancia en el trabajo de investigación y se hace mención de citas bibliográficas de relevancia.
- Se presenta el capítulo 2 de **metodología** del trabajo. Se describe la forma en que se llevó a cabo la investigación de forma ordenada.
- El capítulo 3 de **análisis y diseño del desarrollo del software**, describe los elementos utilizados para la construcción de las herramientas que apoyan a la investigación para alcanzar el objetivo.
- El capítulo 4 de **validaciones** presenta los casos de uso que demuestran el funcionamiento de la herramienta de software y como se alcanzan los objetivos a través de estas.
- Se presentan **discusiones** sobre la investigación en el capítulo 5.
- El capítulo 6 presenta las **conclusiones y trabajo futuro**.
- El documento también presenta al comienzo los acrónimos utilizados y al final un glosario que contiene las definiciones a conceptos dentro del ámbito de la investigación.

Capítulo 2. Marco Teórico

El estudio de los datos y el comportamiento de los mismos han buscado comprenderse con distintas herramientas y teorías. No obstante, para comprender cada una de ellas, es importante definir conceptos clave que son utilizados cuando se trata de bases de datos y gestión de datos. Además, se hace mención de las distintas fuentes de datos y sus características. Lo anterior, pone en contexto el trabajo de investigación y describe los elementos que participan en el proceso para la gestión de datos.

2.1 Datos abiertos en sector salud nacional

En esta investigación se utilizan datos expuestos por la Secretaría de Salud de México. Se debe conocer los recursos existentes que pertenezcan al esquema de datos abiertos.

Muchos de los aspectos sociales y de gobernanza del intercambio de información han sido pioneros en el área del software de código abierto. Desde hace años los sistemas de información pública han sido de gran ayuda para el crecimiento de la comunidad científica, ingeniería y desarrollo (Open Source). Existen plataformas con miles de usuarios que disponen de fuentes de datos y desarrollos en progreso, GIT es un ejemplo claro de esto. Estos tienen como finalidad brindar confianza al ciudadano y hacerlos partícipes en la opinión contextual de los mismos. los gobiernos tienen como responsabilidad el dar confidencialidad y calidad a los datos abiertos, por lo mismo existen estrategias para dar legitimidad de los datos. Sin embargo, las entidades gubernamentales se enfrentan a desafíos tales como: seguridad, velocidad, diferencias entre oferta y demanda de los mismos e insuficiencia de valor hacia los datos. La definición de la carta internacional de datos abiertos (2018) afirma, “Los datos abiertos son datos digitales que son puestos a disposición con las características técnicas y jurídicas necesarias para que puedan ser usados, reutilizados y redistribuidos libremente por cualquier persona, en cualquier momento y en cualquier lugar”.

Los objetivos que se persiguen es el incremento de la transparencia a través de la mayor publicación posible, por lo tanto, es de suma importancia la integridad de los mismos. El contribuir con actores externos, definir políticas públicas basadas en evidencia. A su vez las instituciones crean estrategias como la promoción de crecimiento sostenible e inclusivo, facilitar los procesos de licitación, oportunidades económicas, participación en debates políticos y cívicos.

2.2 **Fuentes de datos**

El trabajo de investigación es realizado con una sola fuente de datos, a continuación, se describe su concepto y los distintas fuentes de datos.

Por fuente de información se entiende cualquier instrumento o, en un sentido más amplio, recurso, que nos pueda servir para satisfacer una necesidad informativa.
Tienen como objetivo el facilitar la localización e identificación de datos. Se dividen en primarias y secundarias. Las primarias, son aquellas donde la información es nueva, es decir se encuentran en artículos o manuscritos. En cambio, la secundaria, es aquella que se observa como referencia en los documentos, no se encuentra en su estado original ni completo.
Es importante mencionar que se buscan fuentes de información confiables, estas deben ser claras y contener la fuente de credibilidad. También son válidas aquellas fuentes a las cuáles se les ha aplicado interpretaciones o razonamiento, pero estas deben estar sustentadas con bases firmes. Las fuentes que contienen distintas perspectivas pueden resultar confiables, esto se debe a la credibilidad del responsable. El otro tipo de fuente confiable es aquella que es legitimada por terceras personas, aquí recaen los documentos de investigación ya que son evaluados por expertos en el área del cual se escribió la fuente de información. A su vez, existen distintos tipos de fuentes de información las cuales son utilizadas para diferentes puntos de reflexión.

2.2.1 **Datos puntuales**

La información es clasificable en categorías no numéricas, contienen variables que le dan identidad al valor que contiene supuesto por escalas ordinales. Contiene información como sexo, edades. A su vez como puntual también están aquellas que recaen en la escala ordinal la cual contiene números de promedios, pesos, etc.

2.2.2 **Datos Espaciales**

Las fuentes de datos espaciales son aquellas que analizan y comparten información geográficamente referenciada, se les conoce como SIG. Estas están estructuradas por su ubicación, dimensión y forma. También existen datos no espaciales dentro del mismo concepto, estos funcionan como diccionarios y habitualmente se encuentran en tablas, también toman el nombre de datos descriptivos. Para complementar los datos espaciales, existen las capas geográficas las cuales son características del sitio, estas se pueden modelar y suelen ser

varias y divisibles por tema de investigación. Así mismo, la entidad se le conoce como objeto o concepto al cual rodea la información de datos espaciales. Otro factor importante es la representación geométrica, está conformado por la representación digital del componente espacial, pueden ser líneas, figuras o puntos. Y por último se tiene el modelo de datos, son un conjunto de herramientas para describir los datos, relaciones y límites.

2.2.3 **Datos de Series de tiempo**

Por definición, una serie temporal es una sucesión de observaciones de una variable realizadas a intervalos regulares de tiempo. Existen diversos tipos de series temporales, pero cada una de ellas se basa en la forma en que se mide el tiempo, es importante destacar que los datos deben ser homogéneos, es decir se debe mantener la medición de la magnitud del objeto de estudio.

2.2.4 **Fuentes Históricas**

Estas fuentes comprenden la información en base a testimonios u objetos que se referencian de un hecho ocurrido, son recopiladas por historiadores que se encargan de corroborar la veracidad de la fuente. Al igual que la definición de fuente de información estas se clasifican en primarias y secundarias. Los temas que abarcan suelen ser sociales o personas, actividades, sucesos y opiniones.

2.2.5 **Fuentes en Tiempo real**

Son aquellas fuentes que proporcionan información al momento de ser consultadas, además de la edición de la misma también en tiempo real. Estas fuentes tienen como objetivo el proporcionar información al investigador o lector para asegurar que los datos siguen patrones previamente establecidos o descubrimiento de nuevos.

2.2.6 **Fuentes de datos epidemiológicas nacionales**

Conforme al decreto publicado en el diario oficial de la federación el 20 de febrero del 2015, la dirección general de epidemiología optó por dar a disposición los datos abiertos a la población general, anuarios estadísticos de morbilidad y mortalidad 2015-2017. Además de la información con los casos asociados al COVID-19 (salud, 2021).

Los datos epidemiológicos tienen como principio el seguir las guías éticas en el ámbito de la investigación, protección del derecho a la intimidad de los datos sobre la salud de las personas.

El grupo de trabajo sobre la confidencialidad y protección de los datos de epidemiología describe, “El retroceso ocasiona una merma en los derechos de los individuos, tanto desde el punto de vista individual, valorando la salud como derecho subjetivo de todos los individuos en lo que respecta a un tratamiento individualizado, como desde la concepción de la salud como un bien social, un valor que pertenece a toda la sociedad y en el que los procesos de investigación tienen como objetivo la protección de aquélla y la mejora de la calidad de vida”. Esto a su vez trae como consecuencia conflictos, ya que tiene que existir un consentimiento informado, también se tiene que brindar el derecho a cancelar dicha información, además los miembros pueden establecer excepciones de información al momento de ser publicados como interés de investigación.

Para que los miembros de registro sean válidos deben poseer información de carácter de identificación personal. La figura 1 describe un proceso de clasificación de muestras aleatorias y segmentación de variables epidemiológicas.

Al contar con información confidencial orientada a un caso específico de enfermedades, está sirve de base para el problema científico y se espera que tenga descripción hipotética, para someterse a pruebas. La hipótesis es seccionada por fragmentos más pequeños que son evaluados como variables susceptibles que puedan ser medidas, si al aplicar procedimientos empíricos la hipótesis prevalece esta es considerada como verdadera. En la epidemiología se lleva a cabo este proceso de medición y obtención de variables al igual que la mayoría de las ciencias.

El principal objetivo de la investigación epidemiológica es describir la distribución de las enfermedades en la población, también el caracterizar y clasificar los objetos de contagio (Fajardo-Gutiérrez, 2017). El generar concientización en la población es una responsabilidad fundamental para esta rama de la medicina.

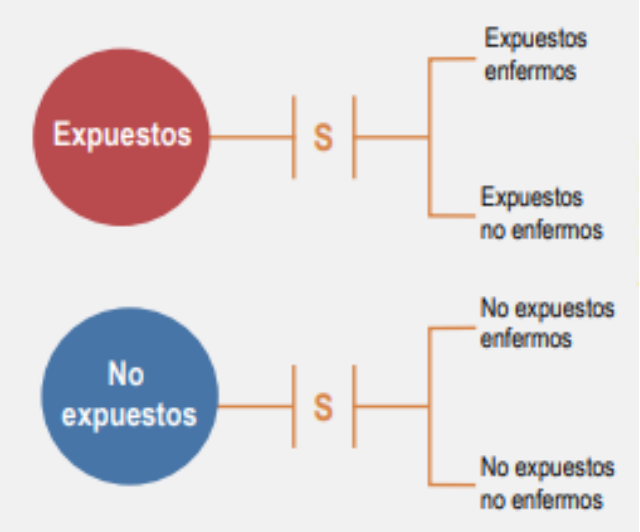


Figura 1 Proceso de muestras en población.

2.3 **Variables epidemiológicas**

Moreno y López-Moreno definen que “las variables tienen como función proporcionar información asequible para descomponer la hipótesis planteada en sus elementos más simples” (2000). Estas se consideran como las características de eventos que transcurren a través del tiempo, determinan factores de cambio por lo que toma diferentes valores y deben ser estudiadas dentro de un marco de problema e hipótesis aplicada.

En la epidemiología las variables permiten elaborar modelos descriptivos, explicativos y predictivos sobre el marco teórico del sector salud, generalmente son visualizadas como tablas categóricas para clasificar eventos. El concepto de medición también es explicado por Moreno (2000) y López-Moreno (2000) como “la asignación de un número o calificación a una propiedad específica de un individuo, población o evento usando ciertas reglas”. Lo anterior también puede ser visto como abstracción de características, para medir es necesario tomar una entidad teórica, transformarla en escala conceptual y posteriormente en escala operativa. Las escalas son clasificadas como cualitativas y cuantitativas, es necesario que las características pertenezcan a una sola categoría.

La escala nominal consiste en clasificar observaciones en categorías de presencia o ausencia en palabras más simples existe o no en el marco de investigación. Por lo general estas son representadas con números, estas no tienen valor como tal o peso, solo difieren las características de las demás.

Escala ordinal, clasifica por categoría según el grado de observación de los objetos, eventos o características. Es decir, una vez asignando que la medición existe esta debe poseer un peso distinto a las demás.

Escala de intervalos, ordena las observaciones por categoría de atributo y mide la magnitud de la distancia que existe entre las categorías, esta no proporciona información del atributo como tal o de categoría.

Escala de razón, esta existe en la razón de dos números y su relación real existente entre características, en otras palabras, se puede dar por concluida a partir de valores preexistentes para confirmar el valor medido.

Una vez propuestas las mediciones que se requieren para el estudio de datos en epidemiología, se puede tomar partido en cálculos de proporciones y tasas, estas se traducen en términos probabilísticos. A la frecuencia en que ocurre un evento en relación de un objeto de estudio se le llama proporción.

2.4 **Dependencias de información**

2.4.1 **Organización Mundial de la Salud**

Es el organismo de la Organización de las Naciones Unidas (*ONU*), el cual gestiona procedimientos y políticas entorno a la salud a nivel mundial. Está designada por la asamblea mundial de la salud, la cual tienen como reunión mayormente en mayo de todos los años, toman en cuenta temas además de la salud como finanzas, presupuestos para programas y hace efectiva la toma de decisiones.

2.4.2 **Secretaría de salud del Gobierno de México**

Según el sitio Web www.gob.mx, la definición de Secretaria de salud en México es: “Dependencia del Poder Ejecutivo que se encarga primordialmente de la prevención de enfermedades y promoción de la salud” (2021), establece las políticas para que cada persona en territorio mexicano tenga acceso a la protección en tema de salud. Actualmente opera conforme al decreto publicado en el diario oficial de la federación el 20 de febrero del 2015,

este establece la regulación con tema de datos abiertos. La dirección general de epidemiología pone a disposición a la población la información contenida en temas de salud y agregó en 2020 los primeros datos COVID-19 en México.

La estructura que se propuso para la divulgación de datos fue un archivo diario con contenido de campos de observación entre ellos identificadores únicos anónimos para la asignación de casos y variables de categorización. Además, cuenta con un diccionario de datos que brindan la información contenida en el archivo principal, también ponen a disposición el histórico COVID-19.

2.5 **La epidemia COVID-19**

El 31 de diciembre del 2019, China informó a la organización mundial de salud la existencia de neumonía viral en la ciudad de Wuhan en la provincia de Hubei. Los días posteriores, médicos en Singapur alertaron a los alrededores de un viajero que provenía de Wuhan el cual había sido identificado con neumonía, para el 20 de enero del 2020 se había levantado una campaña de no propagación para un coronavirus. Para el 19 de febrero del 2020 existían confirmados 84 casos de la enfermedad COVID-19 derivada del nuevo coronavirus SARS-CoV-2, dichos exámenes fueron expuestos por el laboratorio ORF1ab de genética. Con un total de 2593 contactos del virus, los primeros síntomas se presentaron en media la primera semana desde el contacto, estos desarrollaron neumonía severa y otros síntomas.
En enero de 2021 existen más de 17, 500, 000 casos positivos que han reportado más de 213 países en todo el mundo. La OMS declaró emergencia de salud el 30 de enero del 2020.

2.5.1 **Panorama Mundial**

La enfermedad COVID-19 ha impactado al mundo en tema de salud, no obstante, ha traído cambios significativos en el sector médico, político, social y tecnológico (Brown, 2021). “El rápido cambio a trabajar en remoto y la comunicación en línea ha redefinido la forma en que colaboramos”. A pesar de ser un tema de mortalidad mundial, los cambios que ha sufrido la población mundial han sido contrastantes, los trabajos tuvieron que evolucionar sus prácticas para no dimitir en sus operaciones, la educación se apoya en las plataformas de educación digital ahora más que nunca. Por otra parte, los expertos en tecnología tomaron más presencia de la que se tenía, el campo de la ciencia de datos juega un papel sumamente importante, ya

que ha servido de apoyo para el análisis y predicción del comportamiento de la presente pandemia. Múltiples organizaciones optaron por dar un empuje al campo de los datos y transformarlos en toma de decisiones, el objetivo es presentar el panorama mundial del que está sucediendo en tiempo presente y cómo nos afectará en el futuro.

2.5.2 **Panorama Nacional**

El primer caso de COVID-19 fue detectado el 27 de febrero del 2020 en México. La curva de contagio incrementó exponencialmente 30 días después, los primeros patrones en descubrirse fueron la edad promedio de contagio y los padecimientos en historia clínica de los contagiados. La media de 46 años, una mayor probabilidad en hombres (58.18%) y la comorbilidad de hipertensión fueron los primeros signos de preocupación para el país. Posterior a esto se decidió crear una estructura de contagio y enfermedad la cual describe que los casos positivos tienen que resultar de una prueba de reacción en cadena de la polimerasa de transcripción inversa *(RT -PCR*). Los casos sospechosos son catalogados como aquellos que cumplen por lo menos dos síntomas entre los principales, fiebre y prueba de existencia de neumonía. Por último, los asintomáticos es categorizado como aquel individuo que presenta temperatura corporal normal y molestias menores, pero es positivo a la prueba *RT -PCR.* En el ámbito educativo, la SEP (Secretaría de Educación Pública) decidió adelantar el periodo de vacaciones de semana santa extendiéndolo un mes en todo el país.
El 24 de marzo del 2020 el gobierno federal decretó la segunda etapa de la pandemia debido a la velocidad y gran número de contagios, en esta etapa se suspenden las principales actividades económicas y regulan reuniones con alto índice de participación. Gobierno federal inicia su campaña de “Quédate en casa”.

El 21 de abril del 2020 se inició la tercera etapa de la pandemia en México, se dio el aviso de suspensión de actividades no esenciales en sectores público, privado y social.

2.6 **Bases de datos heterogéneas**

Hoy en día es de uso común para las entidades privadas y públicas, utilizar diferentes tipos de bases de datos para lograr sus objetivos en el contexto de manejo de datos operacionales.
Estos sistemas son diseñados para el uso aislado y funcionan de manera autónoma. Aun cuando no se tenga conocimiento de configuraciones de otras bases de datos, estas pueden

estar diseñadas para trabajar de forma cooperativa, es necesario que exista una interoperabilidad entre las mismas (Castro, 2014). Dicho de otra manera, no tiene relevancia que un sistema de base de datos haya sido diseñado con distintos modelos, esquemas y hardware, sino que la salida de información almacenada debe ser capaz de comunicarse con otro entorno de base de datos. Centralizar la heterogeneidad de las bases de datos puede resultar complicado, ya que se espera el funcionamiento distribuido. Cada uno de los gestores de bases de datos tiene exigencias y requerimientos propios, normalmente las organizaciones utilizan bases de datos gratuitas o las que se acercan a su modelo de operación. Una vez que un gestor de base de datos se acopla correctamente al modelo organizacional, existe alta probabilidad que la información fluya de distintas formas para cada departamento. Se aplican métodos para lograr que las bases de datos persistan en los organismos sin tener que modificar el modelo de negocio. El proceso de separar o dividir la información en fragmentos se le llama fragmentación, se clasifica en horizontal y vertical. La fragmentación horizontal divide las relaciones de bases datos en uno o más fragmentos para su interpretación y la vertical divide la relación de los esquemas. Existe también el proceso de replicación, esta es una copia que existe en dos lugares diferentes, estas pueden entregar los mismos resultados a los clientes.

La figura 2 describe los distintos tipos de técnicas de bases de datos a través del tiempo. El sistema de archivos existe aún sin la tecnología, este es utilizado desde el resguardo de la información en archiveros o bitácoras hasta documentos digitales almacenados en equipos de cómputo. A medida que la tecnología crece, se crean nuevos modelos de bases de datos. Estos ofrecen distribución en distintos puntos de información y mayor capacidad de almacenamiento, por lo que fue necesario crear nuevas técnicas de resguardo.

En la era de la información se busca unificar los datos, actualmente se realizan exploraciones globales que buscan resolver los problemas de la heterogeneidad. Se han desarrollado diversos tipos de software que permiten el acceso a distintas bases de datos y también se han creado protocolos de interoperabilidad que sean de uso general. Sin embargo, el alto costo de transformar aplicaciones o implementar protocolos ha sido el principal obstáculo.

En este trabajo de investigación se utiliza una base de datos heterogénea con variables epidemiológicas que son expuestas por el gobierno de México de manera oficial.

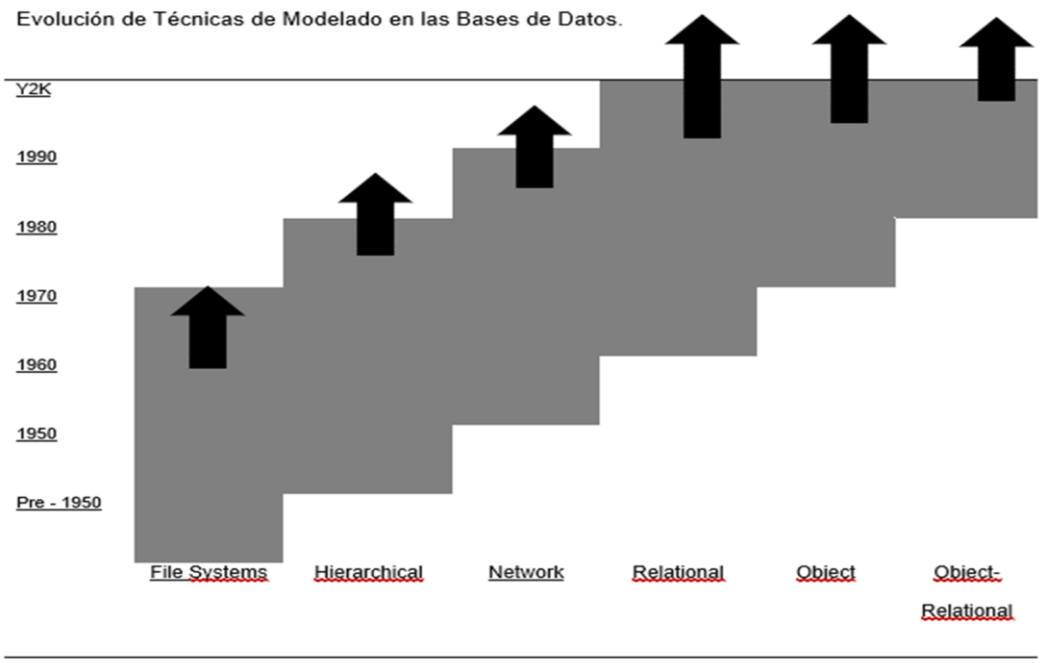


Figura 2. Evolución de bases de datos.

2.6.1 **Bases de datos relacionales**

Para el almacenamiento de datos de esta investigación, se elegirá un gestor de bases de datos relacionales. Es importante describir los paradigmas de las bases de datos y las características que los conforman. Al conocer el funcionamiento o el enfoque de las distintas bases de datos, es posible tomar decisión de cual utilizar, buscando cual se adapta más a las necesidades. Además de la definición es importante mencionar los métodos que se utilizan sobre los datos.

Como definición en su forma básica se tiene que, las bases de datos son una colección de datos, que suelen estar organizadas y estas expresan información. Por otra parte, un modelo de base de datos es el concepto a diferencia del almacenamiento físico de los datos y este es utilizado para elaborar tablas en bases de datos (Anselma, 2016).

La evolución del modelo de base datos ha evolucionado de forma progresiva escalando de forma vertical junto con la tecnología, las primeras definiciones del concepto hacían alusión a el almacenamiento físico por sistema de archivos, posteriormente estos han escalado a virtualizaciones e internet.

Es de suma importancia que una vez comprendido el concepto de bases de datos se especifique de qué forma serán almacenados y consumidos los datos, por muchos años han

existido dos grandes formas de almacenar los datos o interpretarlos, relacional y no relacional. Cuando se habla de base de datos relacional se toma en cuenta el modelo que se utiliza, por el mismo nombre se es seleccionado el modelo relacional, aquel que aporta valor de planificación, este es capaz de interconectarse entre sí, relacionarse entre datos de una tabla con otra y ejercer operaciones de vital importancia (Inmon, 1999). Dicho lo anterior, el modelo representa el paradigma de relación. Los modelos relacionales tienen características que aportan valor al comportamiento de los mismos.

La normalización es la transformación de las vistas de usuario complejas y del almacén de datos en juego de estructuras de datos más pequeñas y estables. Además de ser más simples y estables, las estructuras de datos son más fáciles de mantener que otras estructuras de datos (Gutiérrez, 2006).

El objetivo principal de normalizar es retirar la redundancia que pudiera existir en los datos, para esto se apoya de tres normas o reglas que son descritas en la tabla 1.

La primera forma describe que las tablas deben contener datos que no sean divisibles y grupos de valores repetidos (1FN).

La segunda forma, toma en cuenta que los datos deben depender de una clave única en la tabla, en caso de que existan dos claves esta entidad debe ser separada por otra que relacione a la primera (2FN).

En cuanto a la tercera forma normal, describe que no deben existir dependencias transitivas en las columnas de una tabla (3FN).

La finalidad de las bases de datos relacionales es proporcionar acceso a todos los puntos de vista de los datos almacenados. Aunque en los últimos años ha tenido gran impacto su contraparte, sigue siendo un esquema estable cuando se trata de diseño de bases de datos. Siempre se ha buscado relacionar los datos o buscar patrones por lo que resulta lógico utilizar un modelo relacional de bases de datos.

Tabla 1. Puntos opcionales al tomar en cuenta para realizar la normalización en bases de datos.

Identificador	Descripción
1	Composición de tablas o relación entre si
2	Diferencia de nombre entre las tablas o relaciones
3	Relaciones entre tablas se da por medio de claves primarias o foráneas
4	Las claves principales son la clave principal de un registro de tabla
5	Las claves foráneas se asignan a tablas o relaciones hijas
6	Las relaciones deben ser capaces de normalizarse por lo menos en una ocasión.
7	Los valores de las tablas deben ser del mismo tipo de dato en una misma columna
8	La información puede ser recuperada o almacenada por medio de sentencias (consultas).

2.6.2 **Bases de datos no relacionales**

Las bases de datos no relacionales tienen como finalidad la gestión de grandes volúmenes de datos y estas carecen de estructuras definidas. A diferencia de las bases de datos relacionales no utilizan tablas o registros. A continuación, se define el concepto de base de datos no relacional y se explica su funcionamiento.

El paradigma de las bases de datos no relacionales nace de los diversos requerimientos de las empresas modernas las cuales buscan rapidez, librarse de problemas de volumen entorno a los datos, también se conoce como NoSQL o bases de datos de la era del internet. Al no requerir esquemas fijos, la velocidad de intercambio de información incrementa en gran medida, tienen la característica que su escalabilidad ocurre de forma horizontal por lo que es definido que carecen de RDBMS y no almacenan los datos en tablas ni contienen esquemas, esto ocurre porque evolucionan constantemente. La figura 3, representa un diagrama de documento el cual es interpretado posteriormente por distintas entidades que le dan forma a la información almacenada.

Sadalge (2014) encierra las bases de datos no relacionales en 4 categorías, bases de datos clave valor (Key-Value Database), estas son almacenadas en desde una perspectiva de API (Application Programming Interface***)***, se agrega un valor para la clave o se elimina de la misma

forma. Bases de datos de documentos (Documents databases), estos son almacenados y consultados de en lenguajes de marcado como son XML, JSON, BSON. Son capaces de auto describirse y suelen almacenar datos de mapas, colecciones o valores escalares, también son manipulados por llaves primarias para su sencilla manipulación. Suele utilizarse MongoDB, CouchDB para esta categoría. Base de datos de almacenamiento de familias por columna (column family stores), los datos son almacenados en forma de renglones de columna de familia los cuales contienen columnas con claves que relaciona los datos. Comúnmente es utilizado Cassandra y Amazon DynamoDB. Bases de datos en gráficas (graph databases), esta permite almacenar los datos en forma de entidades las cuales son llamadas nodos, estos contienen propiedades y relaciones que generan patrones, se utilizan para calcular los tiempos en ejecución u operaciones de tiempo.

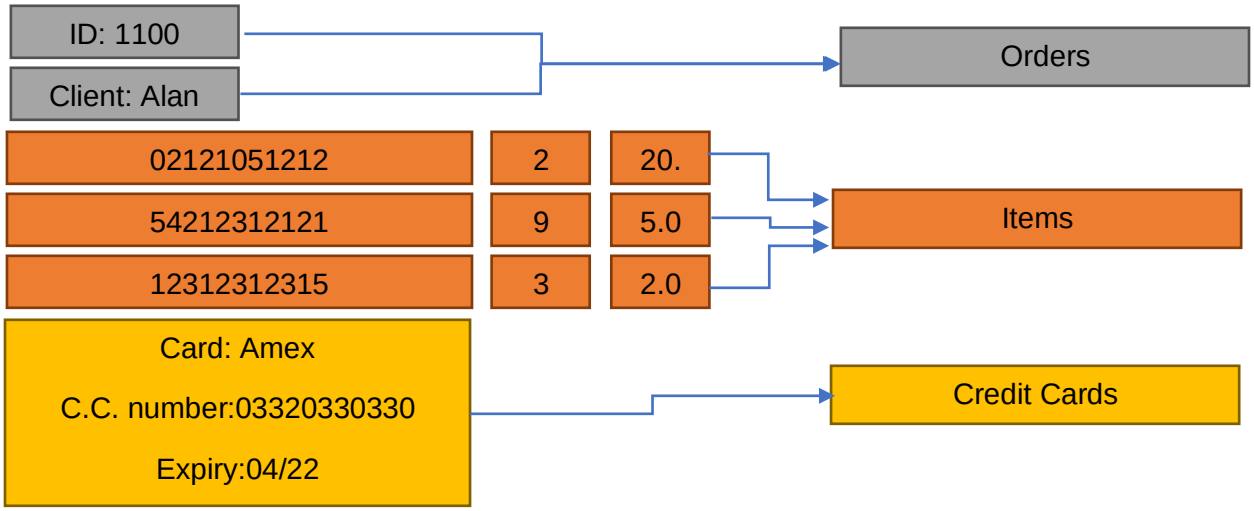


Figura 3 . Ejemplo de base de datos no relacional por documento.

Las bases de datos no relacionales han sido utilizadas con más frecuencia a diferencia de años anteriores. Se debe a la gran cantidad que se procesa segundo a segundo ofreciendo mayor escalabilidad y flexibilidad gracias al gran número de comunidad que las respalda. Con frecuencia se desarrollan nuevas funcionalidades y soluciones bajo el esquema de código abierto. Aunque no cuentan con lenguaje estandarizado como las relacionales, no supone problema al utilizarse.

2.7 **Data warehouse**

En este trabajo se consideran los datos oficiales que exhibe la federación. Además del almacenamiento en la base de datos se propone un proceso de transformación, el cual, deja por sentado las bases para segmentar los datos con fines de análisis. Se debe hacer mención de la técnica de data warehouse y los procesos que la acompañan. El origen de data warehouse ocurre cuando los requerimientos entran en conflicto entre lo operacional y la toma de decisión (Powell, 2006).

Es importante mencionar el concepto de data warehouse cuando se trata de datos con gran volumen. A medida que se busca el análisis de los datos y mejorar la velocidad de consultas, debe tomarse en cuenta ciertos criterios para transformar los datos sin remover el contexto de los mismos. Los métodos que intervienen en la aplicación de un data warehouse buscan facilitar el proceso de gestión. Es importante definir correctamente el concepto y las características que lo conforman.

Se espera que al hacer una consulta en bases de datos esta sea capaz de devolver el valor que se está buscando en el momento en que se realiza la petición, pero esto no ocurre en algunas ocasiones, la actividad de búsqueda y entrega depende de la infraestructura en la cual se encuentra el motor, dicho de otra forma, capacidad de cómputo y conectividad. En el presente trabajo se mencionará el concepto data warehouse el cual se traduce como almacén de datos, pero el uso común recomienda la utilización de su concepto en inglés.

Por otra parte, la arquitectura para aplicar data warehouse depende del alojamiento y la consulta de los datos, además de la distribución. La arquitectura global es la más extensa, está distribuye toda la información en todas las áreas por la cual es diseñada, también es conocida como centralizada, los datos son extraídos por recursos externos para que el usuario final utilice los recursos. Las arquitecturas mencionadas toman como referencia de inicio el modelo de datos, ya que el data warehouse es construido a partir de él. Entonces es necesario separar dos componentes principales a los cuales se les conoce como high level data y mid level data (datos de nivel alto y medio). El alto nivel es considerado como la capa que contiene las entidades y el mediano ayuda al diseño de iteraciones (Inmon, 1999).

Existen factores a tomar en cuenta antes de hacer la implementación del data warehouse, el primero en evaluar es el volumen de datos que serán segmentados, estos deben ser

analíticamente separados para brindar solo la información necesaria para la toma de decisiones, se tiene que tomar en cuenta la periodicidad y la importancia. En lo que refiere a la velocidad de datos al consultar, es necesario evaluar el hardware ya que este, depende directamente de la velocidad de procesamiento. Historia de la organización o justificación del tema, este factor es sumamente importante, es el que da sentido a la lectura de los datos, por ejemplo, existen diferentes analíticas con datos tipo espacial y datos médicos, esta diferencia es la raíz de la justificación del trabajo empleado en los datos. Existen otros factores tales como, cantidad de usuarios que consultan la información, costos de tecnología y límites. Las OLTP (on-line transaction processing) son aplicaciones que se ejecutan día a día entorno a la gestión de datos. Los procedimientos observados en dichas transacciones son las de altas, bajas y modificaciones, consultas, transacciones, actualizaciones en tiempo real, es decir se busca que los procesos sean rápidos y efectivos.

Para consolidar el proceso de data warehouse es necesario obtener los datos resumidos y los transformados, este proceso está compuesto por tres partes. Validación de datos consistentes, mecanismos de consolidación, y factor técnico que este último refiere a la forma en que los datos son transportados, tiempo y datos transformados.

Una vez que los datos son consolidados se tiene como resultado data marts, estos son la base de información del negocio o de la conceptualización en la cual se aplicó la herramienta de data warehouse, lo que se busca es resumir la información de los datos consolidados. Las ventajas de obtener los datos en data marts se encuentra en la facilidad de legibilidad al utilizar alguna aplicación para expresarlos en toma de decisiones.

2.7.1 **Aplicaciones**

Al ser una segmentación de datos para brindar información específica, el data warehouse es utilizado para la toma de decisiones en organizaciones o instituciones con fines de análisis, por lo general los datos sirven a los sistemas de información o aplicaciones para que estas contengan información que mostrar, pero a diferencia de esto los datos tratados o transformados por data warehouse está orientado al cliente final, muestra datos históricos con sentido de análisis y periodicidad, la información tiende a ser más detallada y resumida. Por lo que las aplicaciones data warehouse, suelen existir en objetos de mercadotecnia o dirección.

2.7.2 **Preprocesamiento de datos**

El propósito fundamental de la preparación de los datos es manipular y transformar los datos crudos que contienen información de suma importancia con posibilidad de ser expuestos o hacerlos accesibles (Pyle, 1999). En un set de datos ocurre, que los datos son impuros, esto quiere decir que carecen de sentido o resultan poco útiles. Estos se pueden encontrar como datos incompletos, datos con ruido o inconsistentes. También se considera que el resultado del preprocesamiento puede generar conjuntos de datos menores al original. Si los datos tienen sentido o son consistentes, existe la posibilidad de seleccionar los importantes y esto excluye al resto de los datos, también existen selección por características o la discretización de los mismos. Hay un sin fin de estrategias para el procesamiento de datos, sea cual sea la herramienta a seguir el resultado será el mismo, el sentido a la información.

2.7.3 **Manipulación de set de datos**

Las colecciones de datos son llamadas datasets, comúnmente pertenecen a una misma entidad de base de datos donde cada una de las columnas de matriz representan una variable y los renglones corresponden al dato.

Normalmente cuando manejamos fuentes de datos estas se encuentran de forma de datasets heterogéneos, es decir, están compuestos de diferentes partes que forman un mismo sistema de base de datos, pero al ser percibidos por el usuario estos ya contienen propiedades únicas de la relación.

2.8 **Arquitecturas**

Para el enfoque de esta investigación será necesario implementar arquitecturas para el desarrollo de herramientas de software. Dichas herramientas serán el medio para recorrer el proceso de extracción, diseño, almacenamiento y exposición de los datos. Existen distintos tipos de arquitecturas para el desarrollo de software; sin embargo, se seleccionarán aquellas que se apeguen al objetivo de la investigación. Si bien cuando se planea hacer un sistema o aplicación de software se deben tomar en cuenta todos los requerimientos, esto la mayor parte de las veces supone un gran obstáculo, ya que el software está pensado para el continuo aprendizaje e implementación. La mayoría de los sistemas tienen subsistemas que lo apoyan al realizar los procedimientos para los cuales fueron construidos, es considerado implementar diseños que se adecuen a la orquestación de producto final.

En años anteriores era común pensar en una idea e irla realizando conforme se escribía código o el diseño, pero en la actualidad esto resulta una pérdida de tiempo por la competencia tan cerrada que se tiene entre desarrolladores, las organizaciones cada vez exigen más en menor tiempo (Len Bass, 2003). El nivel arquitectónico de la estructura de un sistema es aquella descripción donde se utilizan conectores diferentes a llamada a procedimiento y/o se imponen restricciones importantes entre los componentes y/o aparecen distintos tipos de componentes en la descripción y las arquitecturas de software se enseñan-aprenden en forma opuesta a la forma en que se realiza en la práctica (Cristiá, 2008). Un componente en software es la entidad computacional activa, el conector es el medio por el cual se comunica, coordinan los componentes.

2.8.1 **Arquitecturas de Plugin**

Se le conoce al plugin como pieza de software que se relaciona con un sistema o aplicación, este tiene como finalidad el extender la funcionalidad del sistema al que está acoplado permitiendo hacerlo más robusto u operativo. Los plugin no deben ejecutarse por sí mismos, estos deben ser invocados por su anfitrión, a su vez el plugin es capaz de registrarse en el propio sistema para dejar una marca de inicio, a esto se le conoce como firma plugin. Comúnmente son utilizados en aplicaciones que crecen con el tiempo o aquellas que permiten múltiples funcionalidades o conexiones con terceros.

Dentro de la arquitectura de plugin se encuentra la reflexión de datos, se le conoce como la capacidad del código de identificar y gestionar sus metadatos con el objetivo de informar al usuario sobre el funcionamiento de las aplicaciones. Otro de los conceptos que aparece en plugin es el manejo de errores, este tiene como función revisar comportamientos no deseados en las aplicaciones, desde mensajes al usuario o excepciones dentro del sistema que no detengan los procesos.

Para el desarrollo de software científico es necesario tener en cuenta qué información se requiere al utilizar una aplicación por lo que, la arquitectura de plugin resulta una herramienta eficaz, los metadatos expresan los valores contenidos, el manejo de errores presenta la eficacia en el sistema.

Uno de los antecedentes de la investigación es un trabajo de software. Dejó como precedente el uso de arquitectura para crear un almacén de datos. La arquitectura soporta la escalabilidad del sistema y la reusabilidad por lo que, será importante implementar arquitecturas para las herramientas de software que se desarrollarán en el trabajo de investigación.

2.8.2 **Middleware**

Se le conoce middleware al nivel que está localizado entre los sistemas operativos y los sistemas de comunicación, este percibe la comunicación por medio de reglas definidas y protocolos para transformarla en peticiones o entregas según el flujo en el que se ejecute, dicho de otra manera, se encarga de esconder la heterogeneidad de los datos. El middleware pertenece a la computación distribuida el cual se compone de servicios que permiten múltiples procesos sin importar a cuantas máquinas se dirijan.
Los servicios usuales que se utilizan en esta capa son los de identificación, autenticación, certificados y seguridad. Este concepto aporta escalabilidad, también aporta que el desarrollador no tome en cuenta protocolos de interacción en sistemas operativos o plataformas. A continuación, se en listan los diferentes tipos de middleware:

- El middleware por gestión de datos, permite que las aplicaciones puedan leer y escribir en bases de datos o recursos remotos (Fig. 4).
- Middleware de comunicación, soporta los protocolos de comunicación para la transmisión de mensajes o datos p2p, así como herramientas que le permitan ejecutarse como servicio.
- Middleware de plataforma, se basa en contenedores para componentes de aplicaciones, es el intermediario entre dos programas que necesiten comunicarse. Este también puede proveer los canales de comunicación, como mensajes multimedia o ejecuciones entre sí.

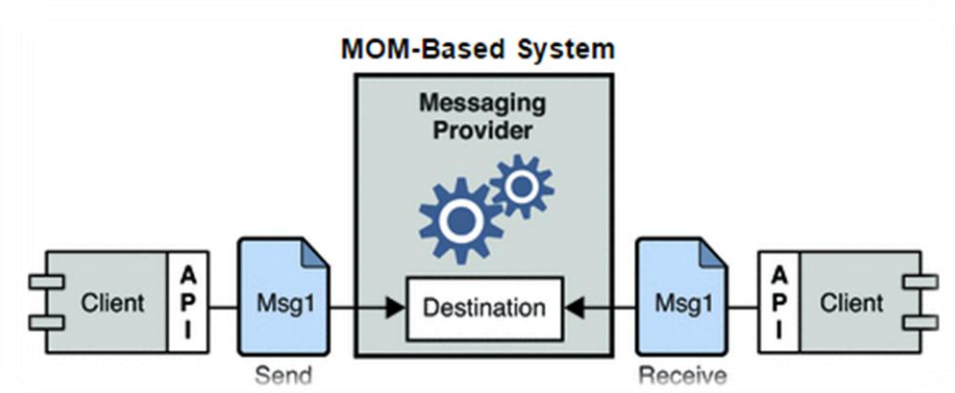


Figura 4. Middleware orientado a mensajes o datos. Extraído de middleware (weizmanc.blogspot.com)

2.8.3 **Servicios**

Los servicios se basan en arquitecturas de objetos distribuidos que se comunican entre sí con XML, utilizando protocolos de internet. Para que exista interacción entre aplicaciones deben existir estándares de comunicación. SOAP (simple object access protocol) es un protocolo de información descentralizado y distribuido.WS-Addressing (direccionamiento de servicios de la web), es el encargado de transportar los servicios y mensajes.

Los servicios están conformados por capas de transporte, información, mensajería y descubrimiento. La capa de transporte es la inferior, es donde se utilizan los protocolos, tales como FTP, SMTP, HTTP. Los servicios de mensajería describen cómo la información que contienen los datos se intercambian entre computadoras. En cambio, los WSDL o descripción de mensaje, aportan información a las aplicaciones sobre el formato en que se comunican. Seguido de las definiciones anteriores, existe la arquitectura orientada a servicios, “SOA es un modelo de componentes que interrelaciona las diferentes unidades funcionales de las aplicaciones, denominadas servicios, a través de interfaces y contratos bien definidos entre esos servicios. La interfaz se define de forma neutral, y debería ser independiente de la plataforma hardware, del sistema operativo y del lenguaje de programación utilizado. Esto permite a los servicios, construidos sobre sistemas heterogéneos, interactuar entre ellos de una manera uniforme y universal.

2.9 **Interfaz de programación de aplicaciones (API)**

Como objetivo específico de la investigación, se propone el desarrollo de un servicio que exponga la información de la base de datos obtenida del gobierno federal COVID-19 México. Se plantea desarrollar un software que sirva de intermediario entre los datos almacenados y el cliente. Dicho esto, es importante resaltar los procesos que se involucran al comunicar dos aplicaciones y que resultados se pueden obtener.

La API es utilizada para comunicar los componentes de software. Esta puede contener variables, rutinas, clases, objetos y estructuras (Neumann, 2018). La API depende del lenguaje de programación, su sintaxis y elementos, pero es independiente al ser utilizada por los distintos lenguajes de programación, es llamada por los servicios web.

El uso común es ejecutar tareas de terceros. Existen diversas normas de cómo escribir las API, pero lo cierto es que se recomienda que se parezca lo más posible al lenguaje humano, en caso de no ser posible por la complejidad en la que fue creada, cada desarrollador tiene la responsabilidad de crear documentación para que externos al servicio puedan utilizarlo sin problemas. Las Web API son las más frecuentes en el mundo de intercambio de información, estas contienen tareas y clases que devuelven la información en modo de objeto por ejemplo JSON, XML. Estas API utilizan protocolos SOAP o REST.

2.10 **Formatos de set de datos**

2.10.1 **Documentos XML**

Lenguaje de marcado extendido (extended markup language, XML), es utilizado para el intercambio de documentos, tiene como principio ser claro al momento de presentar la información que contiene, utiliza distintas marcas para cada aplicación, es conocido como metalenguaje. Pertenece a los formatos abiertos, este es independiente de la plataforma en la que se utiliza, describe su contenido como vocabulario, las aplicaciones pueden interactuar con según el contenido o detonar procesos a partir del mismo.

2.10.2 **Formato JSON**

Es un formato de compresión ligero basado en literales y matrices de javascript, el nombre reside en las palabras Javascript object notation (notación de objeto javascript). Las literales se

especifican con corchetes para delimitar los valores, estos pueden ser cadenas, números, booleanos o nulos. También puede escribirse en forma de literales de objeto, el cual empieza con un string de la llave para ingresar al valor. Las cadenas JSON se utilizan para el intercambio de información, no contienen variables como el lenguaje. Es responsabilidad de los clientes de aplicaciones, codificar o decodificar este formato para su uso.

2.10.3 **Archivos CSV**

Según (Google, 2021) “Un archivo CSV (valores separados por comas) es un archivo de texto que tiene un formato específico que permite guardar los datos en un formato de tabla estructurada.” Estos archivos son utilizados para el intercambio de información y actualmente la mayoría de las aplicaciones lo reconoce, su uso más común aparece en sistemas de organizaciones. Como su nombre lo indica, los registros están separados por comas lo cual hace legible la información contenida, cada línea del archivo es un registro de datos. El CSV no está estandarizado, esto en ocasiones genera problemas con la información contenida, al ser la coma un separador o algunas veces el punto y coma, se vuelve susceptible que el texto o cadena contenga estos caracteres, el resultado suele ser difícil de interpretar.

2.11 **Plataformas de información**

Desde siempre se ha divulgado la información para que terceras personas se informen de temas y conceptos referente a un contexto de descubrimiento. La forma de realizar esto empezó por vía oral, escrita, bibliográfica. En el presente internet y medios digitales han agilizado este proceso a pasos agigantados, la educación llega a quien la necesita y las noticias por igual, se estima que hay tanta información que resulta un problema corroborar la credibilidad de la misma.

2.11.1 **Plataformas informativas**

Estas plataformas brindan información al usuario, pueden ser estáticas o dinámicas, cuando los datos son fijos y su entrada ocurre por periodicidad recaen en lo estático, si ocurre el caso contrario y los usuarios pueden manipular los datos son dinámicas. La OMS tiene a disposición la plataforma informativa de casos COVID-19 (WHO, 2021).

En el mismo contexto el gobierno de México presenta los datos diarios de seguimiento de COVID-19 en el portal de gobierno del estado (GobMx-b, 2021).

2.11.2 **Plataformas exploratorias**

A diferencia de las plataformas informativas, estas ponen a disposición los datos para la manipulación por parte del usuario, usualmente pertenecen a un mismo contexto de investigación y cuentan con múltiples fuentes de información ya verificadas. Con este tipo de plataformas se busca tener los datos históricos o presentes para sustentar investigaciones. Algunas de ellas cuentan con modelado de los datos, mapas para georreferenciados, gráficos para datos administrativos y pronósticos para análisis de datos.

Para COVID-19 los investigadores y expertos en el área de tecnología desarrollaron plataformas para concientizar lo ocurrido y además para fijar pronósticos o tomar decisiones con respecto a ello. *World Meter (sitio web sobre datos),* puso a disposición los datos en tiempo real de casos activos y cerrados, además de múltiples variables por país y continente (meter, 2021). Otra plataforma muy completa es la de la universidad de medicina John Hopkins, que ha sido reconocida mundialmente por la información precisa que contiene y sus modelos (hopkins, 2021).

De manera nacional, la Universidad Nacional Autónoma de México (UNAM), participó con la disposición de datos en su portal, llevando el seguimiento de la enfermedad por estado e índices de población (UNAM, 2021). Así mismo se desarrolló la plataforma del Consejo Nacional de Ciencia y Tecnología (CONACyT) que reúne los mismos atributos, pero presenta información más detallada tanto en datos como en metodología.

2.11.3 **Plataformas como repositorios**

Son aquellas plataformas que se comparten por medio de instituciones con fines educativos, estas se apegan a normas de interoperabilidad de objetos, dichas normas describen la estructura que deben seguir y los datos descriptivos que deben contener.
Se basan en programas educativos y plataformas de auto aprendizaje, comúnmente son utilizados por investigadores o estudiantes que buscan algún contexto en específico.

Capítulo 3. Metodología

En este capítulo se determinan los métodos mediante el cual se propone el modelo más adecuado para cumplir con los objetivos específicos. Se establecen las necesidades y requerimientos para llevar a cabo el trabajo de investigación y se trazan objetivos metodológicos.

- Se realizó un estudio bibliográfico para construir un marco teórico que permite establecer el estado del arte. La intención es poder explicar los procesos y la problemática que envuelve al trabajo de investigación.
- Se realizó un análisis crítico de los paradigmas que sustentan la base del problema.
- Se realizaron las limitantes del problema.
- Se realizó investigación sobre herramientas de software que apoyan el caso de estudio.
- Se definieron las herramientas de software.
- Se definieron el diseño y arquitectura de las herramientas de software.
- Se definió la metodología del desarrollo de software.
- Se validó el uso de las herramientas de software.
- Se realizó un análisis en forma de discusión sobre el caso de estudio.
- Conclusiones del trabajo.
- Trabajo futuro.

Fue necesario revisar literatura acerca del sujeto de estudio COVID-19 con respecto a los datos de forma global y posteriormente a nivel nacional en México. Después se investigaron los recursos existentes referentes al sujeto de estudio. Fue de gran relevancia analizar el proceso de obtención de la información y su comportamiento, además de los puntos débiles y posibles soluciones. En esta última reside el problema del presente trabajo.

Se decidió limitar el trabajo hasta la gestión de datos, excluyendo el análisis de los mismos. Por lo que fue necesario comparar herramientas de software para la administración de datos. Una vez seleccionadas las herramientas se buscó en la literatura técnicas de implementación de arquitecturas y diseño en software. Con esto se pretendió fomentar las bases de un sistema que cubriera cada aspecto del problema y funcionara como gestor de datos. El siguiente paso fue validar que las herramientas de software construidas como prototipos, pudieran solventar los objetivos específicos del trabajo de investigación.

Por último, se construyeron las discusiones sobre el tema, tomando en cuenta los resultados obtenidos, seguido se concluyó la investigación y se describió el trabajo futuro.

La herramienta de software fue construida a partir de los siguientes requerimientos:

- Descargar la fuente de datos oficial de COVID-19 México.
- Gestión de datos COVID-19 México y almacenamiento.
- Aplicar técnicas de software para el uso de grandes volúmenes de datos.
- El software debe ser tolerante a cambios en la fuente de datos.
- Bitácoras de comportamiento interno del software.
- Capacidad para segmentar los datos.
- Extraer datos en segmentos considerando variables de tiempo.
- Almacenamiento de datos históricos.
- Conexión por medio de servicio para realizar consultas.
- Debe ser escalable.

Para asegurar el funcionamiento del software y el cumplimiento de los requerimientos ya mencionados, se elaboraron casos de prueba. Los resultados esperados se describen en la tabla y su descripción se detalla en el capítulo 5 de validaciones, también al final del capítulo se agregan las conclusiones.

Nombre de prueba / Criterio	Almacenamiento	Extracción	Servicio	Comparación	Operaciones
Caso de prueba 1	X	X	X		
Caso de prueba 2		X		X	X
Caso de prueba 3		X			X
Caso de prueba 4	X	X		X	X
Caso de prueba 5	X	X		X	X
Caso de prueba 6		X	X		

3.1 **Metodología del desarrollo**

El desarrollo que se optó por seguir en esta investigación fue el modelo de cascada. El motivo por el cual se eligió dicha metodología es por su amplio análisis en cada uno de los procesos que la conforman. Además, el hecho de que los procesos del software sistema núcleo COVID-19 México funcionen de forma iterativa al igual que la metodología, se acomodó a la forma de evaluar su funcionamiento (figura 5). Como menciona Royce: “La metodología de cascada para el desarrollo de software es ideal al tener claro los requerimientos que se trabajarán, el proceso iterativo de cada módulo hace que no se pueda continuar sin tener resuelto un problema a la vez” (1970).

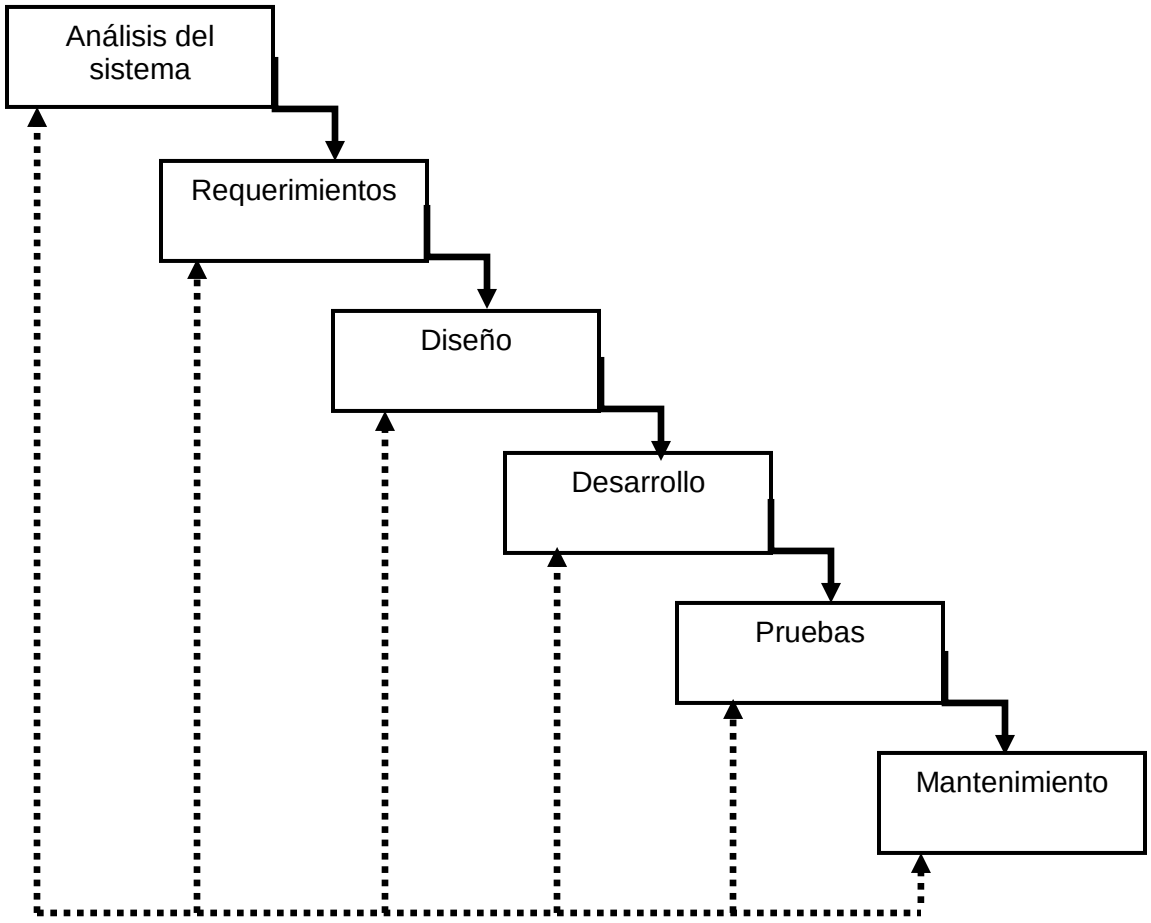


Figura 5. Modelo de cascada. Elaboración propia.

En la metodología fueron analizados y explicados los requerimientos, por consecuente, se asignaron tareas a realizar a cada una de las acciones para llevarlos a cabo. Si bien, esta metodología puede resultar lenta en comparación de las metodologías ágiles, la metodología cascada presenta resultados concisos. Además, al ser iterativo, esta puede regresar y expandir la solución según como se requiera. Aunque de manera negativa, un requerimiento se puede congelar, no completarse totalmente. La metodología permite terminar un módulo y continuar con el siguiente paso, esto tiene como resultado progresar en el sistema que se está desarrollando (Elizabeth Woodward, 2010). El enfoque sistemático y secuencial del método proporciona un correcto control de la entradas y salidas de cada módulo del sistema. El diseño y estructura son previamente analizados conforme a los requerimientos del sistema, estos son trazados como requerimientos que avanzan a través de la planeación por lo que, presentan un valor agregado a la comunicación del proceso.

Un sistema es una colección organizada de partes o subsistemas que se encuentran totalmente interrelacionados para cumplir un objetivo en específico. Este contiene entradas, salidas, procesos, evaluación y resultados. Cada una de estas deben ser consideradas independientes, ya que el conjunto de estos atributos bien elaborados, representan la solución de sistema de software.

El modelo tiene como entrada el análisis de sistemas que se adecuan al sujeto de estudio, toma los requerimientos como base y a partir de ello, se diseña el sistema. El desarrollo del sistema y la etapa de pruebas están en constante comunicación y el mantenimiento se aplica cuando se observa algún cambio en los requerimientos o funcionalidad del sistema.

La arquitectura que se utilizó, está inspirada por el trabajo de “desarrollo de una plataforma para la gestión de datos climáticos en malla” (Hernández, 2018). Dicho trabajo utiliza la arquitectura basada en plugin como se muestra en la figura 6. A diferencia de esta, no se consultan múltiples bases de datos, no obstante, se trabaja con grandes cantidades de datos y heterogeneidad entre los mismos.

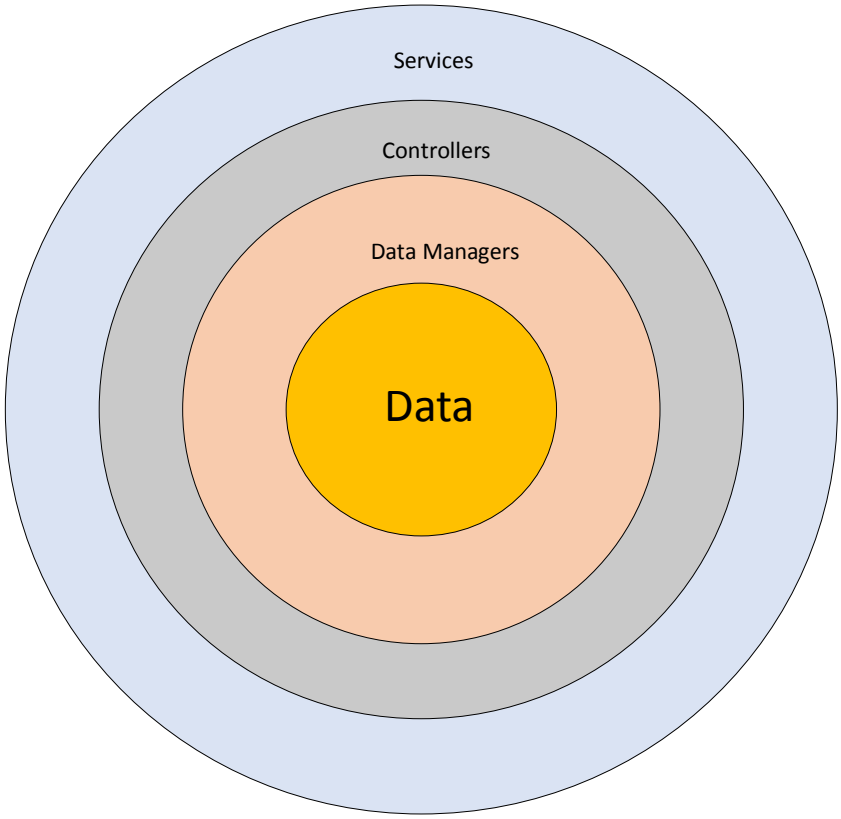


Figura 6. Diagrama de arquitectura Plugin.

Al construir el sistema, se consideró el patrón de diseño de acceso a datos por objeto (en inglés data Access object pattern, *DAO*). Se investigó sobre bases de datos relacionales y no relacionales. Durante la investigación se consideró cual resultaba compatible con el esquema de datos, siendo resultante las bases de datos relacionales. El manejador de base de datos que se utilizo fue PostgreSQL por su rapidez y configuración con respecto a los datos relacionales. Constantemente se cambió la configuración del motor de base de datos, para llegar a un punto estable de consultas y escritura de datos.

La etapa de pruebas se realizó en dos formas: pruebas de procesos del sistema y pruebas unitarias de la solución. La primera prueba está fundamentada con un archivo de logs que almacena el proceso que se ejecuta y el tiempo de duración. También al momento de corroborar que los datos estaban correctos en las nuevas entidades, se realizó un seguimiento de registros evaluados por un número limitado de forma aleatoria, estos registros fueron sometidos a consultas SQL para tomar nota de los resultados obtenidos. Por otra parte, la

segunda prueba está dividida en subpruebas, las cuales corroboran que la solución funciona en ámbitos de consultas, escritura, procesos especiales (véase) y servicio. Cabe mencionar que cada solución de software cuenta con su archivo de versiones, el cual contiene información de las actualizaciones, fecha y motivos.

Ambas soluciones de software se instalaron en un servidor IBM power ubicado en la UABC, en específico en el laboratorio EDUMAT a cargo del Dr. Gabriel A. López Morteo. Desde el mes de noviembre del 2020 los datos son expuestos por el servicio desarrollado.

Capítulo 4. Análisis y diseño de la solución de software

El objetivo del sistema es eliminar los procesos intermedios que pudieran existir para la obtención de los datos de COVID-19 en México y su gestión. El tiempo que se recupera al no invertir herramientas o estructuras para la manipulación de los datos cada vez que se requiera, resulta benéfico para el uso de técnicas de análisis. Además, la plataforma se desarrolló con funcionalidad tolerante a cambios, es decir, si por alguna razón las variables del contexto cambian, no es complicado adaptarse a dichos cambios y estos no presentarán carga de trabajo considerable. El tamaño del sistema es como lo requiera el contexto y los límites de análisis que se necesiten aplicar.

Fue necesario analizar cada módulo del sistema, esto con la finalidad de seccionar los procesos que participan en cada una de las partes del software. Finalmente, este constó de cuatro módulos importantes los cuales se encargan de brindar como resultado los datos COVID-19 en México de una forma estructurada y pensando en que estos serán analizados.

4.1 **Análisis**

La elaboración de un prototipo resuelve la implementación de características parciales de la solución. Este brinda la flexibilidad de realizar el análisis sobre la marcha y detenerse cuando se pierde el objetivo.

- El sistema debe contener legibilidad, expresar facilidad en la interpretación del código.
- Debe de ser independientemente funcional en sus componentes.
- Debe ser adaptable para el cambio de contexto de estudio, desde las fuentes de información y almacenamiento.

Se estudió el comportamiento del usuario del sistema y las posibles peticiones que se pudieran realizar. Otro de los factores a tomar en cuenta fue la gestión de los procesos y la huella digital.

4.2 **Diseño**

Se tomó como referencia para la funcionalidad interna del sistema, la arquitectura basada en capa que consiste en la subdivisión estratégica del sistema en sub módulos, que funcionan por

sí solos, integrados a una capa de presentación la cual, entrelaza los “n” módulos que se especifiquen (Sommerville, 2005).

En primera instancia se realizó la propuesta de un modelo estructural, ejerciendo la separación del sistema en módulos. Paso seguido se elaboró el modelo de proceso dinámico, consiste en la influencia en tiempo de ejecución de los submódulos. El modelo de interfaz, define el uso de los servicios o procesos y por último se asigna un modelo de distribución, el cual conecta el sistema con otros sistemas con distinto contexto.

La figura 7 describe la estructura del sistema dentro de la arquitectura de plugin. Como primer elemento se encuentra el plugin manager, este ejecuta y hace obtención de los componentes del programa, es decir, permite el acceso a solicitudes externas implementando así un esquema de interoperabilidad. Cuando se realiza un enrutamiento del programa, se escribe en un archivo de texto el estatus de la función y los tiempos de ejecución, a esto se le llama bitácora. El archivo de configuración contiene los parámetros para la ejecución del plugin y las rutas de los componentes que se deben ejecutar. Se observa que el componente principal de la arquitectura es el de aplicación ya que, es donde existe más dinamismo de procesos. Esta se encarga de la iteración de los submódulos o fases del sistema, funciona como interface del servidor ya que, procesa de forma iterativa los datos y da lugar a que se realice el almacenamiento de los mismos. Fue llamado server Python al módulo que se encarga de realizar la descarga automática del archivo de datos, además cuenta con funciones que construyen las plantillas de los datos que, serán almacenados en las bases de datos. También realiza todas las operaciones ETL y se apoya de métodos nativos del sistema operativo.

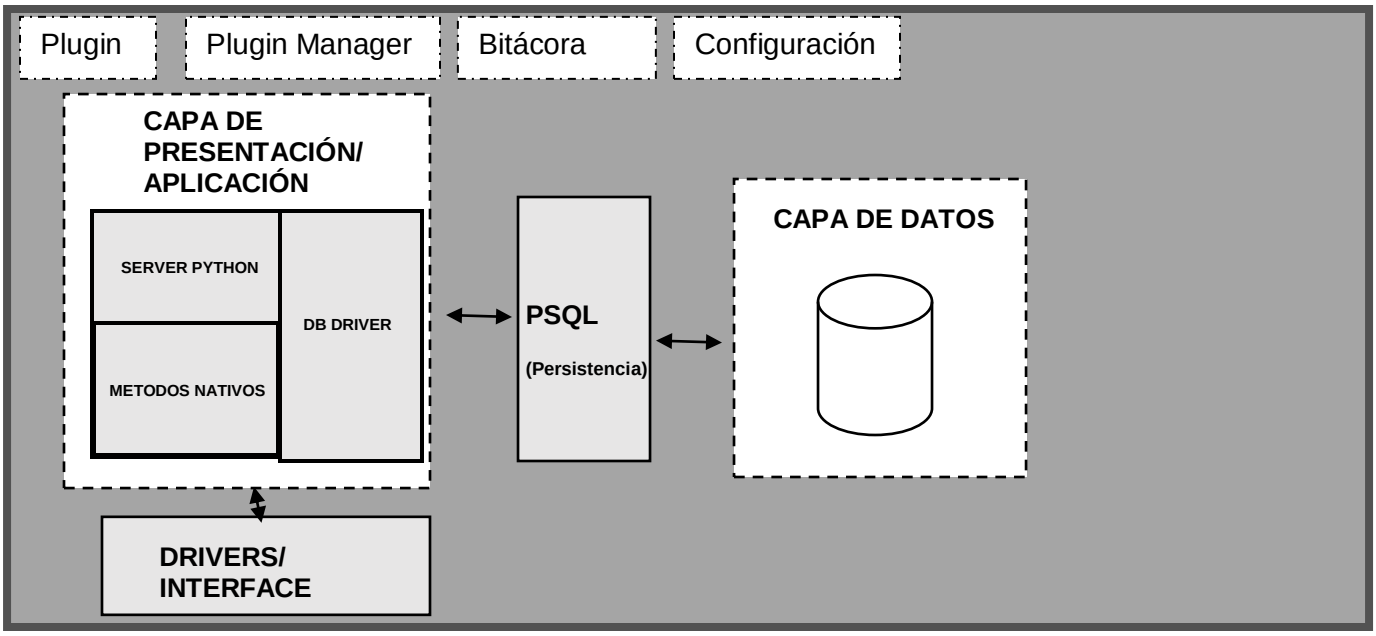


Figura 7. Modelo arquitectónico. Elaboración propia.

El módulo de métodos nativos se refiere a los métodos que se excluyen del lenguaje de programación que se utilizó. Este realiza funciones de creación y destrucción de ficheros temporales y son utilizados para crear rutas que apoyan el enrutamiento del sistema, por ejemplo, los espacios de memoria donde se colocan los archivos históricos de datos o el archivo diario.

El módulo DB driver ejecuta scripts de conexión por medio de controladores de bases de datos, en este caso, se utilizaron conexiones de tipo PSQL los cuales tienen reflejo en la base de datos tanto de escritura como de consulta. La capa de aplicación puede ser utilizada como componente o tiene apertura para que programas externos interactúen con la misma.

La capa de datos es la que se encarga del almacenamiento y consulta de la información. Cuenta con archivos de configuración para el apoyo de transacciones de datos respecto a segmentación de datos en lotes o recursos de cómputo. También asegura los datos por medio de credenciales de acceso. La arquitectura permite que se utilice cualquier gestor de bases de datos para el almacenamiento de la información; sin embargo, como se explicó anteriormente, se tomó en consideración que esta perteneciera a las bases de datos relacionales por el contexto del problema.

Para acceder a los datos, se utiliza la capa de PSQL. Este está constituido por scripts que tienen como función realizar sentencias SQL para consultas, extracción de la base de datos y escritura de un nuevo registro. A este intercambio se le conoce como la capa de persistencia, ocurren los procesos ETL. Esta capa tiene el atributo que puede extenderse según se requiera. Por otra parte, se desarrolló un servicio el cual hace consulta a los datos almacenados..

La capa de drivers interface, permite la conexión de programas externos al sistema que realizan tareas en conjunto con el hardware de conexiones en red.

4.3 **Descripción de procesos del sistema Núcleo COVID-19 México**

El proceso núcleo del sistema heterogéneo es procesado por un plugin de gestión de datos, el cual consta de fases para el control de su funcionamiento. Este tiene como paradigma un acomodo por capas, el cual intenta simular la arquitectura ya descrita con el mismo nombre. En primer nivel se encuentra la capa de presentación, posteriores a esta, capa de aplicación, lógica de negocios y capa de acceso a datos. Como se describió anteriormente es un acercamiento, ya que difiere del comportamiento actual.

4.4 **Fases de funcionalidad**

La figura 8 muestra los procesos que integran la funcionalidad del plugin. Está compuesto por 5 fases que en conjunto realizan el proceso de descarga de servicio de datos y almacenamiento de los mismos, además de un tratamiento especial con respecto a preprocesamiento y asignación de entidades resultado de técnicas de data warehouse.

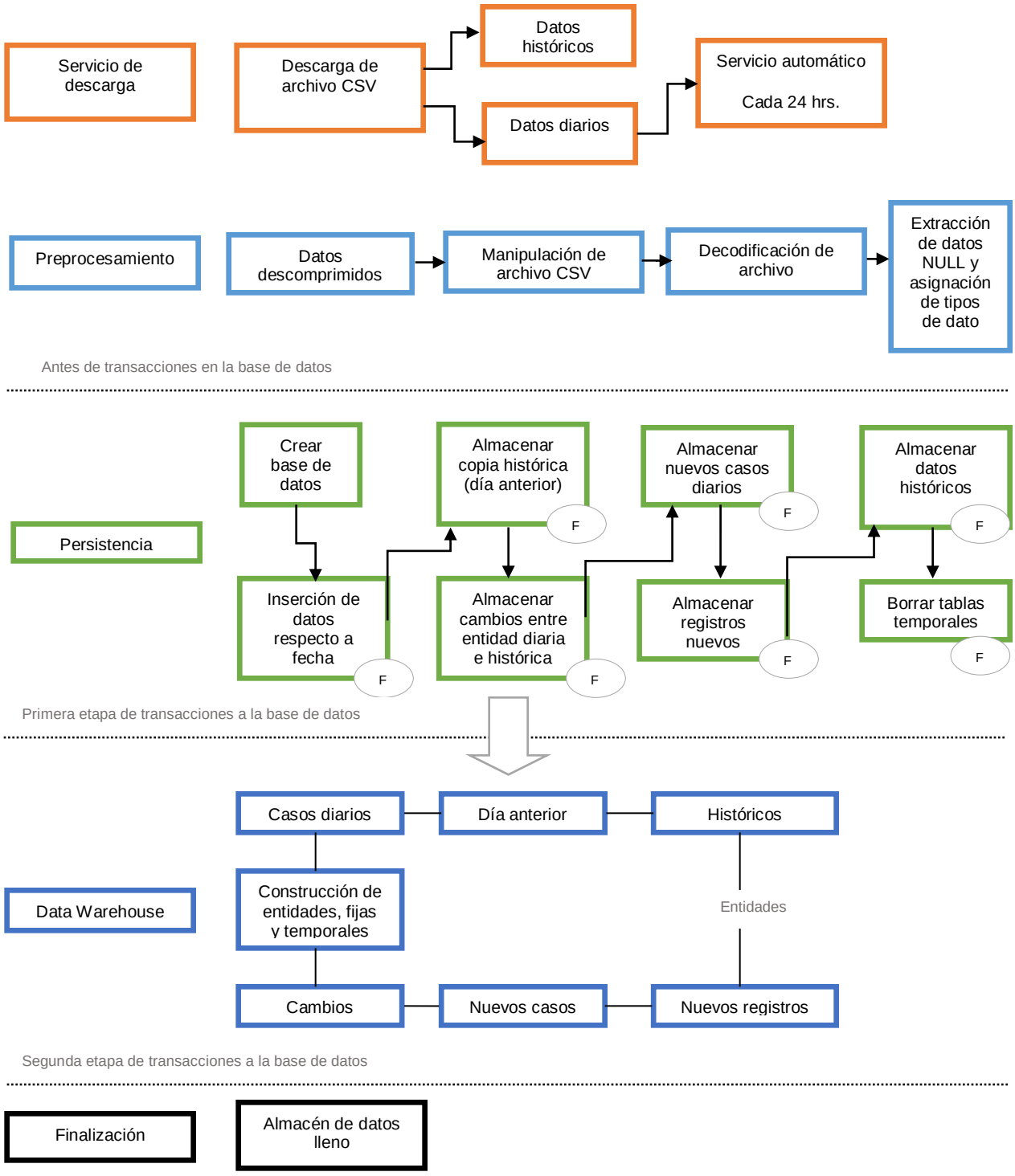


Figura 8. Diagrama de funciones del sistema.

4.5 **Descripción Fase 1. Servicio de descarga de datos**

El gobierno federal, en específico la secretaría de salud emite, de acuerdo a la especificación de frecuencias con clasificación R/P1D (diariamente ISO-8601) los datos abiertos sobre Covid19-Sars2 con fecha de hallazgos desde el 12 de abril del 2020. Estos datos se encuentran compresos en un archivo con extensión zip, el cual contiene un archivo .*CSV*, es decir, separado por comas y es expuesto diariamente. El documento contiene los datos del día anterior a su exposición, por ejemplo:

Abril 13 del 2020 contiene los datos con fecha de actualización del día 12 de abril del 2020.

El acceso al servicio de descarga ha cambiado en varias ocasiones, esto ha generado conflicto en los servicios automáticos de descarga, aunque el plugin permite cambiar esta dirección sin problemas y se tiene una bitácora de acciones, no se descarta que el personal técnico requiera revisar que todo este correctamente en funcionamiento. El nombre del archivo está compuesto por el emparejamiento de dos conceptos, primero la fecha de adquisición seguido de "COVID19MEXICO" para establecer un formato de entrega.

Para establecer una correcta gestión de datos es necesario que la descarga se ejecute de manera automática, tomando en cuenta los obstáculos antes mencionados. En consecuencia, se han asignado bitácoras de control en caso de que la conexión con el servicio de descarga falle.

4.5.1 **Servicio de descarga**

Para que el plugin de gestión de datos heterogéneos funcione es necesario que como entrada se almacenen los datos a estudiar y manipular, dicho esto, nuestra fuente de datos existe como servicio por medio de una URL del portal de la secretaría de salud en la sección de datos abiertos. El primer paso es generar una consulta por *request*, función de la librería de Python que lleva el mismo nombre este contiene la respuesta del servidor por medio de un objeto. Desde el apartado de datos abiertos podemos acceder a los datos diarios e históricos además de los elementos del diccionario de datos que permite la comprensión de los encabezados en la base de datos .

Teniendo en cuenta que la ruta del *URL* difiere en la solicitud del recurso, se enfoca como principal acción la descarga diaria, dejando como extra los procesos de toma de historial o

descarga de diccionarios, este *URL* tiene como formato, protocolo, sub dominio, dominio, extensión de la web, carpetas, recurso.

La clase que efectúa el proceso de descarga se llama *DownloadUrl* la cual tiene como método el descargar la información del servicio. Esta clase también cuenta con la entidad de una bitácora de la clase *Log*, la cual guarda información del tiempo en el que se ejecuta, el paso exacto en el que se encuentra dentro del proceso, el tiempo de ejecución inicial y el final, se tiene excepciones de tipo error en la llamada del recurso la cual no detiene el proceso del plugin, solo guarda el log y este se queda en espera. Dicho proceso es ejecutado por un temporizador al que se le llama *schedule*, componente de la librería *schedule*, se configuro con tiempo de las 16:00 horas todos los días para realizar la llamada al recurso.

Por otra parte, el proceso de descarga de histórico es más complejo y es representado por la figura 9, este pertenece a la clase de extras del plugin donde se genera por medio de la URL correspondiente la llamada al recurso, este se compone por el protocolo, subdominio, dominio, extensión de la web, sub carpetas con nombre de la sección e histórico, sub carpeta del mes en número, recurso.

Se crea una variable de tipo lista la cual contiene los URL con el número correspondiente de mes y se llama una función que itera para realizar una manipulación de cadena de caracteres para cada mes correspondiente, esto porque la posición del número cambia cuando este es de dos unidades, como es el caso de octubre, noviembre y diciembre. Una vez que las listas están preparadas se crea una instancia de la clase de descarga que acepta como parámetro el URL correspondiente, este está iterando, incrementando el número del 1 al 31 de todos los meses del año, cuando no existe el día para el mes, se hace caso omiso y continua con el siguiente, la excepción de consumo de recurso es la que permite que la aplicación se ejecute sin problemas.

En ambos escenarios de descarga una se genera una variable de segmento de datos la cual permite que al momento de realizar la petición esta sea controlada desde el principio hasta el final, esta tiene el valor de 128 mb/seg. Dicho de otra forma, le decimos al servidor del recurso que se ha obtenido la información sin problema de 128 mb por cada segmento.

Estos archivos son resguardados en un folder dentro de la estructura del plugin llamado *history*.

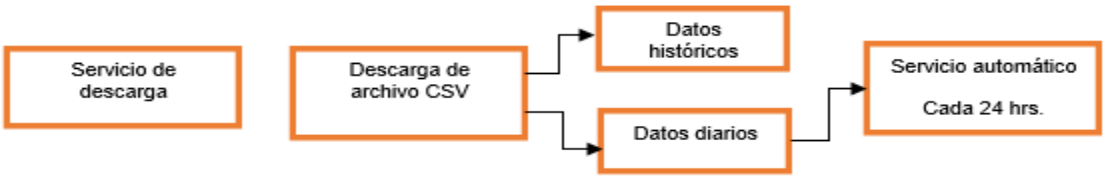


Figura 9. Diagrama de fase del servicio. ***Elaboración propia.***

4.5.2 **Proceso para descomprimir el recurso**

Una vez que el archivo del recurso está almacenado en espacio de información, este existe en formato comprimido .zip

Para descomprimir el archivo se generó una instancia de la clase *Unzip* de la librería de *unzip_python* la cual toma como parámetros la ruta del archivo y el nombre con el que será conocido dentro del espacio de memoria. La capa de aplicación genera un llamado al método de búsqueda del archivo y lo ejecuta. El archivo contiene un archivo .CSV, el nombre que recibe está conformado por la fecha en la que fue creado. Posterior a esto la aplicación toma la ruta y deposita el contenido en la carpeta de “res”, este recibe el nombre por la palabra “recurso”.

4.6 **Descripción Fase 2. Preprocesamiento de los datos**

El concepto de preprocesamiento de datos o información está explícito en el capítulo de fundamentos, pero como referencia, esta fase se encarga de revisar el contenido que expone el servicio de descarga y prepara los datos para su posterior almacenamiento. Es ejecutado desde la descompresión de datos y sigue con la manipulación del archivo separada por comas (*CSV*). Dicho archivo carece de tipado de datos, es decir no se puede apreciar si los campos son de tipo enteros, cadenas de caracteres, booleanos o fechas. Es importante considerar estos tipos de datos por que el resultado del plugin es proveer un servicio de consumo, el cual facilita la interacción con el usuario para consulta. A partir del proceso de manipulación se realizan segmentaciones de datos y se ejecutan las consultas avanzadas SQL, procesos que son explicados en el capítulo de persistencia (figura 10). En el proceso de preprocesamiento de datos ocurren operaciones de disminución de ruido, integración de datos de fuentes externas y transformación de los mismos. S. García et al. (2014).

Los datos que han sido procesados se deben almacenar en variables temporales para su almacenamiento por medio de instrucciones SQL.

4.6.1 **Desde datos descomprimidos**

Figura 10. Proceso de preprocesamiento del archivo de datos. Elaboración propia

El archivo separado por comas es el resultado del último proceso en fase de servicios, por lo tanto, el archivo se encuentra sin formato y contiene toda la información de registros COVID-19. Es alojado en la carpeta de recursos del plugin, este por sí mismo es capaz de encontrarlo para su manipulación por medio de la extensión y una vez encontrado es manipulado en tres ocasiones, una para decodificar, para realizar preprocesamiento de datos y asignar tipos de datos (fig.11).

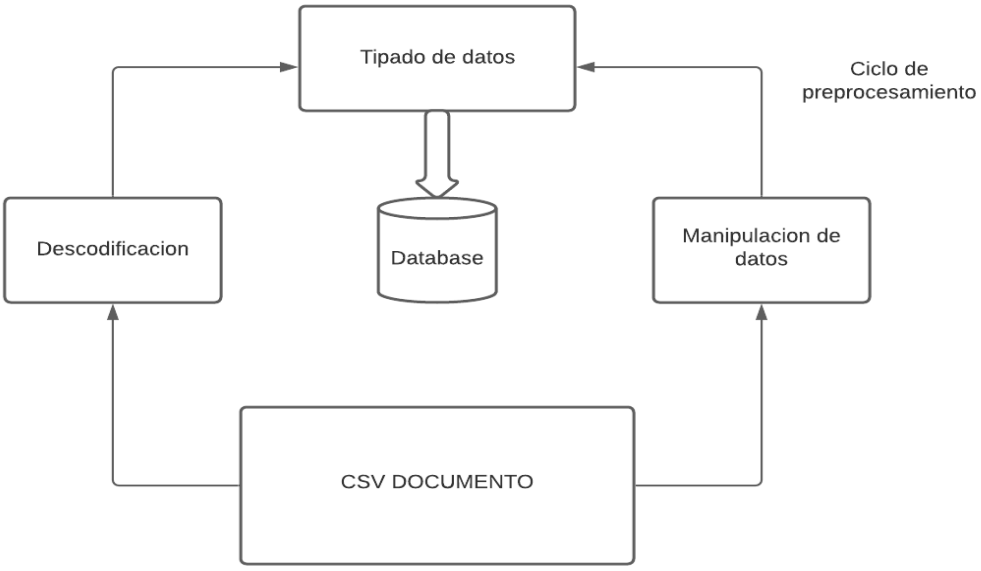


Figura 11 Ciclo de preprocesamiento de datos. Elaboración propia.

4.6.2 **Decodificación de archivo**

Además de tener el control de la posición del archivo es necesario abrir este para revisar que información se tiene y llevar de forma segura la manipulación de datos, dicho archivo es proporcionado por la dependencia de salud mexicana lo que lo hace obvio que se encuentre en lenguaje de habla española y utilice caracteres especiales como acentos o tildes, además de caracteres especiales como la letra “ñ”, un ejemplo de esto es, el encabezado de nacionalidad toma en ocasiones el valor de México, esta palabra contiene un acento como consecuencia se tiene que mantener la palabra al momento de generar su inserción en la base de datos. Visto lo anterior se tomó la decisión de decodificar el archivo en Latin1 (ISO/IEC 8859-1).

4.6.3 **Manipulación de datos**

Por consiguiente, es necesario preparar los datos que se insertarán en la base de datos por medio de la capa de persistencia la cual se explica en el siguiente subcapítulo. La manera de realizar este procedimiento es registro por registro del documento tomando el eje X como valores de variables y encabezados de variables el eje y. (Fig. 12)

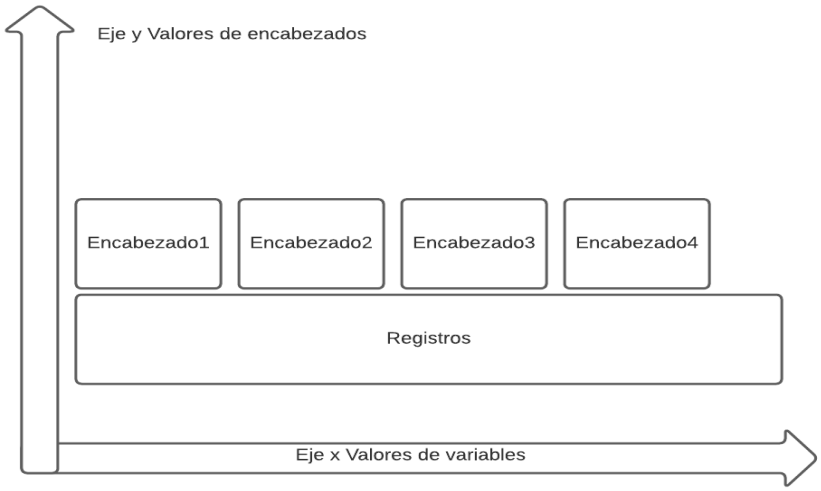


Figura 12. Orientación de los datos. Elaboración propia

En cada iteración se busca cambiar datos de fechas, datos nulos y patrones de dígitos. Se hace referencia que el archivo no cuenta con tipos de datos aun por lo que es necesario tomar las cadenas y comparar con los patrones de búsqueda.

- Datos de fechas: se buscan datos que tengan formato 9999-99-99 es decir, año, mes y día, estos deben ser diferentes a la máscara para que la base de datos acepte su inserción. Los campos evaluados son: *FECHA_INGRESO, FECHA_SINTOMAS, FECHA_DEF. (Véase tipos de datos de archivo csv)*
- Datos nulos: son aquellos datos que no tienen un valor asignado incluyendo el valor cero como punto de comparación. Nombre de datos evaluados: *ID_REGISTRO.*
- Patrones de dígitos: la cadena de caracteres es procesada con una función de la librería de *Python* llamada *isdigit()*, esta se encarga de evaluar la cadena y devolver un falso o verdadero si este resulta número. Los campos evaluados son: *MUNICIPIO_RES.*

De esto resulta necesario decir que los datos están listos para insertarse en la base de datos, se genera una plantilla de tipos de datos por cada uno de los encabezados, los tipos de datos a utilizar son: INTEGER, DATE y TEXT (entero, fecha y texto). En este proceso es cuando se asigna en la base de datos los tipos que llevaran a lo largo del proceso para ser evaluados y consultados.

4.7 **Descripción Fase 3 y 4. Persistencia y data warehouse**

La persistencia es la capa donde se interactúa físicamente en la memoria de disco, como lo son los procesos de lectura, escritura y borrado de datos u objetos. El plugin tiene que crear las plantillas de las tablas que conformarán la base de datos, también se realizan consultas de los datos ya procesados, es necesario resaltar que hasta este punto existen datos escritos y leídos. A fin de apoyar la estructura de datos también interviene una función llamada *data warehouse (almacén de datos)* la cual sirve para separar en términos y características los datos.

La fase de persistencia interactúa en varias ocasiones con toda la funcionalidad del plugin, es por esto que se decidió separar en funciones que pueden ser activadas en cualquier momento, para ejemplificar lo anterior mencionado se usará data *warehouse* como tema aparte y este hace llamar funciones de la persistencia en múltiples ocasiones. (Fig. 13)

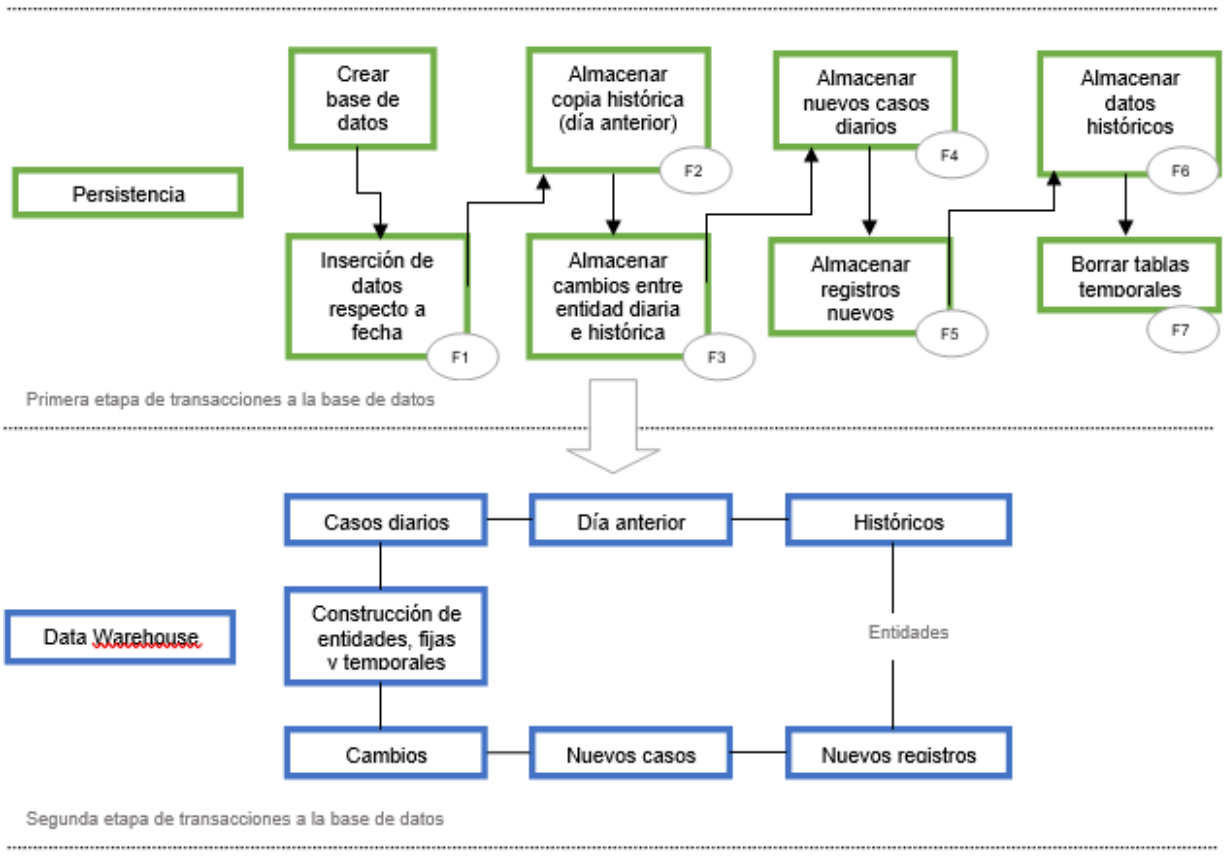


Figura 13. Diagrama de interacción entre persistencia y data warehouse. Elaboración propia.

4.7.1 **Base de datos en PostgreSQL**

La base de datos que se utiliza para almacenar los datos es *PostgreSQL*, las razones de esto están explicadas en el capítulo de estructura de base de datos. PSQL es una base datos relacional orientada a objetos y pertenece al género de open source (Group, 2021). La primera acción a realizar es generar las tablas que se utilizarán como núcleo, entre ellas están los catálogos o diccionarios que el servicio de la federación expone, así como el vaciado completo del archivo separado por comas y el histórico. El sistema tiene un módulo de persistencia que cuenta con un script el cual carga las entidades a utilizar.

- Catálogos o diccionarios: Son archivos separados por comas que solo requieren ingresarse a la base de datos una sola vez ya que se compone de los valores únicos de cada encabezado, cuentan con la descripción del mismo (Figura 14).
- Vaciado histórico de entidad *"historia":* Se refiere a la tabla en base de datos que contiene la totalidad de la información existente hasta el último día de actualización.
- Tablas temporales
- Entidad de cambios, tabla que almacena los cambios en las variables de cada registro.
-

clave	descripcion
1	CRUZ ROJA
2	DIF
3	ESTATAL
4	IMSS
5	IMSSBIENESTAR
6	ISSSTE
7	MUNICIPAL
8	PEMEX
9	PRIVADA
10	SEDENA
11	SEMAR
12	SSA
13	UNIVERSITARIO
99	NO ESPECIFICADO

Figura 14. Ejemplo de catálogo de sectores. Elaboración propia.

4.7.2 **Funciones en persistencia y data warehouse**

A partir de entonces los procesos de catálogos solo se ejecutan una vez, garantizando que las actividades externas solo utilizaran funcionalidad apegada a la persistencia. Por ejemplo, se tiene el proceso de insertar datos correspondientes a la fecha en la que el archivo es leído.

Este proceso puede invocarse las veces que sean necesarias. A las actividades anteriormente comentadas se les dio el nombre de funciones de persistencia, las cuales son activadas y desactivadas dependiendo el proceso que las necesite. El método de data warehouse es aplicado en las funciones de persistencia, el motivo es que reside la verdad de la información en esta capa. Data warehouse se encarga de transformar los datos de una fuente de información en distintas entidades que le darán sentido de importancia a los datos. Por este medio es posible consultar históricos, variantes de tiempo y datos resumidos que, dentro del contexto epidemiológico, aportan variables de alta prioridad.

Para segmentar los datos es importante tener en cuenta un tema de estudio. La transformación de los datos en entidades depende que aporte valor al tema. En esta etapa es posible ignorar información que no aporta conocimientos. La entidad es definida como los datos segmentados que pertenecen al contexto de estudio.

4.7.2.1 Primera función de persistencia, datos de fecha

Al obtener distintos datos en la iteración del plugin, es necesario manipular por separado los datos más importantes que residen en él. Uno de los datos más importantes es el de fecha, este es un sello de tiempo en que ocurren los sucesos y estos pueden ser medidos o estudiados a partir de este. El documento diario contiene algunas fechas, pero al mismo tiempo carece de la fecha en que los datos fueron escritos (tabla 2).

Tabla 2. Datos que se manipulan en el proceso de escritura a la base de datos.

Datos Transformados. Elaboración propia

Encabezados	Descripción
Fecha Actualización	El valor de este encabezado refiere a la fecha en la que el documento fue actualizado en su totalidad.
Fecha de ingreso	Fecha en la que el registro fue ingresado en algún sector salud.
Fecha Síntomas	Fecha en la que el registro presentó los síntomas de la enfermedad COVID-19
Fecha defunción	Fecha en la que el registro falleció por la afección de la enfermedad.

En el instante de que un objeto requiere asignar fecha o guardar fecha para una comparación dentro de la iteración, la función extrae la fecha del documento y esta es capaz de ser insertada a cualquier objeto, esta *actividad* ayuda a crear nuevas tablas en la base de datos que guardan información compresa a la hora de consulta (Figura 15).

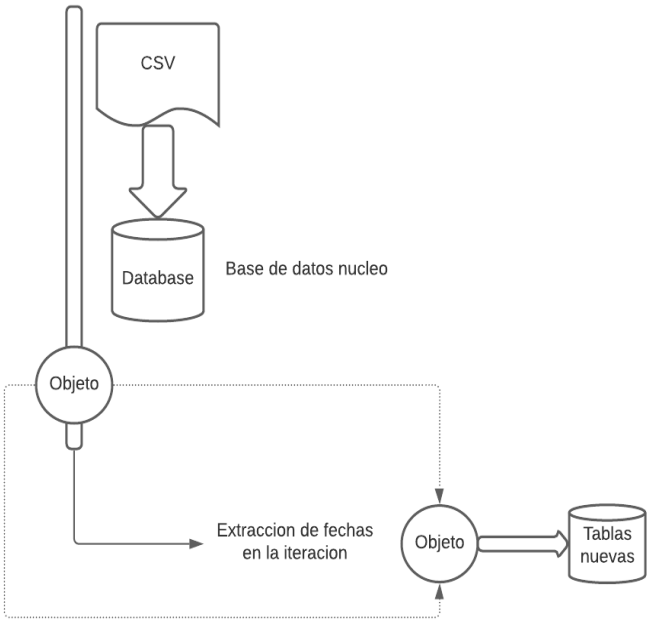


Figura 15. Proceso de extracción de fechas por medio de iteraciones. Elaboración propia.

4.7.2.2 Segunda función de persistencia, copiar datos del día anterior

Para realizar operaciones más complejas es necesario la comparación de información a través del tiempo, esto es posible cuando las fechas están asignadas correctamente en las tablas de consulta. La información ocurre con periodicidad diaria como se comentó en el capítulo de servicio de plugin, es decir siempre se hace referencia a un único archivo, este es actualizado en su totalidad dependiendo el día en el que se consulte. Dicho esto, es necesario implementar una función de escritura de datos como histórica de un día anterior, este proceso da como resultado el total de nuevos registros que se tienen de archivo a archivo (Figura 16).

Los datos diarios son almacenados en una tabla llamada *new_data*, esta contiene la totalidad de datos del día en que se evalúa la información, seguido de esto se crea otra tabla llamada *old_data* que como caso contrario mantiene los datos del momento anterior consultado, es decir un día anterior a la fecha de evaluación. El resultado es tener dos entidades con referencia en el tiempo para agilizar el análisis de la información. Con el mismo proceso se realiza la escritura de otra tabla llamada "*historia"* que contiene la totalidad de datos existentes.

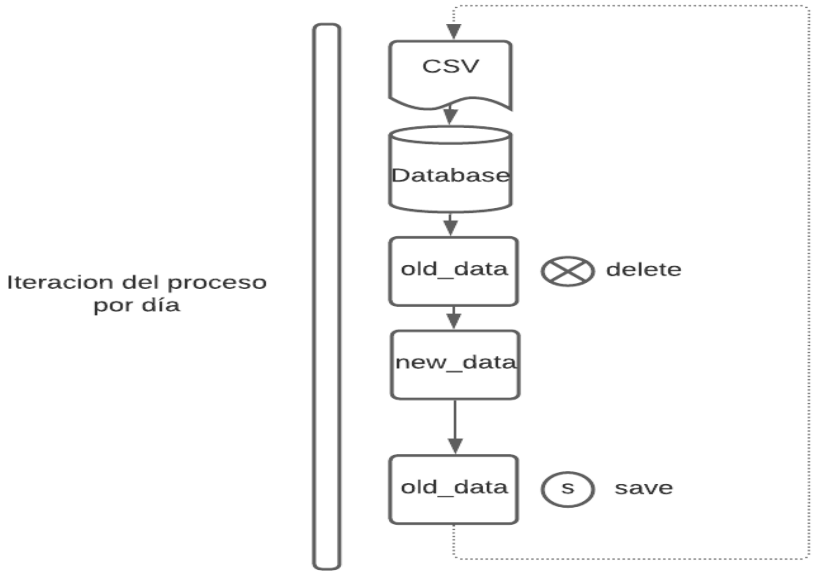


Figura 16. Descripción del ciclo diario. Elaboración propia.

4.7.2.3 Tercera función de persistencia, datos de cambios en tablas

Como se explicó en la descripción y alcances del plugin, es necesario contar con datos que el servicio no nos brinda, tal vez el más importante es el dato del *timestamp* (marca de tiempo). Este es capaz de decirnos en qué momento uno de los valores fue asignado. La función tercera de persistencia busca establecer un camino en el tiempo, dicho de otra manera, qué datos cambiaron en el archivo de ayer con el actual. Existen cerca de 38 variables en el *dataset* que nos caracteriza el comportamiento de un registro.

El procedimiento crea una comparación de las variables de *old_data* entre *new_data*, registro por registro y crea una condición de cambio asignando un valor booleano en donde encuentre diferencia (tabla 3). Como resultado se tiene una tabla con el total de variables representado por ceros y unos.

Tabla 3. Valores de aceptación en entidad de cambios.

Descripción de Variables en Entidad “Cambios”. Elaboración propia

Valor	Descripción
Cero	El resultado de la comparación no encuentra diferencias
Uno	El resultado de la comparación encontró diferencia

4.7.2.4 Cuarta función de persistencia, escritura de registros nuevos

En cada iteración del sistema, se extraen segmentos de datos. Estos se transforman en entidades que proporcionan efectos positivos con respecto a velocidad y robustez. La cuarta función toma criterios de comparación entre entidades ya almacenadas en la base de datos y esta se enfoca en los registros no existentes respecto al histórico, también crean una entidad de registros nuevos diarios. Los registros nuevos son considerados aquellos que no existen en ninguna entidad en la base de datos. La importancia de esta función recae en el seguimiento de las características de los nuevos registros recuperados diariamente. Por medio de esta funcionalidad se pueden analizar variables que resaltan en valor de tiempo.

4.7.2.5 Quinta función de persistencia, cuantificación de datos

Cada vez que los registros se escriben en las entidades correspondientes, se realizan conteo de datos. Estos números son registrados en entidades de casos diarios por número de registros nuevos y cantidad de variables que se solicite. Con el resultado de la función es posible realizar análisis en series de tiempo u otro tipo de estudio.

4.7.2.6 Sexta función de persistencia, entidad de datos históricos

Esta función realiza la escritura de los datos recabados diariamente. Estos previamente fueron manipulados y extraídos para su almacenamiento. La entidad que se ocupa del almacenamiento de los datos es llamada “historia”, contiene los datos que han sido expuestos hasta el momento. Para agilizar las consultas fue necesario agregar varios índices a las variables de la tabla ya que, la cantidad que almacena supera los trecientos millones de registros. La intención es que se puede extraer cualquier cantidad de registros en el momento que se desee. Esta permite la extracción por mes (véase capítulo de servicio), por lo que configurar el índice por este parámetro fue fundamental. La estructura permanece igual al del archivo que es descargado diariamente por el URL de la federación, esto sirve como referencia para indicar que la información no ha sido manipulada y permanece exactamente igual.

4.7.2.7 Séptima función de persistencia, depuración de tablas temporales

Existen dos formas de asegurar la información que recae en el sistema, diaria e histórica. La primera contempla los datos totales copiados directamente a una entidad llamada *new_data*, esta almacena datos pre procesados para posteriormente ser analizados o distribuidos en las demás entidades. Por otra parte, el contexto histórico se subdivide en dos, historia como se explicó en el subcapítulo anterior y entidad de *old_data.* Esta última resguarda la información

de la última vez que el sistema realizó una iteración completa. También es posible asegurar que cuando el archivo no es expuesto un día, es decir, la federación no subió el recurso, la información no se sobrescriba en la base de datos, ya que la entidad contiene la misma información que el día anterior o momento.

4.8 **Descripción fase 5. Finalización**

El sistema núcleo como ya se mencionó funciona de forma iterativa ya sea en documentos o registros. Existen dos formas de dar por finalizada la iteración. La primera está relacionada con los registros históricos. Cuando el sistema termina las n ejecuciones por día asignado, este sale del ciclo y da por finalizado el proceso de recuperación de datos. El segundo momento es el más usual. Ocurre cuando se terminan las funciones de registros diarios. Cabe mencionar que el proceso continua en automático hasta las siguientes 24 horas. En ambos escenarios los datos se encuentran listos para ser consultados por medio del servicio.

4.9 **Descripción de procesos del servicio**

Después que ha sido finalizado el proceso de almacenamiento de datos en el sistema, se necesitó crear un servicio el cual, consulta la información de la bases de datos. Devuelve la información solicitada en formato JSON. En este capítulo se describe la estructura del API y los métodos disponibles por medio de REST.

Se deben de proponer los recursos que se esperan al realizar peticiones al servicio. Para el trabajo de investigación se seleccionó el formato JSON al momento de solicitar los datos. La razón es que, este es compatible con la mayoría de los sistemas y en caso de no ser así, se pueden desarrollar clientes para consumirlo.

Al contar con datos históricos almacenados de gran volumen, es necesario requerir la información por bloques. El proceso se apoya de variables de fecha para dividir las consultas, en otras palabras, se pueden realizar las peticiones por fecha.

El servicio fue diseñado para que no represente problemas al agregar nuevos endpoints (urls que reciben o regresan información de web API) o URL en caso de que se requiera . De forma de prototipo, el servicio cuenta con consultas básicas, desde históricos hasta datos diarios actualizados. La intención es que este sea utilizado por personas con perfil de computación y no represente el mayor reto agregar nueva funcionalidad.

La figura 17 representa el intercambio de funciones al momento de realizar peticiones. El cliente realiza una llamada por medio de URL con encabezado GET. Las URL consumen los recursos del API, estas se conectan a la base de datos por medio de objetos con consultas *SQL.* Para devolver la información es necesario particionar la información en lotes (Batch). Este método permite procesar grandes cantidades de datos en un solo momento a diferencias de los procesos en tiempo real que pueden resultar tardados conforme el volumen de la información (bmc, 2021).

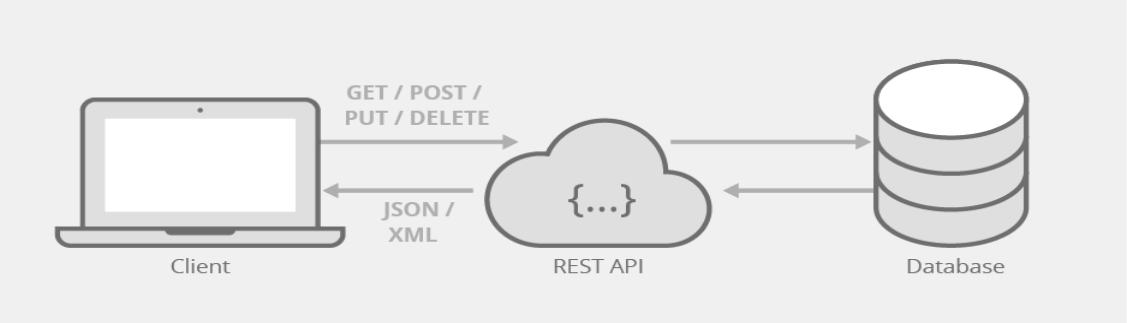


Figura 17. Diagrama API REST

El *API* está desarrollado en FLASK, *framework* que pertenece al lenguaje de *Python*. Se utilizó *Blueprints* como arquitectura, esperando que en el futuro se ampliara con los módulos de desarrollo web. *Blueprint* no es como tal una aplicación, esta debe ser registrada en *FLASK* para utilizar los componentes que la conforman (Python, 2021).

El blueprint que se utilizo fue nombrado “covid”. Este contiene varios componentes, pero solo se hace mención al que fueron utilizados:

- *Static_folder:* folder donde se encuentra los archivos estáticos del blueprint.
- *url_prefix*: la ruta para utilizar recursos del blueprint.
- *Subdomain*: es el nombre otorgado para la compatibilidad de rutas.

El API está implementado físicamente en el servidor donde está alojado el sistema núcleo en el puerto 5000, es servido por un *WSGI HTTP* para *UNIX* llamado *gunicorn* (gunicorn, 2021). La conexión se hace por medio de un web server *NGINX*.

Se decidió utilizar dichas herramientas por el bajo uso de memoria y alta ocurrencia. Maneja un solo hilo para las los eventos de solicitudes con enfoque asíncrono.

4.9.1 **Endpoints o URLs**

A continuación, se describen los endpoints implementados:

http://148.231.90.8/covid/nuevos/(month)

Este método devuelve los datos en formato JSON de la entidad “nuevos”. Los datos pertenecen a los registros nuevos que existen de un día a otro. Los registros se encuentran separados en lotes. Recibe como parámetro el mes que se requiere consultar. Debe ser tipo numérico anteponiendo el cero para los meses menores a diez. Ej. month=06

http://148.231.90.8/covid/cambios/(month)

Este método devuelve los datos en formato JSON de la entidad “cambios”. Los datos pertenecen a los registros que han cambiado su variables cuando se comparan las entidades de *new_data* y *old_data*. Los registros se encuentran separados en lotes. La información se distingue por que devuelve el nombre de la variable con valores de ceros y uno. La ruta recibe como parámetro el mes que se requiere consultar. Debe ser tipo numérico anteponiendo el cero para los meses menores a diez. Ej. month=05

http://148.231.90.8/covid/new_data

Este método devuelve los datos en formato JSON de la entidad “new_data”. Los datos pertenecen a los registros nuevos diarios expuestos por la federación. Los registros se encuentran separados en lotes.

http://148.231.90.8/covid/old_data

Este método devuelve los datos en formato JSON de la entidad “old_data”. Los datos pertenecen a los últimos registros almacenados en la base de datos. Los registros se encuentran separados en lotes.

http://148.231.90.8/covid/historia/(month)

Este método devuelve los datos en formato JSON de la entidad “historia. Registros históricos, es decir desde el primer archivo descargado el 12 de abril del 2020 .Los registros se

encuentran separados en lotes. Recibe como parámetro el mes que se requiere consultar. Debe ser tipo numérico anteponiendo el cero para los meses menores a diez. Ej. month=06

http://148.231.90.8/covid/casosdiarios1

Este método devuelve los datos en formato JSON de la entidad “registros”. Los datos pertenecen al resultado de la resta de *“new_data”* y *“old_data”*. Devuelve la fecha de la operación y la cantidad de registros nuevos.

http://148.231.90.8/covid/estados

Este método devuelve los datos en formato JSON de la tabla “estados”. Los datos pertenecen al catálogo de estados entregado por la federación.

http://148.231.90.8/covid/municipios

Este método devuelve los datos en formato JSON de la tabla “municipios”. Los datos pertenecen al catálogo de municipios entregado por la federación.

http://148.231.90.8/covid/sector

Este método devuelve los datos en formato JSON de la tabla “sector”. Los datos pertenecen al catálogo de sector donde fue hospitalizado el paciente, entregado por la federación.

http://148.231.90.8/covid/resultado

Este método devuelve los datos en formato JSON de la tabla “resultado”. Los datos pertenecen al catálogo de resultados de pruebas para COVID-19 entregado por la federación.

http://148.231.90.8/covid/tipo_paciente

Este método devuelve los datos en formato JSON de la tabla “tipo_paciente”. Los datos pertenecen al catálogo de tipos de pacientes entregado por la federación.

4.10 **Herramientas**

Se utilizaron varias herramientas en la construcción del servicio para el apoyo a los procesos de lectura en la base de datos. Fue necesario aplicar técnicas de manipulación de datos como

la separación de lotes y se utilizaron librerías para lograr el objetivo de servir la información. A continuación, se presenta una tabla resumiendo las librerías y su funcionalidad utilizadas (tabla 4).

Tabla 4. Herramientas utilizadas para API

Librería	Descripción
jsonify	Librería utilizada para la transformación de resultados de query en formato JSON
sqlalchemy	Transforma los querys en objetos relacionales para realizar peticiones u obtener resultados.
pandas	Librería que contiene herramientas para la manipulación y análisis de datos en lenguaje de programación Python.
psycopg2	Adaptador para base de datos PostgreSQL.

4.11 **Validación**

Para corroborar la funcionalidad del sistema es necesario implementar procesos que avalen la funcionalidad de los resultados esperados (Microsoft, 2021). Se utilizan pruebas unitarias para validar métodos en concreto del desarrollo. Las características que tienen las pruebas unitarias implementadas al sistema han sido ejecutadas independientemente del estado del sistema. Estas se llevaron a cabo con seis pruebas unitarias que se encuentran ligadas al contexto de investigación y respetan los objetivos anteriormente propuestos. La ejecución de cada una de ellas no afecta a la otra, es decir son pruebas aisladas y con parámetros de tiempo definidos. Otro de los objetivos es identificar que lo que se realizó como proyecto, es de calidad y está dentro de magnitudes de tiempo reales. El resultado deseado es un valor positivo, pero en caso de ser lo contrario, indicaría un punto a favor de cambios y refactorización del código. Las pruebas utilizadas se dividen en obtención del servicio, gestión de datos, almacenamiento de datos y extracción de datos. Cada una contiene las actividades realizadas y las herramientas que se utilizaron para su obtención, además de algunos ejemplos como resultados.

La mayoría de las pruebas unitarias se hacen comparando información directa de la base de datos con consultas externas. Se realiza a través de plataformas informativas como lo son CONACyT y UNAM. Ambos exponen los datos diarios de COVID-19 además de otros valores epidemiológicos. Las plataformas cuentan con metodología documentada, gracias a ello se pudo realizar dichas pruebas y saber con exactitud el valor esperado.

4.11.1 **Caso de prueba 1. Extracción desde el servicio y almacenamiento**

Es importante corroborar que los datos que se expiden por el gobierno federal con respecto a COVID-19 México, sean los mismos que se descargan y almacenan en la base de datos. Por lo que se planteó revisar el archivo de un día en específico y calcular la cantidad de registros que contiene. Después compararlo con el valor que se obtiene en la base de datos. Otro factor a contemplar es el tiempo de respuesta de ambos.

4.11.1.1 Descripción

Fecha de prueba: 21 de febrero del 2021

- Descarga de archivo: Se hizo por medio de un script en Python el cual solicita el valor de fecha del documento. La función devuelve el archivo en formato ZIP, en caso de que se haya encontrado.

- Descomprimir archivo: Se descomprimió el archivo con la codificación que se requirió, en este caso fue LATIN1. Como resultado se obtuvo un archivo de tipo separado por comas (CSV)

- Conteo de registros: Cuando el archivo CSV supera los 4 GB. este no puede ser abierto con algún programa externo, por ejemplo, Excel. Por lo que se transformó en dataframe, tipo de dato de la librería Pandas en Python. Posterior a esto, se contaron los registros del dataframe, también se realizaron estimaciones de duración del proceso.

- Conexión directa a la base de datos: Para realizar la conexión fue necesario construir una función, la cual toma como parámetros el hostname, username, password, database name. Estos tienen que corresponder con los valores previamente configurados en el servidor donde se aloja la información COVID-19 México.

- Consulta SQL: una vez que la conexión de la base de datos es establecida, se realizó una consulta SQL por medio de un driver PSQL y se estimó el tiempo que se demoró.

4.11.1.2 Herramientas

Se utilizó como plataforma de desarrollo Jupyter notebooks y Azure notebooks. Python fue el lenguaje de programación para realizar la mayoría de los procesos. También se requirió utilizar librerías como Pandas, esta es utilizada para manipulación de datos. Request, librería que

permite descargar recursos por medio de URL. Pycopg2 realiza la conexión con la base de datos tipo PostgreSQL. Funciones Time para evaluar el tiempo de los procesos y para las consultas se utilizó SQL.

4.11.1.3 Resultados

Se encontró que el total de los registros para el 20 de febrero del 2021 correspondió con el total de datos almacenados. Las comparaciones que se llevaron a cabo fueron: entidad histórica, entidad diaria y catálogos.

4.11.2 **Caso de prueba 2. Comparación de variables con una muestra mensual**

Una vez que se obtiene que los datos expuestos son los mismos a los que se almacenan en la base de datos, es importante realizar peticiones individuales por variables. Esto quiere decir que es posible seleccionar un atributo de alguna plataforma y corroborar que se tiene la misma cantidad siempre y cuando la metodología para calcular sea idéntica. En esta prueba se tomó un mes como histórico para validar que los datos corresponden. Además, se calcularon variables de positivos en hombres, mujeres, contagiados y negativos.

4.11.2.1 Descripción

Fecha de prueba: mayo 2020

- Descarga de archivo: Se descargo el histórico del mes de mayo por lo cual represento un mayor consumo de recursos por la cantidad de registros que existen en un mes.
- Tratamiento: El archivo paso por un proceso iterativo en el cual cada día transcurrido fue procesado por separado y posteriormente pegado a una variable de tipo dataframe.
- Conteo: se contaron los hombres y mujeres que tuvieran como valor de variable 1 en “resultado_res” y que estos no hubieran fallecido, información que se corroboró con la variable “fecha_def”. Se realizó el mismo procedimiento para los contagiados y negativos.
- Conexión: Se realizo una conexión a la base de datos para realizar consultas sobre la información que se tenía de mayo 2020.

- SQL: Se realizaron consultas para extraer variables con valor de conteo sobre los datos que se buscaban.
- Comparación: se realizó una comparación final de variables y se graficaron los resultados.

4.11.2.2 Herramientas

A diferencia de la caso de prueba 1, se utilizaron librerías de gráficas para mostrar los resultados obtenidos, también como se explicó la consulta de histórico requirió más recursos, por lo tanto, se diseñó una función para iterar sobre los datos históricos del mes en cuestión, además de la configuración que se realizó sobre las entidades en la base de datos PSQL.

4.11.2.3 Resultados

Todas las variables que se solicitaron tuvieron cien por ciento de exactitud. Los resultados fueron graficados para su fácil entendimiento. La primera grafica representa las cantidades de contagios en fecha de prueba (figura 18). Esta se representa en forma numérica y separado por dos variables, hombres y mujeres, al mismo tiempo es clasificada por resultados del archivo directamente y de la base de datos que se utilizó en el estudio. La segunda grafica sigue el mismo comportamiento, pero solo muestra la cantidad de defunciones de cada fuente de información (figura 19).

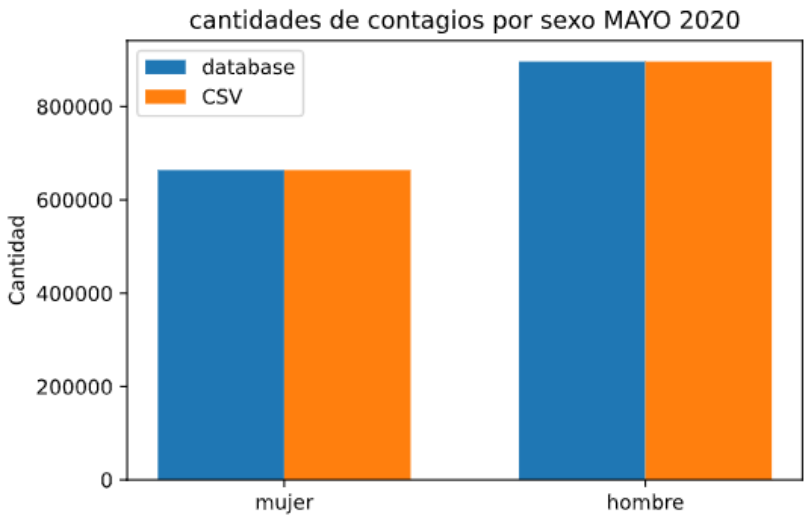


Figura 18. Gráficas de documentos vs base de datos. Elaboración propia

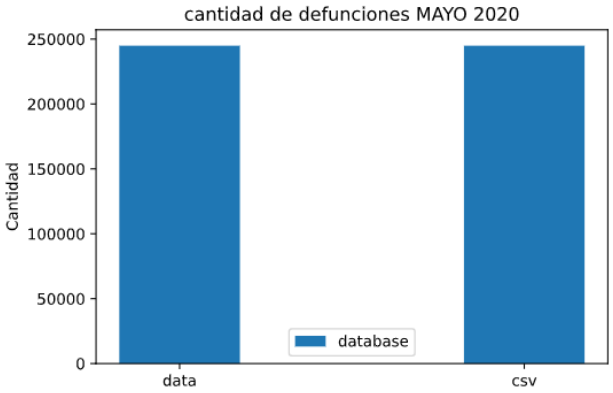


Figura 19. Gráfica de defunciones. Elaboración propia

4.11.3 **Caso de prueba 3. Verificar datos de archivo no expuesto**

Es importante demostrar que ocurre cuando la federación no expone un archivo de registros COVID-19 México. Según el histórico nos dice que, los días festivos o días feriados suele no existir el archivo del día en cuestión. Cuando esto ocurre, el sistema no duplica información ya que cuenta con una validación para esto, pero es importante describir que pasa cuando hay un vacío en fechas. Una vez que el servicio se reactiva el sistema realiza una comparación desde el último momento hasta el presente por lo que los datos faltantes solo se encuentran en el archivo final y este no cuenta con una variable que distinga en qué fecha se modificó el

registro. El experimento que se propuso es exactamente comprobar lo anterior, se espera que la diferencia de registros pertenezca al faltante.

4.11.3.1 Descripción

Fecha de prueba: 21 – 23 de marzo, día faltante 22 de marzo

- Conexión: Función para realizar conexión a la base de datos
- Identificar datos: se realiza un mapeo por la entidad de historia y se extraen dos variables que corresponden al 21 y 23 de marzo.
- Segmentación: Se realizó un proceso de segmentación por la cantidad de datos contenida en cada variable, este valor superó los 5 millones de registros. El tamaño del segmento se definió en un millón por iteración.
- Calculo: Una vez que se obtuvieron todos los registros de ambas variables, estas se restaron.

4.11.3.2 Herramientas

Se utilizo Azure notebooks para realizar la prueba unitaria. El desarrollo fue en Python y se utilizó Psycopg2 para la conexión a la base de datos. También fue necesario el uso de la librería Numpy la cual se emplea para los cálculos de variables y la segmentación y consultas SQL.

4.11.3.3 Resultados

El resultado de la resta de las variables del día 21 y 23 de marzo, no corresponde al día 22. Esto quiere decir que no es posible recuperar los días faltantes cuando el servicio de la federación falla o por cualquier razón no es expuesto.

4.11.4 **Caso de prueba 4. Entidad de cambios**

Como se explicó en el capítulo 5.2.9.3. en el sistema se colocó una entidad especial que mide los cambios en las variables a través del tiempo. El experimento propone tomar cien registros aleatorios del total de registros históricos y verificar el funcionamiento con cuatro variables epidemiológicas, fecha contagio, fecha síntomas, fecha recuperación y fecha defunción.

4.11.4.1 Descripción

Fecha de prueba: sin fecha

- Conexión: Se ejecutó la función que conecta la base de datos
- Consulta SQL: se realiza una consulta que toma cien registros aleatorios, después se crea una tabla temporal con estos valores.
- Iteración: se realizó una iteración sobre la tabla de los cien valores. Cada uno de los registros se buscó dentro de la entidad de “cambios”, la información se filtra para que cada id corresponda.
- Análisis: Se realizó un análisis de la información obtenida con filtro de las cuatro variables a verificar.

4.11.4.2 Herramientas

Se utilizo Azure notebooks, Python y SQL. Además, comandos batch para el mapeo de las tablas y su localización.

4.11.4.3 Resultados

Los resultados no fueron favorables. La cantidad de registros a comparar resulta en un alto consumo de recursos de cómputo lo cual genera lentitud en las consultas. Por otra parte, el cambio que han presentado los documentos de la federación con respecto a las variables ha impactado negativamente a este proceso.

4.11.5 **Caso de prueba 5. Comparación con plataformas informativas**

Como se comentó al comienzo de este capítulo, existen plataformas informativas dentro del territorio nacional. Estas exponen los datos de COVID-19 México en forma de resumen y graficas. Algunas cuentan resúmenes de análisis de la información presentada y la metodología que se utilizó para llegar a los mismos resultados.

Es importante comparar los datos que se tienen almacenados con otras plataformas informativas para mostrar credibilidad. Para este caso de estudio se seleccionaron las plataformas de CONACyT y UNAM por su impacto a nivel federal y el respaldo de sus expertos.

Los datos para comparar son los diarios y las variables activos, positivos y defunciones. La presentación es por estado y totalidad.

4.11.5.1 Descripción

Fecha de prueba: 31 de marzo del 2021

- Casos activos UNAM: Para realizar el cálculo de casos activos SARS-COV-2 se toma en cuenta aquellos que iniciaron sus síntomas con 14 días o menos y no fallecieron.
- Casos activos CONACyT: Los casos activos son todos aquellos positivos a SARS-CoV-2 con fecha de inicio de síntomas en los últimos 14 días. Las defunciones de casos activos se consideran parte de los casos activos, porque, desde una perspectiva poblacional, contribuyeron a la transmisión del virus. Se filtran todos los casos positivos (CLASIFICACION_FINAL valores “1”, “2” y “3”) registrados en la base de datos. Se cuentan los casos según fecha de inicio de síntomas (FECHA_SINTOMAS) y se consideran solo aquellos con menos de 14 días.
- Casos positivos CONACyT: Se filtran todos los casos positivos con CLASIFICACION_FINAL valores “1”, “2” y “3”
- Defunciones ambas plataformas: Se filtran todos los casos o registros por FECHA_DEF NOT NULL, esto significa que tiene un valor en la fecha de defunción.
- Conexión: se realizó la función para conectar a la base de datos PSQL.
- Consultas: Se realizaron consultas de casos activos, positivos y defunciones.
- Variables: se extrajeron tres variables que representan los valores buscados.
- Dataframe: Construcción de dataframe que contiene las variables extraídas y las variables a comparar de cada plataforma.

- Datos por estatales: Se realizó una consulta al catálogo de estados de la base de datos para extraer los id, después se realizó una operación SQL para extraer la información perteneciente a cada estado y colocarlo en datafreame.

4.11.5.2 Herramientas

Se realizo la prueba en ambiente Azure notebook. La información de variables epidemiológicas se extrajo directamente de los portales de CONACyT y UNAM. El desarrollo se realizó en Python con consultas SQL y para vaciar la información se utilizó la librería Pandas.

4.11.5.3 Resultados

Los resultados de las variables activos, positivos fueron cien porcientos aceptables. En el caso de sospechosos, UNAM no cuenta con esta información. Respecto a los valores de los registros por estado, hubo diferencias en los números, esto se debe a que no exponen en la metodología la semana epidemiológica que utilizan. La figura 20 refleja la información condensada del portal de CONACyT mientras que la figura 21 muestra los datos de la UNAM. En la esquina superior derecha se coloca la fecha de actualización para los datos que utiliza. Las variables más importantes se encuentran en el primer renglón y estimaciones en los siguientes. Se resaltaron en recuadros rojos aquellas variables que fueron comparados para esta prueba. En la figura 22 se encuentra el comparativo de tres variables respecto a las tres soluciones.

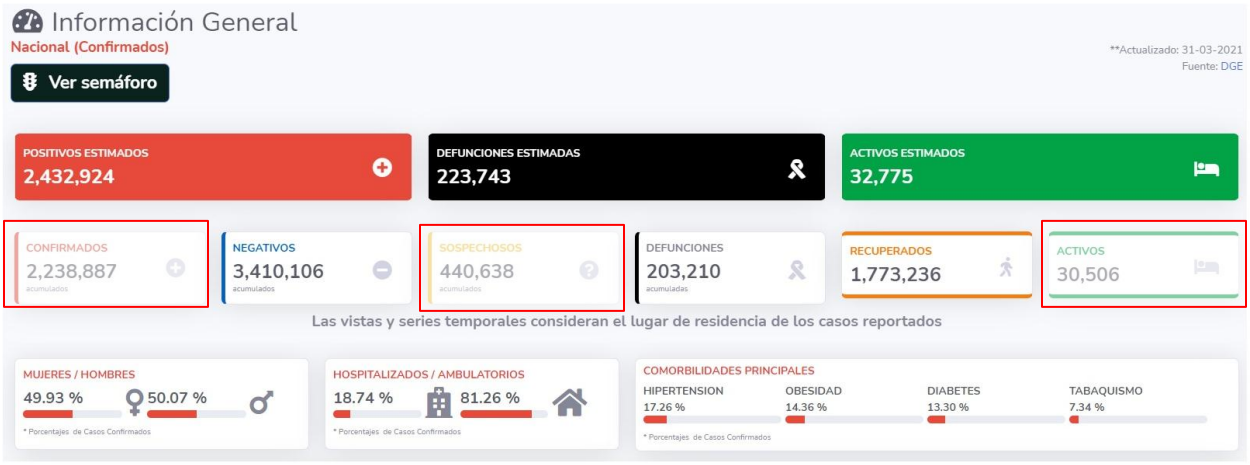


Figura 20. Información de plataforma CONACyT al 31 de marzo del 2021.

Figura 21. Información de plataforma UNAM al 31 de marzo del 2021.

	CONACYT	UNAM	DB
Activos	30506	30506	30506
Confirmados	223887	223887	2238887
Sospechosos	440638	0	440638

Figura 22. Comparativos entre plataformas y bases de datos.

4.11.6 **Caso de prueba 6. Servicio y base de datos**

Se tiene un servicio que proporciona la base de datos del sistema COVID-19 México, la intención del experimento es corroborar que la información entregada sea igual a la que está almacenada en la base de datos para cada uno de sus *endpoints*. Los datos a corroborar son: Cantidad de registros nuevos diarios, entidad de casos nuevos, entidad de cambios y diccionarios de datos.

4.11.6.1 Descripción

Servicio: Se realizaron peticiones al servicio en búsqueda de las variables propuestas para la prueba.

Dataframes: Con la librería *requests* de Python se almacenaron los datos devueltos en *dataframe*.

Conexión: Se ejecuto la función de conexión a la base de datos PSQL.

Consultas: Se realizaron consultas SQL referentes al mismo patrón de las variables solicitadas al servicio, posterior a esto fueron almacenadas en *dataframe*.

Comparación: Se realizó una comparación de los valores de los dataframe de servicio y los de las consultas *SQL*.

4.11.6.2 Herramientas

Se utilizó el ambiente Azure notebooks para la prueba. Se realizaron llamadas al API construido para consultar la información proveniente de la base de datos, esto con la librería requests de Python.

4.11.6.3 Resultados

Los datos del servicio resultaron cien por ciento 100% exacto con respecto a lo contenido en la base de datos.

Se considera que, con los resultados obtenidos de los casos de prueba, se acepta la solución de software y cubre los requerimientos plasmados en la metodología y a su vez con los objetivos específicos del trabajo de investigación.

Capítulo 5. Discusiones

El sistema núcleo COVID-19 México es una herramienta que posibilita la obtención de datos heterogéneos en contexto de variables epidemiológicas.

Si bien existen plataformas informativas que proporcionan dicha información, este sistema a diferencia de los demás otorga los datos crudos, es decir, sin procesar con metodologías de terceros o resumir y no está sujeto a una sola fuente de datos. Se encuentra en una capa anterior a estas plataformas. La metodología y arquitecturas fomentan los procesos de gestión de la base de datos. La plataforma de UNAM y CONACyT describen la metodología en la operación de los datos, sin embargo, excluyen la gestión de los mismos. Además, dichas plataformas entregan segmentos de datos por lo cual ya está trazado el camino para analizarlos.

El sistema desarrollado no es consumido por clientes, este aspecto se dejó abierto para permitirle al interesado flexibilidad en la obtención de datos por medio del servicio. Es posible que se requiera adecuar a algún sistema ya existente.

El proceso de descarga automática trajo consigo obstáculos que sobrepasar. El constante cambio en la estructura del documento se vio reflejado en distintas ocasiones. En consecuencia, fue necesario agregar bitácoras de control y módulos para que el sistema reflejara tolerancia al cambio. Se repitió el mismo escenario para los cambios de la dirección de descarga del servicio. Lo anterior es considerado como no automatizable ya que no es posible conocer la nueva dirección o variable del proceso, pero los controles permiten que se pueda actuar con rapidez.

De los resultados obtenidos en esta investigación, se pueden deducir varios aspectos entorno a la forma que se expone la información de COVID-19 en México. La información es recopilada con motivos de análisis y se excluye la importancia de la gestión de los datos, ojo no se menciona que la inexistencia. Los caso de pruebas demuestran que los datos obtenidos son los mismo que son almacenados y que es posible extraer segmentos de datos por medio de un servicio. En la prueba de la comparación de plataformas existe inconsistencia con ciertas variables, no todas. Al momento de comprobar las operaciones se encontró que no está explícito la metodología para el cálculo cada una de las variables si bien, aparecen las de mayor relevancia. Existe la posibilidad que las diferencias se deban al cálculo de las variables dentro en tiempos distintos de lo estipulado. La federación estipulo las semanas

epidemiológicas por medio de la Secretaría de la Salud y es posible que con motivos de análisis estos valores cambien en cada plataforma.

La intención de crear herramientas de software para la presente investigación, fue corroborar que los datos respetan la fuente de donde fueron adquiridos y estos no fueron manipulados para mostrar información errónea. Los sistemas cuentan con validaciones que resaltan la integridad de la información. Si los datos resultan íntegros , estos pueden ser utilizados por investigadores o cualquier persona que necesite hacer uso de ellos. El sistema núcleo sigue la metodología que consiste en manipular el set de datos de la federación, decodificar, formatear, transformar, segmentar, almacenar y exponer. Cada una de estas fases es explicada detalladamente, el cual significa se sigue el método científico y lo justifica. Desde el punto de vista de desarrollador, el sistema cuenta con arquitectura documentada, lo cual simplifica el entendimiento del mismo y deja abierto la posibilidad de expandir la solución.

Desde el comienzo fue necesario imponer los límites de la investigación. El trabajo fue centrado en la gestión de los datos, justo antes de su procesamiento. Esto para proveer un mecanismo simplificado de acceso a los sets de datos a través de un servicio y sin la complejidad que implica trabajarlos como los ha expuesto la federación.

Por otra parte, para fundamentar el presente trabajo fue requerido formular preguntas de investigación las cuales, fueron utilizadas para justificar la investigación. La primera pregunta cuestiona la existencia de portales que brinden información sobre COVID-19 en México y que estructura se utilizó. Al momento de investigar sobre las plataformas se dedicó tiempo al estudio del brote de la pandemia y como se convirtió propiamente en información. Como se ha mencionado existen diversas entidades que regulan los datos abiertos y datos epidemiológicos. A nivel mundial encontramos la OMS, la cual se encarga de divulgar información referente a la salud. Realiza varios procedimientos que avalan los datos expuestos y trabajan en conjunto con diversos países. Al ser un organismo con presencia mundial fue lógico que su postura representara los datos COVID-19 como portal informativo. La metodología que siguió fue recopilar información de cada uno de sus países inscritos al programa y exponer la información resaltando las variables con más impacto en la población. Posterior a esto, en lo individual los países participes comenzaron a crear sus propias plataformas con el fin de comunicar a la población el estatus o campañas elaboradas por los respectivos gobiernos. Este proceso no fue inmediato, tardo meses en construir dichos portales y esquematizar los datos. En México actualmente existen tres portales oficiales los cuales están respaldados por instituciones

científicas o equipos de profesionistas. En primer instancia resulto lógico construir un portal informativo gubernamental, el motivo es que les pertenece la fuente principal de información.
CONACyT consolido su portal, exponiendo una parte de la metodología que se aplicó para elaborar dicha información, donde explicó que variables han sido utilizadas para los cálculos de sus gráficas y análisis. También asignó las referencias bibliográficas y la descripción de sus colaboradores. Por otra parte, el portal de la UNAM también cuenta con gráficas y análisis de la información, pero escasea en la metodología aplicada.

La segunda pregunta de investigación toca el tema de la integridad de la información respecto al contenido de las plataformas, se cuestiona sobre la utilidad para personas sin perfil computacional. En el capítulo de validaciones se realizó una prueba unitaria que consistió en comparar distintas plataformas informativas con respecto al sistema desarrollado. Las variables de impacto como cantidad de contagios, activos, defunciones correspondieron correctamente por lo que, de esta forma se aseguró la credibilidad de los datos. No obstante, cuando se realizaron cálculos históricos por rangos de fechas hubo diferencias, esto se debe la diferencia en las metodologías de cada uno de ellos. Por otra parte, las plataformas informativas de uso para datos COVID-19 México tienen a su disposición información condensada o resumida por lo que, depende directamente el alcance de la investigación que se requiera realizar.

La tercera pregunta de investigación hace hincapié en las herramientas de almacenamiento. En la revisión de literatura existen múltiples formas de almacenar los datos. Se optó por las bases de datos relacionales porque se conocían previamente los datos y PostgreSQL por ser un manejador orientado a objetos. No se probó con distintos motores de bases datos ni tampoco se realizaron comparaciones de tiempos de ejecución. Desde un principio se decidió seguir este camino por la capacidad de cómputo y estabilidad del software. Se estudiaron las características de dichos manejadores y en base a eso se reafirmó la solución con PostgreSQL.

Durante el proceso de la investigación se encontraron varios obstáculos, a continuación, se mencionan los más importantes:
El servicio que utiliza la federación para exponer los datos es limitado y carece de estructura. Se descarga por medio de una URL el archivo que, contiene la totalidad de datos del día en curso. No existe hora de actualización, a lo largo del proyecto se observó que la subida de información ocurrió en distintos horarios, además, existen días que la información no estaba

disponible. Cabe mencionar que la información no está disponible en días festivos o de asueto por lo que, se infiere que el proceso depende del factor humano y carece de estructura.

Los primeros archivos contenían datos repetidos por lo cual se aseguró por medio del análisis de limpieza de datos, que la información no resultará incompleta y al ser un archivo separado por comas carece del tipado de datos correcto para cada variable en el documento.

El no contar con una estructura de servicios dificultó esta fase del desarrollo. Se concluye que faltó infraestructura para la exposición de los datos COVID-19 México.

En tema de procesamiento de datos, la obtención de los mismos por un archivo CSV condujo al leer las variables una por una así que, el tiempo de ejecución se extiende según la cantidad de datos contenidos en el archivo. Cada registro fue pre procesado antes de su inserción en la base de datos. Se tuvo que hacer una actualización de versión de Python ya que, la cantidad de registros requería toda la memoria cache del script. Esto se solucionó en la versión 3.8 de Python, ya que esta separa los datos por batch durante cada iteración.

El documento descargado carece de variable que ayude a registrar los cambios a través del tiempo por registro. Durante el proceso se creó TIMESTAMP que significa valor de tiempo asignado. La inexistencia de esta, hace imposible corroborar cuando es cambiado el valor de un registro en la bases de datos, sin embargo, fue creada en base a históricos en la presente investigación y pertenece a las entidades del data warehouse.

El portal de gobierno www.gob.mx en la sección de documentos de datos abiertos cuenta con históricos, pero estos se encuentran en archivos CSV y se accede a ellos por medio de URL que contiene el valor de fecha en su cadena, por lo que infiere la carencia de estructura antes mencionada.

La seguridad de la capa de persistencia es prioridad ya que el software está pensado para que distintas personas lo puedan utilizar siempre y cuando sea tema de soporte a la aplicación. En caso de ser necesario escalar el software para agregar nuevas entidades, no se registra bitácora de estos cambios por lo que resulta problemas en la integridad del sistema. Además, con un sistema de usuarios, se pueden agregar roles y permisos sobre la base de datos. El sistema fue desplegado en entorno de pruebas por lo que carece de configuración para pertenecer al esquema de producción.

El API contiene rutas de recursos limitadas, estas fueron creadas con propósito de consultar datos de diario e históricos. Al desarrollar el servicio se presentaron una serie de problemas respecto a la cantidad de información que se generaba en cada petición. Se resolvió en conjunto con configuración del gestor de bases de datos y un método de batch por hilo de lado del API, este separa los datos en cada iteración en cantidades de un millón de datos. Hubo

problemas con la cantidad de peticiones que se hacía al recurso, resulto en bloqueos de IP, sin embargo, las pruebas registraron que este comportamiento es exento al software y tuvo que ver con la configuración de firewall del servidor donde se alojó el servicio.

Capítulo 6. Conclusiones y Trabajo futuro

La presente tesis tuvo como objetivo la Gestión y obtención de datasets de registros COVID-19 México, eliminando procesos intermedios a través de plugin Sistema núcleo COVID-19 México en periodo de abril del 2020 hasta mayo del 2021.

El proceso para desarrollar lo anterior requirió que primero se realizara un análisis de la situación actual. Se observó que la fuente de datos es expuesta por la Secretaría de Salud por medio de un documento al cuál, se le van sumando los datos diarios de registros COVID-19 en México. Esto significó que los datos tenían una estructura tal que requiere un perfil de profesional especializado en la gestión de datos a través de la computación, por lo que resulta inaccesible para investigadores de distintas áreas a la computación; en especial considerando que para mayo del 2021 la cantidad de datos rondaba alrededor de los seis millones. El hecho de realizar la descarga diariamente de un archivo que al paso del tiempo va incrementando en tamaño de información se traduce en dificultad para trabajar con la información. Si bien existen plataformas de información que trabajan la misma base de datos, estas brindan la información de forma estática y resumida. Aunque la necesidad de análisis se ve solventada con gráficas y predicciones no resultan ser para el público en general, se debe contar con conocimientos estadísticos o de ciencia de datos.

Con esto se concluye que no existe un servicio que entregue los set de datos según lo requiera un perfil en específico. Al estudiar los componentes del proceso general de gestión, se ve representado en la importancia de contar con los datos en tiempo real y la posibilidad de realizar cualquier pregunta en forma de consulta al contexto de estudio. En consecuencia, se ve la complejidad de elaborar una herramienta capaz de lograr lo mencionado y también expone la cantidad de conocimientos que se deben aplicar para la solución del problema. En el tercer capítulo se describe la metodología que se siguió, esta brinda un acercamiento a los distintos análisis que se realizaron; desde el problema hasta las herramientas utilizadas. Siendo más específicos, se tuvo que hacer una extensa investigación en los distintas arquitecturas que soportaron los sistemas construidos, esto con el fin de crear un sistema que dependiera de la misma fuente de datos y contexto.

Se debe mencionar que el sistema no fue realizado para el uso directo por cualquier usuario, ya que para extraer sets de datos, es necesario contar con la asistencia de un profesional de la computación que pueda aplicar herramientas para generar un cliente y así una tercer persona

pueda acceder a los datos. Así, el profesional de la computación puede simplemente utilizar el servicio, hacer uso de los datos en formatos tales como JSON, XML o CSV a través del servicio y presentarlos en formatos accesibles al público, tales como gráficas, tablas interactivas, archivos para hojas de cálculo entre otros formatos. Como resultado de lo anterior se considera que el primer objetivo específico fue cumplido de manera satisfactoria.

Se utilizaron técnicas como data warehouse, el cual facilita la obtención de sets de datos en forma de entidades a partir del almacén de datos y por lo tanto resuelve el segundo objetivo específico.

Como un aspecto a resaltar, si en un futuro se decidiera implementar análisis de los datos, el sistema cuenta con los módulos para agilizar su elaboración.

El tercer objetivo específico consistió en "*Desarrollar un servicio el cual exponga la información de COVID-19 en México tal como se da a conocer por las entidades de salud en México. Está a su vez con la capacidad de crecer según las consultas requeridas".* Para ello se desarrolló el API con el cual se pueden construir endpoints para obtener sets de datos específicos a través de una simple llamada a un servicio mediante una URL. Las URL permiten recuperar sets de datos históricos y también tienen la capacidad de recibir parámetros para extraer segmentos de datos. Es decir, se puede consultar la entidad de "historia" que contiene todos los datos de COVID-19 México, filtrado por el mes que se solicite, por lo que reduce el tiempo de respuesta y la cantidad de datos devueltos; o bien recuperar sets de datos a partir de criterios distintos al tiempo.

La conclusión final es que el sistema de gestión realiza todas las operaciones para el correcto almacenamiento de la información manteniendo la integridad de los datos y proveyendo mecanismos simples para la extracción de datos en distintos formatos según sea requerido. De esta manera, se considera que el cuarto objetivo específico es llevado a cabo correctamente.

A continuación, me gustaría mencionar algunos puntos clave para la conclusión:

- Hasta ahora las plataformas de información existentes brindan información resumida provenientes de la fuente oficial. Esto crea oportunidades de mejora en la metodología que se utiliza para llegar a estos resultados.
- Aunque se puede considerar que la pandemia está llegando a su final, el sistema está listo para utilizarse bajo cualquier contexto en el que se generen datos y cuya

publicación siga el modo descrito en este caso de estudio empleando un solo archivo de forma periódica, esto debido a los cimientos tecnológicos con los que fue construido.

Como resultado de la investigación se tienen dos sistemas, uno para almacenar datos y otro para exportarlos, en el contexto del COVID-19 México, que están validados y listos para utilizarse por potenciales usuarios interesados: investigadores, periodistas, estudiantes, entre otros.

6.1 **Trabajo futuro**

El sistema tanto el servicio se pueden expandir según lo requiera las necesidades, es importante mencionar que lo anteriormente desarrollado no recae bajo el concepto de producto. El sistema núcleo COVID-19 México y API son considerados prototipos. A continuación, se enumeran las actividades que no se desarrollaron o que excedieron los límites establecidos.

6.1.1 **Diseño**

- Ampliar la configuración de persistencia para que acepte otros manejadores de bases de datos sin tanta inversión de tiempo y esfuerzo.
- Estructura de seguridad del manejador de bases de datos, segmentado por roles de usuarios y contraseñas.
- Sección de metadatos para los distintos manejadores de bases de datos y en la creación de entidades.
- Módulo de pruebas unitarias hacia los datos y la integridad del sistema.
- Módulo de llave *API*, donde se cree automáticamente el código del servicio para su consumo.
- Aplicación *Frontend* del servicio para mostrar los datos que sean requeridos hacia los usuarios.
- Gráficas, *dashboard* y curvas de frecuencia respecto a los datos COVID-19 México.
- Portabilidad del sistema núcleo COVID-19 México.

- Avisos de inicio o termino de operación del sistema por medio de correo electrónico o mensaje de texto a dispositivo móvil.

6.1.2 **Seguridad**

- Sistema de autentificación de usuarios tanto en el sistema núcleo COVID-19 como en el *API*.
- Bitácora de consumo recursos *API*. Donde se registren la cantidad de usuarios que han solicitado recursos.
- Usuario y contraseña para crear nuevas entidades y bitácora de usuario.
- Crear llave *API* la cual se solicita al sistema.

6.1.3 **Análisis de datos**

- Aplicar algoritmos para el análisis de la información con contexto estadístico.
- Machine Learning para la proyección, etiquetado y predicción de datos.
- Crear entidades de apoyo para métodos estadísticos.
- Comparación de curvas de contagio históricas para clientes finales.
- Integración con otras bases de datos con el mismo contexto epidemiológico, INEGI como ejemplo.
- Comparación con diferentes contextos, datos económicos, geográficos, industriales, etc.
- Adicionar datos ambientales para la eventual construcción de análisis de fenómenos compuestos. Por ejemplo, la ocurrencia de una inundación durante la pandemia.

IV. Referencias

Anselma, L. L. (2016). A Comprehensive Approach to 'Now' in Temporal Relational Databases: Semantics and Representation. *IEEE Transactions on Knowledge and Data Engineering Vol 28*, 24-28.

Bestougeff, H. D. (2013). *Heterogeneous Information Exchange and Organizational Hubs.* Springer Science & Business Media.

bmc. (15 de abril de 2021). *bmc blogs*. Obtenido de https://www.bmc.com/blogs/batch-processing-stream-processing-real-time/

Brown, M. E. (2021). Intra-COVID collaboration: Lessons for a post-COVID world. *Medical Education*, 55(1), 122-124.

Bustio-Martínez, L. C. (2013). Arquitectura basada en plugins para el desarrollo de software científico. *Conferencia Internacional de Ciencias Computacionales e Informáticas* (págs. 1-10). Habana: CICCI.

Carrión García, A. C. (2005). *Conceptos básicos de estadística.* Universidad Politécnica de Valencia.

Castro, L. L.-M.-A.-A. (2014). *Norma Mexicana para la Interoperabilidad entre Entornos para Objetos de Aprendizaje. Volumen 1. Marco teórico de la interoperabilidad entre entornos para objetos de aprendizaje.* Mexicali: Universidad Autónoma de Baja California.

Chouhan, P. S. (2015). Image Retrieval Using Data Mining and Image Processing Techniques. *International Journal of innovative researche in electrical, electronics, instrumentation and control engineering vol. 3*, 53-55.

Correa, J. M. (2010). uso de las plataformas digitales en las universidades de andalucía. *Linhas*, 60-74.

Cristiá, M. (2008). Introduccion a la arquitectura de software. *Technical reports*, 4-5.

D'Agostino, M. M. (2020). Estrategia para la gobernanza de datos abiertos de salud: un cambio de paradigma en los sistemas de información. *Revista Panamericana de Salud Pública* , 41.

Daniel Lemus-Delgado, R. P. (2020). Ciencia de datos y estudios globales: aportaciones y desafíos metodológicos. *Colombia Internacional*, 41-62.

DCC. (24 de abril de 2014). *Because good research needs data*. Obtenido de https://www.dcc.ac.uk/guidance/how-guides/five-steps-decide-what-data-keep

Elgendy-Nada, E.-A. (2014). Big Data Analytics: A Literature Review Paper. *Lecture Notes in Computer Science*, 1-2.

Elizabeth Woodward, S. S. (2010). *A Practical Guide to Distributed Scrum.* Pearson Education.

Fajardo-Gutiérrez. (2017). Medición en epidemiología: Prevalencia, incidencia, riesgo, medidas de impacto. *Revista alergia México*, 109-120.

Flores Castro, E. G. (2018). *Implementación de una base de datos heterogénea distribuida entre los SGBDs ORACLE, MySQL y PostgreSQL con replicación, mediante un script bash implementado en el sistema operativo CentOS usando software libre. INNOVA Research.*

García, S. L. (2015). *Data Preprocessing in Data Mining. Springer.* Springer International Publishing.

GobMX-a. (14 de marzo de 2021). *Datos Abiertos de México*. Obtenido de https://datos.gob.mx

GobMx-b. (25 de marzo de 2021). *Datos Abiertos de México*. Obtenido de https://datos.gob.mx

Google. (13 de febrero de 2021). *Archivo CSV: Definición—Ayuda de Google Ads*. Obtenido de https://support.google.com/google-ads/answer/9004364?hl=es-419

Group, T. P. (20 de 03 de 2021). *PostgreSQL: About*. Obtenido de https://www.postgresql.org/about/

gunicorn. (11 de mayo de 2021). *gunicorn*. Obtenido de https://gunicorn.org/

Gutiérrez, P. M. (2006). Data Warehouse marco de calidad. Universidad Carlos tercero.

Hallo, M. (2014). Bases de datos NoSQL. En I. L. Abiertos, *Tópicos avanzados de Bases de datos* (págs. 104-114). Proyecto Latin.

Hernandez, J. D. (2018). *Desarrollo de una plataforma de software para la gestión de datos climáticos en malla.* Mexicali: UABC.

hopkins, j. (14 de marzo de 2021). *coronavirus resource center*. Obtenido de https://coronavirus.jhu.edu/map.html

Inmon, W. H. (1 de 1 de 1999). *Building the Operational Data Store.* Obtenido de the data administrator newsletter: https://tdan.com/building-the-operational-data-store-2nd-ed/5552

Len Bass, P. C. (2003). *Software Architecture in Practice. 3 ed. Westfor, united states. pp. 25-38.* Addison-Wesley Professional,.

meter, w. o. (14 de marzo de 2021). *COVID-19 coronavirus pandemic*. Obtenido de https://www.worldometers.info/coronavirus/

México, S. d. (30 de OCTUBRE de 2021). *Información referente a casos COVID-19 en México*. Obtenido de Datos.gob.mx/busca. (s. f.).: https://datos.gob.mx/busca/dataset/informacion-referente-a-casos-covid-19-en-mexico

Microsoft. (12 de 04 de 2021). *Curso .NET con C#*. Obtenido de https://si.ua.es/es/documentacion/c-sharp/documentos/pruebas/07pruebasunitarias.pdf

Moreno-Altamirano, A. L.-M.-B. (200). Principales medidas en epidemiología. *Salud Pública de México, 42*, 337-348.

Muente-Kunigami, A. S. (2018). *Los datos abiertos en América Latina y el Caribe. Inter-American Development Bank.* Banco Interamericano de Desarrollo. Obtenido de https://doi.org/10.18235/0001202

Neumann, A. L. (2018). An Analysis of Public REST Web Service APIs. *IEEE Transactions on Services Computing*, 1-1.

NIH. (25 de abril de 2021). *Instituto nacional del cancer*. Obtenido de https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/sars-cov-2

Parra, R. J. (15 de 02 de 2021). *Análisis de series temporales*. Obtenido de Bookdown: https://bookdown.org/franciscoparrod/analisis_series/Analisis_Series.html

Powell, G. (2006). *Beginning Database Design, Indianapolis, Indiana pp. 6-14.* John Wiley & Sons.

Pyle, D. (1999). *Data Preparation for Data Mining, San Francisco, USA. pp 117-125.* Morgan Kaufman Publishers.

Python. (15 de Mayo de 2021). *realpython*. Obtenido de https://realpython.com/flask-blueprint/#:~:text=Each%20Flask%20Blueprint%20is%20an,is%20not%20actually%20an%20application.&text=This%20is%20the%20key%20concept%20behind%20any%20Flask%20Blueprint.

Rahul K Arora, A. J. (2021). SeroTracker: a global SARS-CoV-2 seroprevalence dashboard. *The Lancet Infectious Diseases, Volume 21, Issue 4*, 75-76.

Sadalge, P. (02 de 10 de 2014). *NoSQL Databases: An Overview.* Obtenido de ThoughtWorks: https://www.thoughtworks.com/insights/blog/nosql-databases-overview

salud, s. d. (14 de febrero de 2021). *Secretaría de Salud | Gobierno | gob.mx.* Obtenido de https://www.gob.mx/salud

Santamaría, R. (s.f.). *Opte.* Obtenido de Docplayer: http://vis.usal.es/rodrigo/documentos/sisdis/teoria/1-introduccion.pdf

sidn. (30 de 10 de 2020). *Estructura de una URL y buenas prácticas SEO*. Obtenido de Agencia digital | Agencia posicionamiento SEO Madrid: https://www.sidn.es/noticias/546-estructura-url-seo

Sommerville, I. (2005). *Ingeniería del software. Séptima edición. Madrid, España. pp 219-279.* Pearson Education.

Suárez, R. A. (2013). *Model de arquitectura de middleware para un sistema de transporte público de pasajeros en ciudades basado en Smart City.* Obtenido de udistrital.edu.co: https://repository.udistrital.edu.co/bitstream/handle/11349/8022/GomezSuarezRicardoAlfonso2016.pdf;jsessionid=34497F7C36C39A97CFB18B0878EC8E77?sequence=1

Umar, A. (2004). *Third Generation Distributed Computing Environments. nge solutions, inc.* nge solutions.

UNAM. (14 de marzo de 2021). *plataforma de informacion geaografica de la UNAM sobre COVID-19 en México*. Obtenido de https://covid19.ciga.unam.mx/

Velavan, T. P. (2020). The COVID-19 epidemic. *Tropical Medicine & International Health, 25(3)*, 278-280.

WHO. (14 de marzo de 2021). *https://www.who.int/es*. Obtenido de https://www.who.int/es

Wong, J. E. (20 de 02 de 2020). *COVID-19 in Singapore—Current Experience: Critical Global Issues That Require Attention and Action. , 1243. https://doi.org/10.1001/jama.2020.2467.* Obtenido de Jamanetwork: https://jamanetwork.com/journals/jama/fullarticle/2761890?resultClick=1

Glosario

Application Programming Interface: Interface de programación de aplicaciones. Punto de acceso de una aplicación. Es utilizada para que la aplicación que la ofrece se conecte con otras. Conjunto de métodos públicos que pueden ser llamados por código externo.

Backend: Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos.

Batch: Es una técnica que se aplica a los servicios para segmentar por lotes y permitir enviar datos segmentados.

Big data: Término utilizado en la comunidad científica y la industria para referirse a *datasets* de gran dimensión cuyo manejo resulta poco práctico para las herramientas convencionales actuales; se caracterizan por su gran volumen, variabilidad y velocidad.

Blueprint: Es un diagrama que visualiza las relaciones entre los diferentes servicios y componentes de un negocio— personas, lugares, objetos y procesos.

Dataframe: Es una estructura de datos con dos dimensiones en la cual se puede guardar datos de distintos tipos (como caracteres, enteros, valores de punto flotante, factores y más) en columnas.

Data Logger: Bitácora de datos.

Data Mart: Componente de una arquitectura típica de almacén de datos. Consiste en un modelo dimensional que contiene datos orientados a un tema o departamento en especial; agregados de tal forma que resulte sencillo la consulta de resúmenes y reportes.

Dataset: Colección o representación de datos residentes en memoria con un modelo de programación relacional coherente e independientemente sea cual sea el origen de los datos que contiene.

Framework: Marco de trabajo, librería de software o conjunto de ellas, diseñadas para facilitar el desarrollo de cierto tipo de aplicación al proveer componentes y herramientas para ello.

Frontend: Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

Hostname: Es el nombre de red de tu servidor.

Javascript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,2 basado en prototipos, imperativo, débilmente tipado y dinámico.

NoSQL: Término que normalmente se interpreta como "Not Only SQL" y es usado para referirse a los DBMSs con paradigma distinto al relacional. Por ejemplo, MongoDB y Cassandra.

***Open Source*:** Término utilizado para referirse a los proyectos de software que liberan el código fuente al público, en vez de mantenerlo cerrado o privado.

Pool: Conjunto de conexiones o de hilos utilizada en computación para administrar dichos recursos en una aplicación, con el propósito de optimizar su utilización y rendimiento.

Request: El objeto Request permite el acceso a toda la información que pasa desde el navegador del cliente al servidor. Al recibir esta información, es repartida y almacenada entre cinco colecciones.

Schedule: Planificación en unidad de tiempo.

String: En software se refiere a la cadena de caracteres que puede contener una variable u objeto.

Query: También conocida como search query, es una palabra, conjunto de palabras o frase que se utiliza como término de búsqueda en un buscador.

Anexos

Anexo A. Bitácora de versiones de sistema núcleo COVID-19 México.

Version	Description	Release
1.01	Engine descarga de datos covid19 del portal de secretaria de saludo. Acciones: Descargar y copiar template de datos a base de datos PostgreSQL.	**24-jun-20**
1.02	*Modificación de la estructura de código para monitorear el recolector de memoria, este causaba que no se pudiera pedir el total de consultas, marcaba problemas de memoria. solución: se cambió a 64 bits Python 3.8 donde hay una mejora en el recolector. se agregaron los catálogos adicionales a la base de datos para realizar las consultas entrelazadas Cambios al motor que realiza la descarga automática, se programó una bitácora en formato.txt para dar seguimiento al comportamiento de descargas.	**08-ago-20**
1.03	Se agregan los campos “indígena”, “resultado”.	**10-ago-20**
1.04	se hicieron cambios en el debugger para revisar un problema con timeout, después de los cambios si realizo la descarga correctamente y el storage en la base de datos.	**16-ago-20**
1.05	Solución de Encoding de archivo descargado	**20-ago-20**
1.06	Se modifico la arquitectura de la aplicación se agregó funcionalidad por capas. se creó una clase de logger que crea bitácora para debug, Se instalo en el server de UABC. Funciona correctamente -modo manual la ejecución.	**27-ago-20**
1.07	Proceso automático de descarga, configurado a las 23:00 todos los días.	**28-Aug-20**
1.08	cambios: Se guarda la base de datos row por row del csv revisando el tipo de dato y haciendo cambios en las fechas con 9999-99-99, en conclusión, postgres ya está con tipo de dato correcto.	**31-ago-20**
1.09	agregado tabla temporal y consultas SQL para hacer operaciones con las tablas, se descarga tabla daily, se compara con tabla old data, el resultado de ellas dos es guardar cuantos registros nuevos hay de una tabla a otra, la información se guarda en tabla de registros.	**07-sep-20**
1.1	se optimizaron SQL de conteo de cambios entre tablas, también se generó un SQL extenso que compara los cambios que se realizan en cada uno de los registros y lo almacena en una tabla llamada cambios.	**17-sep-20**

1.2	cambios en la tabla de cambios, se optó por dejar una tabla que tiene valores de 0 y 1 donde 1 es que se realizó el cambio. Esta tabla está separada por cada variable a diferencia del modelo anterior que concatenaba los valores.	**18-sep-20**
1.3	Modificación de estructura por cambio de federación, se agregan los campos: Indígena, toma muestra, resultado_lab, clasificacion_final.	**06-oct-20**
1.4	Modificación de estructura por cambio de federación, se agregan los campos: toma_muestra_antigeno, resultado_antigeno. Se modifica el nombre de toma_muestra por toma_muestra_lab.	**28-nov-20**

Anexo B. Bitácora de versiones API

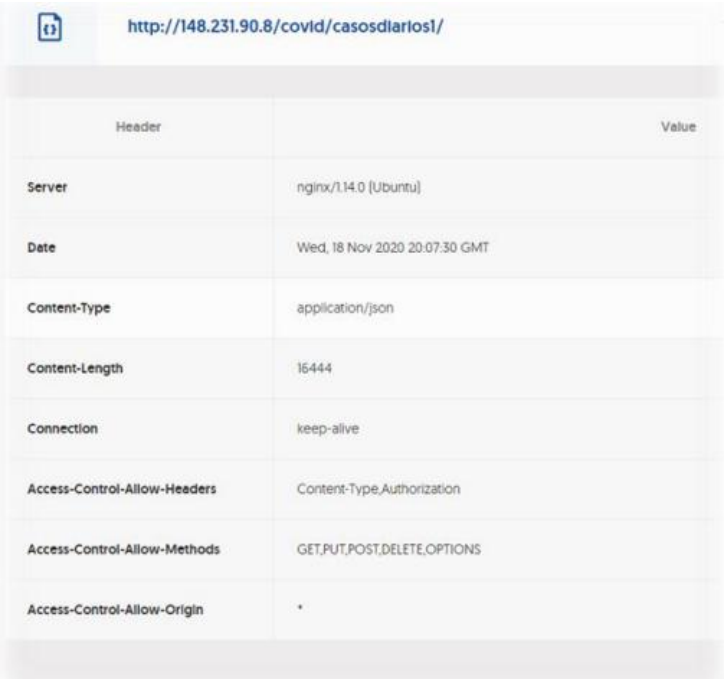
.

Version	Description	Release
1.1	Contiene la exposición de datos de registros diarios y catálogos de Estados, Municipios, Sexo, Tipo_paciente, Sector, Resultado. Desarrollo con ORM de SQLalchemy	**30-jun-20**
1.2	Cambios en estructura de consulta, se utiliza ORM para las tablas de la versión 1.1 y se agregan tablas de cambios, nuevos, historia, new_data, old_data	**25-jul-20**
1.3	Se agregan tipos de respuesta, tipos de datos en el formato Json, cambios en tiempos de respuesta junto con configuración de Server.	**16-sep-20**
1.4	Se modifica el nombre de toma_muestra por toma_mustra_lab y se agregan dos campos nuevos: toma_muestra_antigeno, resultado_antigeno	**28-nov-20**

Anexo C. API endpoints

#	URL	Description	Response	GET
1	http://148.231.90.8/covid/nuevos/	Datos de entidad “nuevos”	JSON	# month
2	http://148.231.90.8/covid/cambios/	Datos de entidad “cambios”	JSON	# month
3	http://148.231.90.8/covid/new_data/	Datos diarios crudos	JSON	-
4	http://148.231.90.8/covid/old_data/	Datos referentes a la última vez que se escribió en la base de datos	JSON	-
5	http://148.231.90.8/covid/historia/	Datos históricos desde el primer registro hasta el ultimo	JSON	#month
6	http://148.231.90.8/covid/casos_diarios/	Numero de registros nuevos diarios	JSON	-
7	http://148.231.90.8/covid/estados/	Datos de estados de México	JSON	-
8	http://148.231.90.8/covid/municipios/	Datos de municipios de México	JSON	-
9	http://148.231.90.8/covid/sexo/	Datos de sexo	JSON	-
10	http://148.231.90.8/covid/sector/	Datos de sector saludo en México	JSON	-
11	http://148.231.90.8/covid/resultado/	Datos de catálogo de resultados COVID-19	JSON	-
12	http://148.231.90.8/covid/tipo_paciente/	Datos de catálogo de tipos de paciente	JSON	-

Anexo D. Ejemplo de respuesta de API



http://148.231.90.8/covid/casosdiarios1/

Header	Value
Server	nginx/1.14.0 (Ubuntu)
Date	Wed, 18 Nov 2020 20:07:30 GMT
Content-Type	application/json
Content-Length	16444
Connection	keep-alive
Access-Control-Allow-Headers	Content-Type,Authorization
Access-Control-Allow-Methods	GET,PUT,POST,DELETE,OPTIONS
Access-Control-Allow-Origin	*

Anexo E. Tabla de estados y casos positivos

	count	clave_entidad	entidad_federativa
0	300	01	AGUASCALIENTES
1	330	02	BAJA CALIFORNIA
2	677	03	BAJA CALIFORNIA SUR
3	100	04	CAMPECHE
4	331	05	COAHUILA DE ZARAGOZA
5	84	06	COLIMA
6	92	07	CHIAPAS
7	914	08	CHIHUAHUA
8	11811	09	CIUDAD DE MÉXICO
9	298	10	DURANGO
10	1399	11	GUANAJUATO
11	491	12	GUERRERO
12	415	13	HIDALGO
13	679	14	JALISCO
14	3081	15	MÉXICO
15	388	16	MICHOACÁN DE OCAMPO
16	593	17	MORELOS
17	76	18	NAYARIT
18	714	19	NUEVO LEÓN
19	299	20	OAXACA
20	1165	21	PUEBLA
21	1104	22	QUERÉTARO
22	360	23	QUINTANA ROO
23	517	24	SAN LUIS POTOSÍ
24	474	25	SINALOA
25	603	26	SONORA
26	889	27	TABASCO
27	350	28	TAMAULIPAS
28	235	29	TLAXCALA
29	475	30	VERACRUZ DE IGNACIO DE LA LLAVE
30	554	31	YUCATÁN