Desarrollo de un sistema de gestión de base de datos mediante procesos ETL para la optimización de la información en el DirectorioUsme.com

Edwin Daniel Ceballos Sastoque

Asesora

María Alejandra Varona Taborda

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2025

Maria Alejandra Varona Taborda

Director de Trabajo de Grado

_____ _____

Jurado Jurado

2025

Dedicatoria

Al universo, a Dios y a mí mismo, por tener siempre claro lo que quería y por encontrar el camino para lograrlo.

Por la voluntad, la concentración y el compromiso diario que me permitieron llevar a cabo este proyecto.

Por cada noche larga acompañada de silencio, por esos días que alargué para poder cumplir con todo, sin importar que algunos no vieran mi esfuerzo.

Este proyecto no solo es académico, es profundamente personal.

Hoy sigo confirmando que toda meta que me propongo, la logro. Creo profundamente en mí, y sé que siempre tendré la energía, el flow y la determinación para demostrarlo y alcanzar muchas más metas.

Resumen

El proyecto aborda la dispersión de datos en el DirectorioUsme.com, donde la información de negocios, usuarios y métricas de rendimiento se encuentra fragmentada en Google Sheets y reportes manuales, lo que dificulta la toma de decisiones y afectaba la calidad de los datos. Para solventar esta situación, se diseñó e implementó un Sistema de Gestión de Bases de Datos (SGBD) centralizado en MySQL, reforzado con un proceso automatizado de ETL (*Extract, Transform, Load*). Dicho proceso extrae diariamente datos de fuentes internas (formularios de registro) y externas (Google Analytics, Search Console, AdSense), los transforma mediante limpieza, normalización de URLs y los carga en un esquema relacional normalizado que garantiza la integridad y evita redundancias. La metodología CRISP-DM orientó cada fase del proyecto: desde la comprensión del negocio y de los datos, pasando por la preparación y modelado, hasta la evaluación e implementación. Se crearon vistas, procedimientos almacenados y roles de usuario con permisos limitados para reforzar la seguridad y facilitar consultas especializadas. Finalmente, se integró Power BI para generar dashboards interactivos que permiten monitorear métricas clave, analizar el comportamiento de usuarios por categoría y zona, y detectar oportunidades comerciales en tiempo real. Este desarrollo no solo resolvió la necesidad inmediata de consolidación y análisis, sino que estableció una base escalable y documentada, lista para futuras mejoras como roles de usuario avanzados y análisis predictivo. La solución demuestra cómo herramientas de código abierto y arquitecturas modulares pueden transformar datos dispersos en un activo estratégico para el crecimiento sostenible del DirectorioUsme.com.

***Palabras clave*:** ETL, MySQL, Normalización, Procedimientos almacenados, SGBD.

Abstract

The project addresses data dispersion at DirectorioUsme.com, where business, user, and performance metrics information is fragmented across Google Sheets and manual reports, which complicates decision-making and affects data quality. To resolve this situation, a centralized Database Management System (DBMS) was designed and implemented in MySQL, reinforced by an automated ETL (Extract, Transform, Load) process. This process daily extracts data from internal sources (registration forms) and external sources (Google Analytics, Search Console, AdSense), transforms it through cleaning and URL normalization, and loads it into a normalized relational schema that guarantees integrity and prevents redundancy. The CRISP-DM methodology guided each phase of the project: from business and data understanding, through preparation and modeling, to evaluation and implementation. Views, stored procedures, and user roles with limited permissions were created to enhance security and facilitate specialized queries. Finally, Power BI was integrated to generate interactive dashboards that allow for monitoring key metrics, analyzing user behavior by category and area, and detecting business opportunities in real-time. This development not only resolved the immediate need for consolidation and analysis but also established a scalable and documented foundation, ready for future improvements such as advanced user roles and predictive analytics. The solution demonstrates how open-source tools and modular architectures can transform dispersed data into a strategic asset for the sustainable growth of DirectorioUsme.com.

***Keywords*:** DBMS, ETL, MySQL, Normalization, Stored Procedures,

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Introducción

En la actualidad, el manejo adecuado de la información estructurada en entornos digitales se ha consolidado como un factor fundamental para el crecimiento y la sostenibilidad de iniciativas tecnológicas. La capacidad de acceder a datos de manera oportuna, organizada y analizable permite optimizar procesos, facilitar la toma de decisiones y generar valor para los diferentes actores involucrados.

En este contexto, el sitio web *DirectorioUsme.com* ha sido realizado como un sitio web orientado a conectar a los habitantes de la localidad de Usme (Bogotá, Colombia) con los servicios, negocios y entidades públicas del sector. Su propósito principal es promover la visibilidad del comercio local y facilitar el acceso a información confiable en su entorno.

No obstante, en su estado actual, la plataforma presenta limitaciones relacionadas con la dispersión de los datos, la ausencia de un modelo estructurado de almacenamiento y dificultades para realizar análisis estratégicos. Esta situación afecta tanto la actualización como la validación y trazabilidad de la información, lo que repercute negativamente en la experiencia del usuario y en la capacidad de los gestores del sistema para sustentar sus decisiones en evidencia empírica. Frente a este panorama, el presente proyecto tiene como objetivo diseñar e implementar un sistema de gestión de bases de datos, orientado a consolidar, transformar y analizar la información recopilada de diferentes fuentes. Para ello, se adopta la metodología CRISP-DM, la cual proporciona una guía estructurada para la minería de datos (Chapman et al., 1999).

Asimismo, se integran herramientas tecnológicas como Python (mediante el entorno Jupyter Notebook), Google Sheets, Google Forms, MySQL y Power BI, con el fin de establecer un flujo automatizado y escalable de extracción, transformación y carga (ETL, por sus siglas en inglés).

A lo largo del desarrollo se abordan aspectos clave como el diseño del modelo entidad-relación (MER), la normalización de datos, la validación de relaciones y claves, y la construcción de consultas para análisis posteriores.

Este proyecto aporta una solución técnica que no solo responde a las necesidades particulares del DirectorioUsme.com, sino que también puede ser replicada en otros contextos locales donde el acceso estructurado a la información comercial represente una oportunidad de crecimiento y fortalecimiento comunitario.

Descripción del Problema

El sitio web DirectorioUsme.com busca facilitar el acceso a información actualizada sobre negocios, servicios y entidades públicas de la localidad de Usme, en Bogotá. Su misión es conectar a los habitantes con la oferta local. No obstante, actualmente enfrenta limitaciones críticas en la forma como gestiona sus datos.

Cuando un negocio se registra, la información recolectada a través de formularios se almacena en Google Sheets, mientras que datos provenientes de herramientas como Google Analytics, Search Console y AdSense se descargan en archivos separados. Esta dispersión dificulta la integración, el análisis, la trazabilidad y la toma de decisiones informadas.

Según (Orozco et al., 2021), la falta de integración y documentación técnica impide aprovechar plenamente los datos. Además, la ausencia de controles formales puede dar lugar a fugas de información o a accesos indebidos. En este contexto, (Machuca Vivar et al., 2022) destacan la importancia de aplicar seguridad desde el sistema operativo hasta la gestión de privilegios en las bases de datos.

Adicionalmente, Colombia cuenta con un marco legal robusto en protección de datos, basado en la Ley Estatutaria 1581 de 2012, que exige un tratamiento responsable de la información personal (GALINDO, s. f.).

Más allá del cumplimiento normativo, la correcta gestión de datos permitiría potenciar estrategias de marketing digital en eventos clave (como Black Friday, Día de la Madre, etc.), con el fin de incrementar el tráfico y mejorar la visibilidad de los negocios registrados. No obstante, durante 2023 y 2024, DirectorioUsme.com solo logró activar campañas en dos temporalidades por año, debido a la falta de análisis unificado.

Tal como señala (Treviño et al., 2020), el análisis de datos funciona como un puente entre las necesidades del consumidor y las decisiones empresariales. En el caso del DirectorioUsme.com, ese puente aún no existe.

Planteamiento del Problema

¿Cómo diseñar e implementar un sistema de gestión de datos que centralice el procesamiento, almacenamiento, análisis y visualización de la información en el sitio web DirectorioUsme.com, optimizando la toma de decisiones y el aprovechamiento de los datos disponibles?

Sistematización del Problema

Con el fin de descomponer y abordar de forma estructurada el problema identificado, se plantean las siguientes preguntas orientadoras que guían el desarrollo del proyecto:
¿Qué tipo de información recolecta actualmente DirectorioUsme.com y cómo está distribuida entre las distintas fuentes de datos?

- ¿Qué modelo de base de datos relacional es más adecuado para estructurar y consolidar dicha información de manera eficiente?
- ¿Qué herramientas tecnológicas permiten implementar un flujo ETL que automatice la extracción, transformación y carga de los datos recopilados desde Google Forms, Google Sheets y otras plataformas como Google Analytics, AdSense y Search Console?
- ¿De qué forma un sistema de gestión de datos puede mejorar la toma de decisiones, la ejecución de estrategias de marketing digital y el aprovechamiento de eventos de temporalidad para aumentar el tráfico y la visibilidad de los negocios registrados?

Estas preguntas permiten el análisis del problema desde distintos niveles técnico, funcional y estratégico, facilitando el diseño e implementación de una solución integral que responda a las necesidades actuales del sitio web DirectorioUsme.com.

Justificación

Actualmente, DirectorioUsme.com carece de un sistema de gestión de bases de datos definido. La información de los negocios interesados en registrarse se recolecta a través de diversos medios y se aloja en Google Sheets. Asimismo, los informes de herramientas como Google Analytics, AdSense y Search Console se gestionan de forma independiente en archivos Excel. Esta fragmentación genera duplicidad, posible pérdida de datos críticos y procesos manuales que extienden el tiempo de análisis de cinco a siete días. La constante manipulación de estos archivos por múltiples usuarios puede resultar en inconsistencias entre los distintos reportes.

Un ejemplo concreto de estas deficiencias se evidenció en 2023, durante la campaña del Día de la Madre. Se destacaron categorías como "desayunos sorpresa" y "floristerías" en la página de inicio, pero la subcategoría "productos personalizados" se omitió debido a una clasificación incorrecta en el sistema actual. Esta situación no solo afecta la percepción de los usuarios al proyectar una imagen de escasa variedad y falta de cobertura en eventos clave, sino que también impacta negativamente a los negocios registrados al no recibir la visibilidad esperada, lo que puede desincentivar su participación continua en la plataforma.

Falencias técnicas identificadas: actualmente, se han identificado tres falencias técnicas claves que limitan el potencial de DirectorioUsme.com. En primer lugar, existe una ausencia de un sistema de gestión de bases de datos estructurado, lo que dificulta la organización y el acceso eficiente a la información. En segundo lugar, se presentan deficiencias significativas en los procesos de Extracción, Transformación y Carga (ETL) y en la estandarización de datos, lo que conduce a inconsistencias y errores. Finalmente, se carece de una visualización centralizada de métricas en un *dashboard*, impidiendo un monitoreo ágil y una toma de decisiones informada.

Beneficios esperados con la implementación del sistema: En el área de publicidad, se estima una reducción del tiempo de análisis a entre dos y tres horas, lo que facilitará decisiones estratégicas clave. Esto permitirá la identificación de barrios con baja representación comercial para el diseño de campañas de captación, el monitoreo de comportamientos durante temporadas clave para la proyección efectiva de espacios publicitarios, y la visibilidad para negocios destacados, usándolos como casos de éxito para atraer nuevos registros. Respecto a la experiencia de usuario (UX), el sistema permitirá la identificación de patrones de uso y horarios de mayor tráfico. Esto contribuirá directamente a la inclusión estratégica de categorías menos visitadas para mejorar su alcance, así como a la revisión de comercios con baja interacción para verificar su vigencia o corregir posibles errores, optimizando así la interacción del usuario con la plataforma.

En cuanto a las métricas de *benchmarking*, el sistema posibilitará la medición de conversiones, como usuarios, eventos, para establecer referencias claras por categoría. Esto, a su vez, permitirá realizar estimaciones más realistas del tráfico esperado según el tipo de negocio.

Desde una perspectiva profesional, este proyecto representa una oportunidad invaluable para consolidar y aplicar los conocimientos adquiridos en la Especialización en Ciencia de Datos y Analítica. Permitirá integrar lenguajes como Python y SQL, entre otras áreas relacionadas, en un caso real con impacto comunitario.

Al tratarse de una iniciativa propia, en desarrollo desde hace más de dos años, su evolución también se convierte en una evidencia tangible del crecimiento personal y técnico.

Objetivos

Objetivo General

Desarrollar un sistema de gestión de base de datos para consolidar y analizar la información del DirectorioUsme.com, mediante herramientas tecnológicas, con el fin de mejorar su organización y uso eficiente.

Objetivos Específicos

Diseñar la arquitectura del sistema de base de datos, a partir del análisis de las fuentes de información del DirectorioUsme.com, con el fin de estructurar adecuadamente los datos que serán integrados.

Implementar el proceso de ETL utilizando herramientas tecnológicas especializadas, para extraer, transformar y cargar eficientemente la información en la base de datos centralizada.

Desarrollar funcionalidades de consulta y análisis de datos a través de interfaces interactivas y visualizaciones, para facilitar el acceso a la información consolidada y apoyar la toma de decisiones.

Marco de Referencia

Estado del Arte

Para respaldar la implementación de un sistema de gestión de datos en el sitio web DirectorioUsme.com, se realizó una revisión de antecedentes centrada en tres ejes fundamentales: el uso de bases de datos relacionales, la optimización de procesos ETL y la incorporación de herramientas de visualización como Power BI.

En cuanto a la estructuración y almacenamiento de datos, se destaca el proyecto desarrollado en la Universidad de Montemorelos (UM), donde se implementó una base de datos utilizando MySQL *Community Edition* para gestionar la información del sistema de capellanía. Este trabajo evidenció cómo el uso del modelo relacional, junto con claves primarias y foráneas, mejora la consistencia de los datos, facilita la normalización y reduce riesgos asociados a la pérdida de información por manipulación manual o almacenamiento en papel (Sánchez, 2020) La utilización de motores transaccionales permitió asegurar la integridad referencial y mejorar el rendimiento en las consultas, aspectos clave también en el diseño propuesto para el DirectorioUsme.com.

Por otra parte, en el estudio desarrollado por (Encalada Garcia, 2025), se diseñó un marco de trabajo para la implementación de procesos ETL optimizados, utilizando Python y Power BI como herramientas centrales. Este enfoque permitió integrar diversas fuentes de datos (CSV, JSON, Excel) y mejorar significativamente los tiempos de procesamiento, al tiempo que redujo errores en la transformación. Las bibliotecas *pandas*, *NumPy* y *pyodbc* fueron clave en la automatización de la limpieza, carga y análisis, alineándose con el flujo de trabajo propuesto en este proyecto.

Además, se identificó que el uso de Power BI como plataforma de visualización permite estructurar *dashboards* interactivos para tomar decisiones basadas en datos confiables. En el caso de su aplicación en pequeñas empresas, se demostró que herramientas como Power BI, conectadas a MySQL, representan una alternativa viable y accesible (Rivera Resina & others, 2018). Este hallazgo es especialmente relevante para un entorno como el de DirectorioUsme.com, que busca facilitar la toma de decisiones a partir de la información centralizada.

En síntesis, los antecedentes revisados demuestran que la integración de bases de datos relacionales, procesos ETL y herramientas de visualización para análisis como Power BI no solo mejora la administración de datos, sino que también permite escalar soluciones en diferentes sectores, desde instituciones académicas hasta microempresas. Estos hallazgos respaldan la pertinencia del presente proyecto, que adapta estas tecnologías a un contexto comunitario con potencial de alto impacto social.

Marco Contextual

La localidad quinta de Usme, ubicada al sur de Bogotá D.C., se caracteriza por tener un 86 % de su territorio en suelo rural, un 10 % en suelo urbano y un 4 % en suelo de expansión urbana (Secretaría Distrital de Planeación, 2023). Su población está conformada por comunidades tradicionales y nuevos asentamientos urbanos. La actividad económica es diversa, abarcando desde la agricultura, la pesca y el turismo hacia miradores, hasta servicios institucionales como registraduría y notarías, además de una amplia oferta comercial que incluye restaurantes, droguerías, colegios privados, trabajadores independientes, fábricas textiles y microempresas entre muchos más.

No obstante, gran parte de esta actividad aún permanece desconectada del entorno digital, lo cual limita su visibilidad y reduce su capacidad para competir en igualdad de condiciones con sectores más desarrollados. Con el impacto de la pandemia, muchos negocios locales se vieron obligados a cerrar, sin disponer de canales adecuados para comunicar sus productos o servicios.

En este contexto, surge la iniciativa DirectorioUsme.com, un sitio web creado con el objetivo de conectar a los habitantes de la localidad con los comercios, servicios y entidades públicas de su entorno. Lo que comenzó como un proyecto personal enfocado en el aprendizaje de analítica y SEO, se convirtió en un servicio comunitario que promueve la organización y divulgación comercial local. Como parte de su consolidación, el 17 de marzo de 2023, la Superintendencia de Industria y Comercio (SIC) otorgó a DirectorioUsme.com el reconocimiento oficial como marca registrada en Colombia, fortaleciendo así su identidad legal y digital.

Actualmente, la operación del sitio depende del uso de hojas de cálculo (Google Sheets y Excel) para recopilar información personal y comercial de los negocios registrados. Paralelamente, el análisis de métricas del portal se realiza con herramientas como Google Analytics, AdSense y Search Console, cuyos datos se gestionan manualmente y de manera independiente.

Esta fragmentación en las fuentes y formatos de datos ha generado limitaciones en el análisis estratégico, la trazabilidad histórica y la visualización integrada de la información. Además, la ausencia de un sistema de gestión de base de datos robusto incrementa los riesgos de pérdida, duplicación o manipulación accidental de los registros, lo cual afecta la toma de decisiones y la experiencia de los usuarios.

En consecuencia, la implementación de un sistema centralizado de gestión de datos representa una oportunidad para optimizar el manejo de la información, fortalecer los procesos internos y generar reportes útiles tanto para los administradores del portal como para los negocios registrados y la ciudadanía. La propuesta se desarrolla en un entorno real, comunitario y en crecimiento, donde la tecnología se convierte en un recurso clave para fortalecer el tejido económico y social del territorio.

Marco Conceptual

Para avanzar en la ejecución del proyecto, se definen y explican las siguientes variables, con el fin de establecer el lenguaje técnico que se empleará en el desarrollo de la solución al problema planteado.

Normalización de los Datos

Como menciona (Lopez-Nunez, s. f.), el proceso de normalización en bases de datos es fundamental al construir un modelo relacional, ya que permite evitar errores y reprocesos futuros al momento de actualizar o eliminar tablas o registros. Estos errores se evidencian actualmente en el DirectorioUsme.com, donde la información no está unificada, existe dispersión en diferentes archivos y no se ha definido el tipo de dato en cada tabla ni las posibles relaciones entre ellas.

Tipos de Datos. Según (Lopez-Nunez, s. f.), “el tipo de un dato es como una etiqueta, característica o atributo que va asociado al dato y que define una cualidad muy importante del mismo”. Estos se clasifican comúnmente en:

Entero: Números reales sin decimales, ya sean negativos o positivos (ej. 1, 2, -3)

- Flotante: Números con decimales (ej. 1.2, -3.3, 7.6).

- Booleano: Solo admite dos valores: True o False.
- String: Cadenas de texto, como nombres, apellidos o atributos categóricos.
- Fecha: Representa datos temporales, como fecha de nacimiento, ingreso o salida.

Base de Datos

Inicialmente, se creía que tener un archivo con información era suficiente para considerarlo una base de datos en el DirectorioUsme.com. Sin embargo, de acuerdo con (Valverde et al., 2019), *"las bases de datos son grandes cantidades de información almacenadas en registros para lograr una mejor eficiencia al momento de ingresar, buscar, actualizar o eliminar la información"*. Esta definición permite comprender que una verdadera base de datos requiere estructura, integridad y capacidad de gestión eficiente.

Modelos de Bases de Datos

Existen dos tipos principales de modelos de bases de datos: relacionales y no relacionales. (Valverde et al., 2019) explica las características que componen cada una:

Bases de Datos No Relacionales**:** Este tipo de base de datos trabaja con documentos y, aunque su costo inicial puede ser mayor que el de una base de datos relacional, dicho costo se mantiene constante con el tiempo. Poseen escalabilidad horizontal y se fundamentan en el Teorema CAP, que contempla los siguientes principios:

- Consistencia: Cualquier actualización o modificación debe reflejarse de forma inmediata en todos los accesos al sistema.
- Disponibilidad: El sistema debe permitir acceso constante y rápido a la información.
- Tolerancia al particionamiento: El sistema debe continuar funcionando incluso si alguna parte de este no está momentáneamente disponible.

Bases de Datos Relacionales**:** Trabajan con datos estructurados y su costo inicial suele ser menor. Su escalabilidad es vertical, aunque puede ampliarse si se presenta un incremento excesivo en el tráfico de datos. Estas bases de datos se rigen por los principios ACID, por sus siglas en inglés (Atomicity, Consistency, Isolation, Durability), los cuales se describen a continuación:

- Atomicidad: La transacción debe ejecutarse en su totalidad o no ejecutarse en absoluto.
- Consistencia: Garantiza que el sistema pase de un estado válido a otro también válido, cumpliendo siempre con las reglas de integridad establecidas.
- Aislamiento: Permite que varias transacciones puedan ejecutarse sobre el mismo registro de forma independiente.
- Durabilidad: Una vez completada una transacción, los cambios realizados permanecen almacenados de forma permanente.

Estos principios también son respaldados por el estudio de (Marrero et al., 2019), el cual compara las bases de datos relacionales con las no relacionales. En sus conclusiones, *"En línea general se puede concluir que no hay un motor de base de datos que logre el mejor tiempo de respuesta para todas las consultas evaluadas. Los resultados obtenidos para los casos de estudio planteados dependen del tipo de consulta y del volumen de datos asociados.*

ETL

Como resalta (Palacios Martel, 2019a), el proceso ETL, por sus siglas en inglés (*Extract, Transform, Load*), es un conjunto de tres pasos secuenciales mediante los cuales los datos se trasladan desde un sistema origen hasta un almacén de datos. Este proceso garantiza que la información esté limpia, normalizada y lista para su análisis, visualización y toma de decisiones.

Además, permite integrar datos provenientes de múltiples fuentes en una única estructura unificada y coherente.

Sistema de Gestión de Bases de Datos (SGBD)

Un Sistema de Gestión de Bases de Datos (SGBD) es un software que actúa como interfaz entre los usuarios y la base de datos, permitiendo almacenar, acceder y manipular datos de manera eficiente. Según (Soberón & Jesús, 2020), estos sistemas proporcionan herramientas para describir, administrar y consultar los datos, facilitando así la interacción del usuario con la información. Por ejemplo, un SGBD permite ejecutar consultas complejas mediante un lenguaje accesible como SQL.

Los SGBD ofrecen una amplia gama de funcionalidades, tales como seguridad, control de concurrencia, recuperación ante fallos y optimización de consultas. Además de los sistemas relacionales tradicionales como PostgreSQL, MySQL y Oracle, en los últimos años han surgido nuevas opciones como MongoDB, las cuales pertenecen al ecosistema NoSQL y están diseñadas para gestionar grandes volúmenes de datos no estructurados o semiestructurados.

Las tendencias actuales en el ámbito de los SGBD incluyen la adopción de soluciones en la nube, el uso de contenedores (como Docker) para su despliegue, y su integración con tecnologías emergentes de Big Data y analítica avanzada.

MySQL*:* De acuerdo a su libro (Urtiaga, 2020), **MySQL** es un sistema de gestión de bases de datos relacionales que utiliza el lenguaje **SQL** para realizar consultas, inserciones, eliminaciones y actualizaciones de datos. Además, permite la creación de roles con distintos niveles de acceso, lo que contribuye a una mejor administración de la seguridad y los permisos dentro del sistema.

SQL *(Structured Query Language):* SQL, cuyas siglas en español significan (Lenguaje de Consulta Estructurado), es el lenguaje estándar de facto para interactuar con bases de datos relacionales. Según (Soberón & Jesús, 2020), este lenguaje permite a los desarrolladores declarar un amplio conjunto de condiciones, restricciones, afirmaciones y transformaciones sobre los datos, convirtiéndose así en una herramienta indispensable en el desarrollo y gestión de aplicaciones basadas en bases de datos.

Marco Teórico

La falta de un sistema de gestión de datos eficiente en el DirectorioUsme.com ha dificultado la toma de decisiones basadas en datos, lo que ha conllevado una disminución en la calidad de los servicios ofrecidos. Con el objetivo de solucionar esta problemática, se llevará a cabo una revisión exhaustiva de la literatura científica, consultando artículos y libros de diversos autores para construir un marco teórico sólido que sustente las soluciones propuestas en este proyecto. Este marco teórico permitirá identificar las mejores prácticas y herramientas para desarrollar una solución efectiva y sostenible.

Sistema de Gestión de Bases de Datos (SGBD)

Un Sistema de Gestión de Bases de Datos (SGBD) actúa como un intermediario entre el usuario y la base de datos, operando como una interfaz que facilita el acceso, almacenamiento y recuperación de la información de manera eficiente y segura (Soberón & Jesús, 2020). Esto cobra especial relevancia en proyectos como el DirectorioUsme.com, donde se prevé gestionar grandes volúmenes de datos relacionados con categorías, subcategorías y registros de negocios locales.

Además de garantizar la seguridad y consistencia de la información, un SGBD se vuelve indispensable cuando la escala del sistema alcanza “centenares de gigabytes y millares de

usuarios” (Torres, 2021), escenarios comunes en sistemas destinados a la consulta masiva de información. En este contexto, el SGBD no solo organiza los datos, sino que también optimiza las consultas y operaciones, garantizando tiempos de respuesta rápidos incluso bajo una alta carga de usuarios.

Para este proyecto, se propone implementar un sistema de base de datos relacional mediante un SGBD como **MySQL**, el cual permite estructurar la información en tablas relacionadas y ejecutar consultas eficientes mediante **SQL**. Esto resulta esencial para ofrecer funcionalidades como:

- Filtrar negocios por categorías específicas (transporte público, academias, etc.).
- Consultar subcategorías de servicios.
- Garantizar la integridad y coherencia de los datos al escalar la base con nuevas entradas.

El uso de un SGBD también permite establecer políticas de seguridad y control de acceso para proteger los datos de los negocios y usuarios que interactúan con el sistema. En este sentido, el SGBD no es únicamente una herramienta técnica, sino un pilar fundamental para garantizar la funcionalidad y sostenibilidad del proyecto a largo plazo.

MySQL

MySQL se destaca como uno de los sistemas de gestión de bases de datos relacionales más populares y utilizados a nivel mundial, adoptado por empresas como Facebook y Google (Marrero et al., 2019). Su capacidad para gestionar grandes volúmenes de datos, combinada con su alta velocidad de consulta (Rawat et al., 2021), lo convierte en una opción ideal para proyectos como DirectorioUsme.com. Además, MySQL ofrece sólidas características de seguridad, una amplia comunidad de usuarios y una variedad de opciones de licenciamiento.

Considerando estas ventajas, y en alineación con las habilidades adquiridas durante el proceso de formación en SQL, se concluye que MySQL es la solución más adecuada para satisfacer las necesidades del proyecto, garantizando un sistema de gestión de datos eficiente, escalable y seguro.

SQL

Cuyas siglas en inglés significan *Structured Query Language* (Lenguaje de Consulta Estructurado), es fundamental para la manipulación de datos en bases de datos relacionales. Según (Rawat et al., 2021), este lenguaje permite realizar operaciones como filtrar, seleccionar e introducir datos de manera eficiente. (Pacheco Castañeda, 2019) destaca "la inminente necesidad de tener un dominio avanzado del lenguaje SQL" para trabajar con grandes volúmenes de datos, ya que permite realizar consultas complejas y obtener respuestas precisas que respaldan la toma de decisiones. El lenguaje SQL se divide en tres componentes principales:

Data Definition Language (DDL)*:* Se utiliza para definir la estructura de la base de datos. Permite crear tablas, índices y atributos, así como establecer relaciones entre ellos, como claves primarias y foráneas. Según Pacheco Castañeda (2019), "cualquier información que vaya a formar parte del catálogo o diccionario de la base de datos tendrá sentencias básicas de DDL". Las sentencias principales son:

- CREATE: Crea nuevos objetos, como tablas o índices, en la base de datos.
- ALTER: Modifica la estructura de objetos existentes.
- DROP: Elimina objetos existentes de la base de datos.

Data Manipulation Language (DML): Esta es probablemente la parte más utilizada y conocida de SQL. A través del DML se redactan sentencias para seleccionar, eliminar,

actualizar, filtrar o agrupar conjuntos de datos (Pacheco Castañeda, 2019). Las sentencias básicas de DML son:

- SELECT: Permite seleccionar campos y registros específicos de la base de datos.
- INSERT: Inserta nuevos registros en una tabla.
- UPDATE: Modifica registros existentes.
- DELETE: Elimina registros específicos de una tabla.

Data Control Language (DCL): *C*orresponde al lenguaje de control de datos. Estas sentencias permiten definir privilegios para los usuarios y administrar transacciones dentro de la base de datos. Según (Pacheco Castañeda, 2019), las sentencias principales de DCL son:

- COMMIT: Confirma los cambios realizados en la base de datos.
- ROLLBACK: Revierte los cambios no confirmados, restaurando el estado anterior.
- GRANT: Concede permisos o privilegios a un usuario para operar sobre la base de datos.
- REVOKE: Retira permisos o privilegios previamente concedidos.

Bases de Datos

Una base de datos es un conjunto de tablas que almacenan datos estructurados, permitiendo realizar procesos como ingresar, buscar, actualizar o eliminar información. Según (Valverde et al., 2019), estas bases también permiten establecer controles de acceso personalizados, lo que garantiza que la información sea segura, consistente y conserve su integridad a lo largo del tiempo.

En el contexto del DirectorioUsme.com, la implementación de una base de datos centralizada resulta esencial para optimizar el almacenamiento y la gestión de datos. Este

enfoque no solo permite unificar la información que actualmente puede estar dispersa en hojas de cálculo o registros manuales, sino que también facilita consultas ágiles y análisis confiables que respalden la toma de decisiones estratégicas.

Asimismo, mejora la seguridad del sistema al proteger los datos contra accesos no autorizados, incrementando la confianza de los usuarios.

Tipos de Bases de Datos

Existen diversos tipos de bases de datos, cada uno adaptado a diferentes necesidades de estructura y procesamiento. Los más relevantes para este proyecto son:

Bases de Datos Relacionales*:* En las bases de datos relacionales, la información se organiza en tablas que deben estar relacionadas entre sí mediante llaves primarias y foráneas. Cada tabla tiene una llave primaria única, que puede vincularse a otra tabla mediante una llave foránea. (Valverde et al., 2019) mencionan que "*las bases de datos relacionales tienen atomicidad, es decir, la capacidad de cambiar varias tablas al mismo tiempo o no cambiar ninguna*". Este principio forma parte de las reglas ACID que garantizan la integridad de las transacciones.

Por su naturaleza estructurada y su capacidad para realizar relaciones entre entidades, las bases de datos relacionales se ajustan mejor al modelo operativo del DirectorioUsme.com. En este entorno, es necesario realizar operaciones como filtrar negocios por categorías, vincular subcategorías, y aplicar filtros para identificar patrones. Esto requiere un sistema que permita mantener la coherencia y consistencia en las modificaciones realizadas a lo largo del tiempo.

Bases de Datos No Relacionales (NoSQL): Las bases de datos NoSQL surgieron como una solución ante el crecimiento exponencial de datos no estructurados, en parte impulsado por tecnologías como el Internet de las Cosas (IoT) y el desarrollo ágil de aplicaciones. A diferencia

de las bases relacionales, no utilizan un esquema fijo ni el lenguaje SQL como estándar. (Marrero et al., 2019) aclaran que "*NoSQL no es exactamente un tipo de base de datos, sino un conjunto de tipos de bases de datos*", que incluyen modelos de documentos, grafos, columnas y clave-valor. Estas no requieren estructuras tabulares ni admiten fácilmente operaciones JOIN o relaciones complejas entre entidades.

Dado que el proyecto DirectorioUsme.com se basa en datos con una estructura tabular, aunque aún en desarrollo, las bases de datos NoSQL no resultan adecuadas para su implementación. El sistema requiere realizar relaciones entre registros y consultas cruzadas, por lo que es preferible optar por un modelo relacional. Esto permitirá identificar patrones útiles en estrategias de publicidad, segmentación de servicios y mejoras en la experiencia de usuario (UX).

Una de las conclusiones clave del estudio comparativo realizado por (Marrero et al., 2019) es que "*en línea general se puede concluir que no hay un motor o tipo de base de datos que logre el mejor tiempo de respuesta para todas las consultas evaluadas. Los resultados obtenidos para los casos de estudio planteados dependen del tipo de consulta planteada y el volumen de datos asociado*". Esta afirmación refuerza la importancia de elegir el tipo de base de datos en función de los requerimientos específicos del proyecto.

Nomenclatura de Tablas

Cuando se implementa un sistema de bases de datos relacional y se definen las relaciones entre tablas, resulta fundamental asignar nombres adecuados tanto a las tablas como a los campos que las componen. Aunque este aspecto puede parecer secundario, una nomenclatura bien estructurada es crucial para garantizar la coherencia, claridad y mantenibilidad del sistema a largo plazo. La asignación incorrecta de nombres basada en preferencias personales o criterios

informales puede generar ambigüedades, errores en las consultas y dificultades en la colaboración entre desarrolladores o administradores de bases de datos.

Según (Giménez, 2019), una de las convenciones más recomendadas en entornos profesionales es el uso de la notación "snake_case", ampliamente adoptada en el ámbito de la programación y el modelado de bases de datos.

Esta convención consiste en escribir los nombres en minúsculas, separados por guiones bajos (_), evitando tildes, espacios y caracteres especiales como #$%&/. Además, se recomienda utilizar nombres cortos, descriptivos y consistentes, que reflejen claramente el propósito de cada tabla o campo.

En el contexto del proyecto DirectorioUsme.com, una nomenclatura coherente no solo mejorará la comprensión del modelo de datos, sino que también facilitará futuras tareas de mantenimiento, escalabilidad y colaboración.

A continuación, se presentan ejemplos que se pueden aplicar al proyecto:

- categoria: Tabla para almacenar información general sobre las categorías principales del directorio.
- transporte_publico: Tabla para registrar subcategorías específicas dentro del rubro de transporte.

En lo que respecta a los campos de las tablas, (Giménez, 2019) también sugiere utilizar nombres en singular, como parte de las buenas prácticas de normalización.

Esto permite identificar con mayor claridad el contenido de cada columna y facilita la redacción de sentencias SQL más legibles y precisas, como se muestra en la (tabla 1).

Tabla 1

Ejemplo de Nomenclatura de los Atributos

Campo	Descripción
numero_cedula	Número de identificación del cliente
nombre_cliente	Nombre completo del cliente
correo_cliente	Dirección de correo electrónico
celular_1	Número de teléfono principal del cliente

Nota. Esta tabla ejemplifica la nomenclatura que debe tener los atributos dentro de una tabla

Este tipo de estructuración permite mantener uniformidad y legibilidad tanto en el diseño físico de la base de datos como en el desarrollo de consultas SQL, sentencias DML/DDL y procesos automatizados.

Normalización

La normalización es un proceso fundamental en el diseño de bases de datos relacionales, cuyo objetivo es organizar los datos para eliminar redundancias, garantizar la integridad y mejorar la escalabilidad del sistema. Aplicar este proceso adecuadamente permite mantener una estructura coherente y eficiente a medida que crece la cantidad de información. Según (Giménez, 2019), la normalización se fundamenta en el cumplimiento de una serie de formas normales que progresivamente corrigen anomalías en la estructura de las tablas.

Primera Forma Normal **(1FN).** Para que una tabla cumpla con la Primera Forma Normal, debe cumplir las siguientes condiciones: Presencia de clave primaria en cada tabla debe poseer un atributo o conjunto de atributos que identifique de forma única cada registro. Además, se deben garantizar ciertas condiciones como el uso de campos atómicos, es decir, que cada celda

contenga un único valor; por ejemplo, el campo *nombre_subcategoria* no debe incluir múltiples datos separados por comas. También debe asegurarse la ausencia de valores nulos en la clave primaria, ya que todo registro debe contar con un identificador único. Finalmente, se requiere la independencia del orden de filas y columnas, de modo que el significado y funcionalidad de los datos no se vean afectados si se modifica su disposición.

Tabla 2

Ejemplo de Duplicidad de Claves Primarias

ID_Cliente	Nombre	Teléfono
1	Ana Torres	3124567890
1	Ana Torres	3012345678
2	Luis González	3105674321

*Nota***.** Esta tabla muestra un error en el diseño de bases de datos relacionales, donde se repite el mismo valor en la clave primaria (ID_Cliente), lo cual puede generar inconsistencia y pérdida de integridad en los datos. Cada clave primaria debe ser un identificador único

Segunda Forma Normal (2FN): La segunda forma normal puede aplicarse únicamente si la tabla cumple con la primera forma normal. Ningún campo debe depender de una parte de la clave primaria; tienen que depender completamente de la clave primaria, evitando dependencias parciales. (Giménez, 2019) "*una dependencia parcial ocurre cuando un campo depende de parte de una clave primaria compuesta, en lugar de toda la clave primaria. Para eliminar estas dependencias, se deben crear tablas separadas*".

A continuación, en la (tabla 3) se visualiza un ejemplo de una tabla sin normalizar.

Tabla 3

Ejemplo Tabla de Subcategorías de Academias sin Normalizar

ID_Subcategoría (PK)	Nombre_Subcategoria	Id_Usuario
155	Academia de arte, baile y música	12
138	Academias de belleza	15
115	Academias de conducción	18
101	Academias de seguridad privada	20

Nota. Esta tabla no cumple con la Segunda Forma Normal (2FN), ya que representa únicamente la entidad Subcategorías, pero incluye el campo Id_Usuario, el cual pertenece a una entidad diferente (Usuarios Administradores). Esto viola el principio de separación de entidades, por lo que dicho campo debe eliminarse de esta tabla y gestionarse mediante una relación aparte

Este proceso es esencial para evitar redundancias, facilitar la actualización de datos y garantizar la integridad de la base de datos. En el contexto del proyecto, la normalización asegura que el sistema de gestión de información sea robusto y eficiente, permitiendo que las consultas sean más rápidas y que el mantenimiento de la base de datos sea más sencillo a medida que esta crezca.

Tercera Forma Normal (3FN): La tercera forma normal establece que no debe haber dependencias transitivas en las tablas. Esto significa que los campos no clave no deben depender de otros campos que tampoco sean clave primaria. Evitar dependencias transitivas, una dependencia transitiva ocurre cuando un campo depende de otro campo no clave primaria, en lugar de depender directamente de la clave primaria. Ejemplo: Consideremos una tabla donde se almacenan los datos de subcategorías, junto con el nombre y el correo del administrador:
A continuación, se evidencia ejemplos de lo mencionado en las (tablas 4, 5 y 6).

Tabla 4

Ejemplo de Tabla con Dependencia Transitiva

ID_Subcategoría (PK)	Nombre_Subcategoría	ID_Usuario_Admin (PK)	Nombre_Admin	Correo_Admin
155	Academia de arte, baile y música	12	Juan Pérez	juan.perez@email.com
138	Academias de belleza	15	Ana López	ana.lopez@email.com

Nota**.** En esta tabla, los campos Nombre_Admin y Correo_Admin dependen del ID_Usuario_Admin, y no directamente de la clave primaria compuesta, lo que representa una dependencia transitiva

Para cumplir con la tercera forma normal, la tabla se divide en dos:

Tabla 5

Tabla Subcategoría Normalizada

ID_Subcategoría (PK)	Nombre_Subcategoría	ID_Usuario_Admin (FK)
155	Academia de arte, baile y música	12
138	Academias de belleza	15

Nota. En esta tabla normalizada, el ID_Usuario_Admin se convierte en una clave foránea. No es necesario repetir los datos del administrador en cada registro

Tabla 6

Tabla Administradores Normalizada

ID_Usuario_Admin (PK)	Nombre_Administrador	Correo_Administrador
12	Juan Pérez	juan.perez@email.com
15	Ana López	ana.lopez@email.com

Nota. Esta tabla almacena de forma separada y estructurada los datos de los administradores, cumpliendo con los principios de la tercera forma normal

Ahora, Nombre_Administrador y Correo_Administrador dependen únicamente de la clave primaria ID_Usuario_Admin, y ya no existe dependencia transitiva en la tabla de subcategorías. Esto garantiza una mayor integridad, menor redundancia y mejor escalabilidad en el diseño de la base de datos.

Tipos de Datos: Según (de Mendıvil & Las Encinas, s. f.), "*dato es una palabra que, en el contexto de la informática, describe todo aquello con lo que puede operar un ordenador*". En este sentido, al diseñar una base de datos, es imprescindible asignar adecuadamente los tipos de datos a cada columna o campo. Este paso, aunque técnico, es crucial para asegurar que las búsquedas, operaciones y análisis se realicen correctamente.

Una incorrecta definición del tipo de dato puede provocar errores en los resultados o incluso limitar el uso de ciertas funcionalidades. Por ejemplo, si en una tabla que registra información personal la columna "edad" se define como un tipo de texto en lugar de numérico, se imposibilitará realizar operaciones estadísticas como promedios, comparaciones o agrupamientos. A continuación, se describen los tipos de datos más utilizados en programación y análisis de datos, con ejemplos ajustables al contexto de DirectorioUsme.com:

Tipo de Dato Entero (integer): Este tipo se utiliza para representar números enteros, es decir, aquellos que no contienen decimales, incluyendo tanto valores positivos como negativos y el cero. Según (de Mendıvil & Las Encinas, s. f.), "*se escriben como lo hacemos habitualmente: 122, −33; y nunca usamos el punto o la coma decimal a la hora de escribir estos número*s".

Algunas aplicaciones prácticas son: calcular la cantidad de negocios registrados por categoría o localidad, determinar el número de visitas a un anuncio o perfil empresarial, almacenar la edad del usuario visitante si se recopilan datos demográficos.

Tipo de Dato Flotante: Representa números racionales, es decir, aquellos que contienen decimales. Pueden expresarse con punto decimal o notación científica, como 3.3, -4.5 o 1.23e4 (De Mendívil & Las Encinas, s. f.). Algunas aplicaciones prácticas son: Promedio de valoraciones de un negocio (escala 0.0 a 5.0), número de sesiones promedio por usuario (por ejemplo, 1.2 o 4.5).

Tipo de Dato Booleano: Permite representar valores binarios: verdadero o falso, sí o no, 1 o 0. Según De Mendívil y Las Encinas (s. f.), esto responde al principio de exclusión, donde una proposición solo puede ser cierta o falsa. Algunas aplicaciones prácticas son:

- Estado de un negocio (activo/inactivo).
- Verificación del perfil (verificado/no verificado).

Tipo de Dato Carácter o Cadena de Texto (*string*): Permite almacenar texto alfanumérico, como nombres, descripciones o direcciones. Estos valores suelen estar entre comillas simples o dobles. Algunas aplicaciones prácticas son: nombre de barrios o categorías ("La Aurora", "Restaurantes"), descripciones de los servicios ofrecidos por los negocios.

Proceso ETL: (Extract, Transform, Load)

El proceso ETL en español (*Extracción, Transformación y Carga*) es esencial en proyectos que requieren consolidación y calidad de datos, como el DirectorioUsme.com. Este proceso optimiza la limpieza, estructuración y almacenamiento de la información, asegurando que los datos sean confiables y útiles para el análisis (Conde Ramírez, 2022)

Extracción: Consiste en recolectar datos desde múltiples fuentes, que pueden ser estructuradas o no estructuradas. Estas fuentes incluyen archivos de texto, formularios web, hojas de cálculo, sitios web o bases de datos externas (Acosta Díaz & Del Águila Jacobo, s. f.). Algunas aplicaciones prácticas se encuentran en extraer datos desde formularios de registro de negocios, documentos locales o plataformas comerciales aliadas.

Transformación: Implica limpiar, enriquecer y estandarizar los datos extraídos. Las tareas más comunes son:

- Normalización: Uniformar tipos de datos y formatos (por ejemplo, fechas o nombres).
- Eliminación de duplicados: Asegurar que no existan entradas repetidas.
- Enriquecimiento: Completar datos faltantes o derivados, como asignar categorías.
- Validación: Verificar que los datos cumplan con reglas lógicas (ej. correos válidos).

Carga: Consiste en almacenar los datos transformados en el sistema principal, ya sea una base de datos relacional o un almacén de datos (data warehouse). Esto permite consultas eficientes y análisis avanzados. Algunas aplicaciones prácticas se encuentran en:

- Almacenar datos estructurados en el sistema del DirectorioUsme.com para permitir búsquedas por categoría, ubicación o estado del negocio.

- El proceso ETL garantiza que los datos almacenados en DirectorioUsme.com sean consistentes, fiables y útiles para la toma de decisiones (López Espinoza, 2024). Su implementación asegura no solo la calidad informativa del sistema, sino también su escalabilidad y sostenibilidad a medida que el proyecto crece.

Seguridad y Protección de Datos

La seguridad y protección de los datos es un componente esencial en cualquier sistema de información, especialmente cuando se gestionan datos sensibles o personales, como en el caso del proyecto DirectorioUsme.com. Para garantizar un entorno digital confiable, este proyecto se basa en la Triada de la Seguridad de la Información, la cual está compuesta por tres principios fundamentales: confidencialidad, integridad y disponibilidad (CID). Esta triada constituye el marco base para la gestión segura de los datos en sistemas informáticos (Esquivel & Sevilla, 2021).

Confidencialidad: La confidencialidad asegura que solo los usuarios autorizados puedan acceder a la información almacenada. Se busca evitar que datos personales, como correos electrónicos o información de contacto de negocios registrados, sean vistos, copiados o modificados por personas no autorizadas. Algunas aplicaciones prácticas se encuentran en la implementación de roles de usuario, donde los administradores tienen privilegios diferentes a los usuarios comunes. Control de acceso mediante credenciales seguras (nombre de usuario y contraseña). Integridad, con este principio se refiere a la exactitud y consistencia de los datos durante todo su ciclo de vida. La integridad impide que los datos sean modificados de forma indebida, ya sea por errores humanos, ataques maliciosos o fallos del sistema. Algunos ejemplos en la aplicación**:**

- Validación de formularios para evitar el ingreso de datos incorrectos o manipulados.
- Uso de restricciones y reglas en la base de datos que impidan relaciones inválidas o datos faltantes.

Disponibilidad: La disponibilidad garantiza que la información esté accesible para los usuarios autorizados cuando la necesiten, sin interrupciones indebidas. Es especialmente importante en entornos digitales que deben estar activos en todo momento.

- Servidores con alta disponibilidad y sistemas de respaldo (*backup*).
- Políticas de recuperación ante desastres para restaurar datos en caso de fallos.

Procedimientos Almacenados (Stored Procedure)

Una práctica clave para reforzar la seguridad y la eficiencia en la gestión de bases de datos es la implementación de procedimientos almacenados (*Stored procedure*). Estos permiten encapsular lógica de negocio directamente en el servidor de bases de datos, limitando el acceso directo a las tablas y reduciendo los riesgos asociados a manipulaciones indebidas de los datos (Giménez, 2019)

De acuerdo con (Giménez, 2019), un procedimiento almacenado es "*Un conjunto de comandos SQL que pueden guardarse en el servidor. Una vez que se hace, los clientes no necesitan lanzar cada comando individual, sino que pueden en su lugar llamar al procedimiento almacenado como un único comando.*" Algunas de las ventajas de los procedimientos almacenados son:

Seguridad mejorada: Al encapsular las operaciones SQL dentro de procedimientos, se evita que los usuarios interactúen directamente con la información original, lo que disminuye significativamente el riesgo de ataques como la inyección de SQL.

Control de permisos: Los sistemas gestores de bases de datos permiten asignar permisos específicos sobre los procedimientos almacenados, en lugar de otorgar acceso directo a las tablas. Esto permite establecer con precisión qué acciones puede ejecutar cada tipo de usuario.

Estandarización y eficiencia: Se centraliza la lógica de negocio, lo que asegura que las operaciones sobre los datos sigan un mismo flujo y validaciones. Esto reduce errores, mejora la mantenibilidad del sistema y facilita el trabajo en equipos de desarrollo.

Los procedimientos almacenados, además de mejorar el rendimiento y la organización del código SQL, constituyen una herramienta vital en la estrategia de seguridad de bases de datos. Su uso en el proyecto DirectorioUsme.*com* no solo mejora la eficiencia operativa, sino que también fortalece la protección de la información de negocios y usuarios, asegurando una plataforma más segura, estable y confiable.

Metodología

Para el desarrollo del proyecto del sitio web DirectorioUsme.com, se adopta la metodología CRISP-DM (***C****ross Industry Standard Process for Data Mining*), la cual constituye un estándar ampliamente utilizado tanto en la industria como en el ámbito académico para la ejecución de proyectos de minería y análisis de datos.

De acuerdo a (Espinosa-Zúñiga, 2020) "*…es un estándar empleado a nivel mundial tanto en la industria como en la academia para proyectos de minería de datos*". Esta metodología se caracteriza por su enfoque estructurado, pero flexible, que permite adaptar las etapas según las necesidades específicas del proyecto. Está compuesta por seis fases fundamentales, cada una de las cuales orienta el desarrollo en tareas específicas del ciclo de vida de un proyecto de análisis de datos.

- Comprensión del Negocio
- Comprensión de Datos
- Preparación de Datos
- Modelado
- Evaluación
- Implementación

Esta metodología nos indica que los pasos no son rígidos y define un conjunto de tareas y actividades para cada fase de un proyecto. Como señala (Ramírez González, s. f.), "*…pero no establece cómo llevarlas a cabo, por lo que se considera un aspecto positivo para el equipo de analítica o científico de datos, dado que puede incorporar su criterio en el desarrollo de estas"*.

Fase 1 Comprensión del Problema o Negocio

Esta fase inicial tiene como propósito comprender en profundidad la problemática que afecta al sitio web DirectorioUsme.com. Según (Brzozowska et al., 2023), esta etapa permite identificar los objetivos del negocio y traducirlos en una definición clara del problema que se desea resolver mediante el análisis de datos. La definición precisa del objetivo del proyecto de minería de datos constituye uno de los aspectos más relevantes de esta fase, ya que proporciona la línea base sobre la cual se orientará el desarrollo posterior del proyecto. Esta fase se compone de las siguientes etapas:

Identificación del Problema

En esta etapa se delimita la problemática existente mediante el análisis de las condiciones actuales del entorno. Asimismo, se identifican los requisitos del proyecto, sus restricciones, los actores involucrados y los beneficios esperados. Esta comprensión integral permite orientar el enfoque técnico y estratégico de la solución propuesta.

Determinación de Objetivos

El propósito de esta etapa es establecer los objetivos que la organización busca alcanzar mediante el análisis y aprovechamiento de los datos disponibles. En este contexto, se identifican las metas estratégicas y funcionales del sitio **DirectorioUsme.com**, priorizando el uso de los datos como insumo clave para la toma de decisiones basadas en evidencia.

Evaluación de la Situación Actual

Esta etapa consiste en documentar el estado actual del sitio web, teniendo en cuenta las herramientas utilizadas, los flujos de trabajo existentes y los problemas operativos identificados. Dicha evaluación proporciona una referencia comparativa que permitirá medir el impacto real de

la solución implementada, mediante el seguimiento de indicadores clave antes y después del desarrollo del sistema.

Fase 2 Comprensión de Datos

Esta fase tiene como propósito conocer en profundidad los datos disponibles y evaluar su adecuación respecto a los objetivos definidos en la etapa anterior. Según (Cortina, 2015), esta fase permite establecer un primer contacto con los datos, analizar su calidad y determinar las relaciones iniciales que pueden orientar la formulación de hipótesis. Junto con las fases de preparación de los datos y modelado, esta etapa demanda uno de los mayores esfuerzos en términos de tiempo y recursos, debido a la necesidad de comprender la estructura, el formato, el contenido, la calidad y la utilidad de los datos antes de iniciar cualquier proceso de transformación o análisis detallado.

La fase de comprensión de los datos actúa como un puente entre la comprensión del negocio y el diseño técnico del modelo de datos, ya que permite identificar los elementos clave que serán incluidos en el sistema. En el contexto del proyecto aplicado al sitio DirectorioUsme.com, esta fase se divide en cuatro etapas específicas:

Recolección de Datos

En esta etapa se identifican y organizan todas las fuentes de datos relevantes para el proyecto, especificando su origen, formato, frecuencia de actualización y métodos de acceso. Se contempla el uso de hojas de cálculo y herramientas de análisis digital como fuentes principales.

Las técnicas de recolección pueden incluir exportaciones manuales o integración mediante APIs. Asimismo, se prevé la presencia de problemas como duplicados, inconsistencias o registros incompletos, por lo que se establecerá un esquema inicial que garantice la calidad y consistencia de los datos en fases posteriores.

Descripción de Datos

Esta etapa consiste en realizar un inventario detallado de los datos disponibles, identificando las variables existentes, sus tipos (numéricos, textuales, booleanos, entre otros), formatos (CSV, Excel) y estructura general. También se analiza el rol funcional de cada conjunto de datos dentro del proyecto (por ejemplo, datos de entrada, métricas o claves), lo que permitirá estructurar el diccionario de datos, estandarizar la nomenclatura y preparar el modelo entidad-relación.

Exploración de Datos

Aquí se lleva a cabo un análisis preliminar para detectar patrones, tendencias, valores atípicos y posibles errores. Este análisis se apoya en estadísticas descriptivas y visualizaciones iniciales, utilizando herramientas como Python y bibliotecas de análisis de datos. El propósito no es obtener conclusiones definitivas, sino adquirir un entendimiento más profundo de los datos para orientar las decisiones en la fase de transformación y modelado.

Verificación de Calidad

Finalmente, se evalúa la calidad de los datos mediante la identificación de errores, valores ausentes, formatos inválidos, registros duplicados y otras inconsistencias que puedan afectar la integridad del modelo. Esta verificación permite determinar si los datos están completos, actualizados y listos para su uso. En caso de hallazgos críticos, se definirán medidas correctivas que se aplicarán en la siguiente fase de preparación.

Fase 3 Preparación de Datos

La fase de preparación de los datos es una de las más exigentes dentro del ciclo de vida de CRISP-DM, tanto en términos de tiempo como de esfuerzo. Su objetivo principal es dejar los

datos en condiciones óptimas para su uso en la fase de modelado, asegurando que estén limpios, estructurados, normalizados y alineados con los objetivos del negocio establecidos previamente.

Esta etapa implica transformar los datos brutos en una base estructurada y confiable, lista para el análisis. Las tareas comunes incluyen la selección de variables relevantes, limpieza de registros, generación de atributos derivados, integración de múltiples fuentes y estandarización de formatos (Cortina, 2015). Cabe resaltar que esta fase se encuentra estrechamente vinculada con la fase de modelado, ambas fases interactúan de manera continua, permitiendo ajustes iterativos conforme avanzan las decisiones del proyecto.

Limpieza de Datos

Esta etapa tiene como objetivo asegurar la calidad de la información, eliminando errores y redundancias que puedan afectar la consistencia del análisis. Las principales acciones incluyen: Normalización de datos: Unificación de formatos de fechas, nombres de categorías, barrios y servicios, garantizando coherencia en toda la base de datos. eliminación de duplicados: Detección y supresión de registros repetidos, producto de la integración de múltiples fuentes, corrección de errores: Ajuste de datos mal digitados, campos mezclados o formatos inválidos.

Estas actividades permitirán contar con una base confiable sobre la cual construir relaciones entre entidades y generar indicadores válidos.

Transformación de Datos

Esta etapa se enfoca en adaptar los datos a una estructura técnica que permita su uso dentro de un modelo relacional, con el fin de facilitar análisis posteriores y consultas eficientes. Las acciones para desarrollar incluyen:

Estandarización de formatos: Unificación de tipos de datos como fechas, monedas y nombres de campos.

Integración de datos dispersos: Consolidación de información proveniente de distintas fuentes (Google Sheets, Analytics, AdSense, etc.) en una base de datos unificada.

Reestructuración relacional: Organización de los datos en tablas normalizadas, con definición de claves primarias y foráneas, garantizando integridad referencial.

Filtrado de información no relevante: Eliminación de campos sin uso analítico o registros que no aporten valor al modelo.

Fase 4 Modelado

En la metodología CRISP-DM, el modelado se construye con base en la comprensión previa del negocio y de los datos, permitiendo seleccionar las técnicas más adecuadas según el tipo de problema a resolver.

(Brzozowska et al., 2023) Comenta que existen diversas técnicas de modelado para un mismo tipo de problema, y algunas de ellas requieren que los datos cumplan condiciones específicas de formato o estructura. Por ello, esta fase suele estar estrechamente vinculada con la fase de preparación de los datos, permitiendo ajustes iterativos cuando sea necesario.

Selección de Técnica de Modelado

En esta etapa se determina la técnica de modelado más adecuada según el tipo de problema de análisis y la naturaleza de los datos disponibles. La selección debe considerar tanto los objetivos del negocio como las características técnicas del conjunto de datos. Algunas técnicas requieren datos en formatos específicos o estructuras determinadas, por lo que esta fase está estrechamente relacionada con la preparación de los datos. Entre las alternativas más comunes se encuentran los modelos relacionales, de árboles de decisión, redes neuronales, y algoritmos de clasificación o agrupamiento, entre otros. Además, esta etapa contempla la definición de principios de modelado, el uso de herramientas gráficas como diagramas entidad-relación (ER), y

la elección del sistema de gestión de bases de datos o plataforma analítica en la que se implementará el modelo.

Selección de Datos de Prueba

Esta etapa consiste en definir un conjunto representativo de datos que permita evaluar el rendimiento técnico y lógico del modelo propuesto. Los datos de prueba deben reflejar las principales características del conjunto real, incluyendo distintos tipos de registros, valores comunes, atípicos y casos límite. Su propósito no es generar análisis definitivos, sino validar que el modelo funcione correctamente en cuanto a integridad referencial, relaciones entre entidades y coherencia estructural. Asimismo, los datos seleccionados deben permitir identificar posibles errores en la implementación, facilitando ajustes tempranos.

Obtención del Modelo

En esta etapa se construye el modelo seleccionado mediante un proceso iterativo que incluye diseño, implementación y ajustes sucesivos. Se generan las estructuras lógicas que representan los datos y sus relaciones, de acuerdo con las reglas y principios definidos previamente. El modelo puede representarse mediante esquemas gráficos y luego implementarse en una herramienta o sistema de gestión, donde se validan su funcionalidad, consistencia y alineación con los objetivos del proyecto. Esta fase culmina cuando se obtiene una estructura técnica estable, lista para ser evaluada y posteriormente utilizada en fases analíticas o productivas.

Fase 5 Evaluación del Modelo

En esta etapa se analiza la calidad y el desempeño del modelo relacional desarrollado, con base en métricas estadísticas y pruebas de funcionalidad. El objetivo principal es determinar

si el modelo cumple con los requisitos del proyecto y es lo suficientemente robusto para pasar a la etapa de implementación, o si requiere ajustes adicionales en etapas previas.

Evaluación de la Calidad del Modelo

Esta etapa de la metodología se enfoca en la revisión sistemática del modelo desarrollado para asegurar que cumple con los criterios de calidad, tanto a nivel estructural como funcional. Evaluar la calidad del modelo es fundamental para determinar su viabilidad técnica, su alineación con los objetivos del proyecto y su preparación para ser implementado en un entorno real. De forma general, esta evaluación considera aspectos como:

Coherencia del diseño: Se analiza la correspondencia entre el modelo desarrollado y los requerimientos definidos en las etapas iniciales.

Correcta estructuración del modelo: Se verifica que la organización interna del modelo sea clara, lógica y que respete buenas prácticas de diseño.

Capacidad para integrarse con otras herramientas: Se contempla la posibilidad de que el modelo funcione junto a procesos automatizados, sistemas externos o plataformas de análisis.

Escalabilidad: Se valora si el modelo puede adaptarse a mayores volúmenes de datos o nuevas condiciones sin requerir rediseños fundamentales.

Esta etapa es crítica dentro del ciclo de desarrollo, ya que permite decidir si el modelo conceptual o físico planteado puede ser implementado tal como está, o si requiere ajustes, simplificaciones o rediseños parciales antes de su despliegue.

Análisis Comparativo Frente a la Situación Inicial

Como parte de la evaluación estratégica, se plantea una comparación entre la situación actual del DirectorioUsme.com (basada en hojas de cálculo dispersas y sin estandarización) y el

funcionamiento del modelo diseñado. Este análisis servirá para medir el impacto del proyecto en términos de:

- Reducción del tiempo necesario para generar reportes.
- Mejora en la confiabilidad de las métricas.
- Disminución de errores por duplicidad o inconsistencia de datos.
- Agilidad en la ejecución de consultas clave.

Toma de Decisiones Según los Resultados

Dependiendo de los hallazgos de esta fase, se definirán las acciones a seguir:

Validación completa: Si el modelo cumple con los requisitos funcionales y técnicos, se procederá a su implementación final. Replanteamiento estructural: En caso de identificar fallas críticas en el diseño, se considerará rediseñar componentes clave del modelo antes de su implementación definitiva.

Fase 6 Implementación

La última fase de la metodología CRISP-DM consiste en la puesta en marcha del modelo desarrollado, trasladándolo desde un entorno de diseño y prueba hacia un entorno de producción. El objetivo de esta fase es asegurar que el sistema funcione correctamente, sea sostenible en el tiempo y cumpla con los propósitos estratégicos del negocio. Esta implementación incluye no solo el despliegue técnico, sino también la documentación y entrega final del sistema para su uso, mantenimiento y escalabilidad.

Implementación del Sistema de Gestión de Bases de Datos

Esta etapa se centra en la puesta en marcha del modelo elegido en un entorno operativo real. Abarca la instalación, configuración y activación del modelo dentro de la plataforma tecnológica previamente seleccionada. Esto incluye actividades clave como la creación de

estructuras internas, la configuración de componentes técnicos y la preparación del entorno para asegurar que el sistema pueda funcionar de manera eficiente.

Documentación del Sistema

La documentación adecuada es fundamental para garantizar la continuidad operativa del sistema, facilitar futuras mejoras y permitir su replicación o escalamiento. En esta etapa, el objetivo principal es generar los registros formales que describen el funcionamiento del sistema, facilitando su uso, mantenimiento y futuras evoluciones. La documentación se estructurará en tres niveles y podrá contemplar aspectos técnicos, operativos y de usuario, según las necesidades específicas del proyecto.

Aplicación de la Metodología CRISP-DM

Definida la metodología y establecidas las fases, se procede a aplicar cada una de ellas en el desarrollo del proyecto, siguiendo el enfoque propuesto con la metodología CRISP-DM. Esta aplicación permite abordar el problema de forma estructurada, desde la comprensión del negocio hasta la implementación técnica de la solución basada en datos.

Fase 1 Comprensión del Negocio

En esta fase inicia el desarrollo metodológico mediante la identificación de los objetivos y requerimientos del proyecto con la perspectiva del negocio, con el propósito de traducirlos posteriormente en soluciones técnicas alineadas con dichos fines.

Identificación del Problema

El sitio web tiene como propósito facilitar el acceso a la información de contacto de servicios, negocios, emprendimientos y entidades públicas en la localidad la localidad de Usme.

La información obtenida en la organización proviene de tres fuentes principales.

Negocios registrados: cada negocio obtiene una página exclusiva con datos comerciales (nombre, contacto, dirección, horarios, redes sociales, categoría, entre otros).

Información institucional: recopilada por el equipo del DirectorioUsme-com, incluyendo servicios como bancos, estaciones de policía, transporte público, entre otros.

Usuarios visitantes: quienes generan datos de navegación como tipo de dispositivo, duración de la visita, zonas más consultadas, y comportamiento por los diferentes grupos etarios (first-party data) recolectada desde herramientas como Google Analytics, Search Console y Adsense).

Situación Actual del Problema: actualmente, toda esta información se encuentra dispersa en múltiples formatos y plataformas, como Google Sheets, archivos Excel y reportes manuales.

Esta fragmentación impide consolidar, analizar o visualizar los datos de forma estratégica, y ha generado diversas problemáticas:

- Retrasos de entre 5 y 7 días para generar análisis o reportes útiles.
- Imposibilidad de detectar patrones que permitan aprovechar oportunidades comerciales o de contenido.
- Duplicidad o pérdida de información, al no contar con un sistema centralizado y con control de accesos.
- Ausencia de trazabilidad o auditoría de datos, lo que limita la evaluación de decisiones pasadas.
- Freno al crecimiento e innovación del sitio, al no contar con métricas, indicadores o metas definidas.

Consecuencia del Problema: La dispersión y falta de integración de la información compromete seriamente la escalabilidad, sostenibilidad y utilidad estratégica del DirectorioUsme.com. La plataforma no logra evolucionar como un verdadero ecosistema digital basado en datos, limitando su capacidad para generar valor social y comercial.

Teniendo en cuenta lo anterior se planteó la pregunta problema:

¿Cómo diseñar e implementar un sistema de gestión de datos que centralice el procesamiento, almacenamiento, análisis y visualización de la información del sitio web DirectorioUsme.com, optimizando la toma de decisiones y el aprovechamiento de los datos disponibles?

Determinación de Objetivos

El DirectorioUsme.com pretende consolidarse como un sitio web referente para conectar a los habitantes de la localidad de Usme con los servicios y negocios locales. Para lograrlo, es

fundamental transformar los datos recolectados desde múltiples fuentes en información consolidada, estructurada y útil para la toma de decisiones estratégicas, fortaleciendo el posicionamiento, la sostenibilidad y el valor agregado del sitio web para usuarios, negocios registrados y aliados publicitarios. Teniendo en cuenta lo anterior los objetivos principales del negocio son:

Aprovechar los datos *first-party* generados por usuarios, negocios y herramientas de análisis digital, para personalizar contenidos, construir audiencias y optimizar campañas de marketing local.

Identificar las secciones más relevantes del sitio, como categorías o páginas de mayor tráfico, con el fin de priorizar espacios estratégicos para la visibilidad de negocios o la ubicación de elementos publicitarios. Fortalecer el posicionamiento SEO del sitio, mediante el análisis del rendimiento orgánico, consultas frecuentes y palabras clave con mejor desempeño, asegurando la presencia constante en buscadores. Implementar un modelo de publicidad garantizada, basado en métricas confiables como usuarios únicos, CTR por categoría o por temporada, e impresiones verificadas.

Aprovechar las temporalidades clave del año, como campañas escolares, navidad o eventos locales, para destacar negocios o servicios asociados, lo cual requiere una estructura de datos categorizada, trazable y flexible.

Estos objetivos refuerzan la necesidad de contar con un sistema de gestión de datos robusto, automatizado y alineado con las necesidades del negocio, permitiendo transformar un ecosistema digital estático en una plataforma dinámica, orientada al análisis, la mejora continua y la toma de decisiones basada en evidencia.

Evaluación de la Situación Actual

Actualmente, el sitio web DirectorioUsme.com no cuenta con una infraestructura técnica que permita capitalizar el valor de los datos propios (*first-party data*) La información permanece fragmentada, dispersa en múltiples fuentes y formatos, lo cual impide su consolidación para la toma de decisiones.

Entre las principales herramientas y fuentes utilizadas hasta ahora se encuentran:

Google Sheets utilizados para clasificar categorías y almacenar datos de contacto.

Archivos Excel locales, empleados para guardar reportes históricos de campañas o registros exportados desde otras plataformas.

Herramientas de analítica digital: como Google Analytics, Search Console y Google Adsense, que entregan métricas clave de tráfico, clics e ingresos, pero que actualmente se almacenan de forma aislada y se procesan manualmente. Ausencia de un Sistema de Gestión de Bases de Datos (SGBD): no existe un repositorio centralizado ni un esquema relacional que unifique y administre los datos de forma estructurada. Falta de herramientas de visualización: actualmente no existen dashboards ni reportes automatizados, lo que obliga a realizar tareas manuales para cada análisis.

A continuación, en la Tabla 7 se muestran las diferentes limitaciones operativas actuales, junto con sus respectivas consecuencias.

Tabla 7

Limitaciones Operativas Detectadas

Dimensión	Problema concreto	Evidencia/Consecuencia
Procesos	Tareas manuales (copiar-pegar, filtrado)	5-7 días para generar un informe básico; alto riesgo de error humano
Integridad	Múltiples versiones de un mismo dato	Duplicados y valores contradictorios entre hojas
Dimensión	Problema concreto	Evidencia/Consecuencia
Seguridad	Acceso sin roles ni privilegios definidos	Datos personales expuestos; trazabilidad nula de cambios
Escalabilidad	Crecimiento del volumen de registros (≈300 y +20 % anual)	Hojas de cálculo llegan al límite de filas y reducen el rendimiento
Analítica	Métricas dispersas	Imposible calcular CTR unificado o tráfico por categoría en una sola consulta
Toma de decisiones	Falta de dashboards confiables	Se posponen o se sustentan con suposiciones las campañas de temporada

Nota. Estas son debilidades que han sido identificadas como cuellos de botella en la operación diaria y crecimiento del sitio

Línea Base para Comparación Futura: Se establecido una serie de indicadores que servirán como punto de comparación entre la situación actual y los resultados esperados tras la implementación del sistema de gestión de datos:

Tabla 8

Indicadores Base

Indicador clave	Situación actual	Meta tras la implementación
Estructura organizada de datos	Información dispersa y sin normalizar	Tablas organizadas y funcionales para realizar análisis cruzados
Ingreso de información	Manual, propenso a errores	Procesos semiautomáticos o completamente automatizados
Tiempo de generación de un reporte de tráfico	5–7 días	≤ 2 horas
Porcentaje de registros duplicados	10–15 %	< 1 %
Tiempo medio de respuesta de una consulta	> 60 s (en hojas) las búsquedas son manuales	< 7 s (en sistema gestor de base de datos)
Número de vistas/consultas automatizadas	No existe	≥ 3 vistas estratégicas
Dashboards creados de referencia	No existe	≥ 3 dashboards con actualización mensual o semanal
Existencia de control de accesos por rol	No	Sí (mínimo 2 roles definidos)

Nota. Indicadores que permitirán evaluar la implementación de este proyecto.

Este diagnóstico de la primera fase respalda la urgencia del proyecto y establece la línea base contra la cual se medirá su impacto una vez se implemente el sistema de gestión de datos relacional con procesos ETL

Fase 2 Comprensión de Datos

Entendiendo los objetivos del negocio y su situación actual respecto al manejo de los datos, se procede a aplicar la Fase 2, en la cual se realiza una caracterización y evaluación de las fuentes y tablas vinculadas al sitio web. Esto permite definir qué se debe ajustar y en qué casos es necesario aplicar un proceso de ETL (Extracción, Transformación y Carga) para resolver problemas de formato, duplicidad y datos faltantes, asegurando así la consistencia y calidad de la base de datos centralizada (Palacios Martel, 2019).

Recolección de Datos

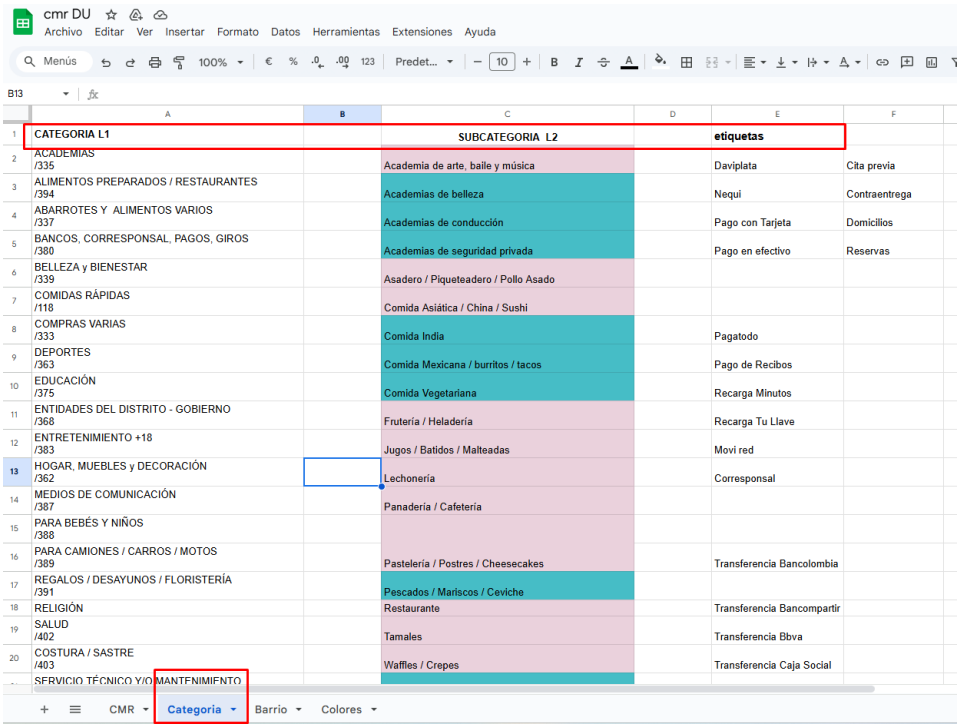
Este proceso permite comprender y evaluar cuáles fuentes y tablas están relacionadas con el sitio web, así como identificar cómo y dónde se encuentran almacenadas. Se establece un inventario inicial de los orígenes de datos, lo cual resulta clave para reconocer estructuras inconsistentes, redundancias o vacíos de información. Además, conocer la ubicación, el formato y la frecuencia de actualización de cada fuente facilita la planificación del proceso ETL, Como parte de la recolección y comprensión de datos, se utilizó una estrategia basada en dos etapas para cada fuente analizada:

Captura del estado actual, donde se presenta una figura (captura de pantalla) de la estructura original del archivo o tabla, tal como está almacenada en el sistema actual y Ficha técnica construida en una tabla de evaluación que resume la información técnica, los problemas detectados y las acciones correctivas previstas para la futura integración en la base de datos relacional.

Este enfoque se repite a lo largo de las siguientes secciones para cada fuente prioritaria.

Figura 1

Categorías, Subcategorías y Etiquetas del Directoriousme.com



cmr DU

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda

	A	B	C	D	E	F
1	**CATEGORIA L1**		**SUBCATEGORIA L2**		**etiquetas**	
2	ACADEMIAS /335		Academia de arte, baile y música		Daviplata	Cita previa
3	ALIMENTOS PREPARADOS / RESTAURANTES /394		Academias de belleza		Nequi	Contraentrega
4	ABARROTES Y ALIMENTOS VARIOS /337		Academias de conducción		Pago con Tarjeta	Domicilios
5	BANCOS, CORRESPONSAL, PAGOS, GIROS /380		Academias de seguridad privada		Pago en efectivo	Reservas
6	BELLEZA y BIENESTAR /339		Asadero / Piqueteadero / Pollo Asado			
7	COMIDAS RÁPIDAS /118		Comida Asiática / China / Sushi			
8	COMPRAS VARIAS /333		Comida India		Pagatodo	
9	DEPORTES /363		Comida Mexicana / burritos / tacos		Pago de Recibos	
10	EDUCACIÓN /375		Comida Vegetariana		Recarga Minutos	
11	ENTIDADES DEL DISTRITO - GOBIERNO /368		Frutería / Heladería		Recarga Tu Llave	
12	ENTRETENIMIENTO +18 /383		Jugos / Batidos / Malteadas		Movi red	
13	HOGAR, MUEBLES y DECORACIÓN /362		Lechonería		Corresponsal	
14	MEDIOS DE COMUNICACIÓN /387		Panadería / Cafetería			
15	PARA BEBÉS Y NIÑOS /388					
16	PARA CAMIONES / CARROS / MOTOS /389		Pastelería / Postres / Cheesecakes		Transferencia Bancolombia	
17	REGALOS / DESAYUNOS / FLORISTERÍA /391		Pescados / Mariscos / Ceviche			
18	RELIGIÓN		Restaurante		Transferencia Bancompartir	
19	SALUD /402		Tamales		Transferencia Bbva	
20	COSTURA / SASTRE /403		Waffles / Crepes		Transferencia Caja Social	
	SERVICIO TÉCNICO Y/O MANTENIMIENTO					

CMR | Categoria | Barrio | Colores

Nota. Captura de Google Sheets del DirectorioUsme.com - archivo CRM / hoja “categoría”

Tabla 9

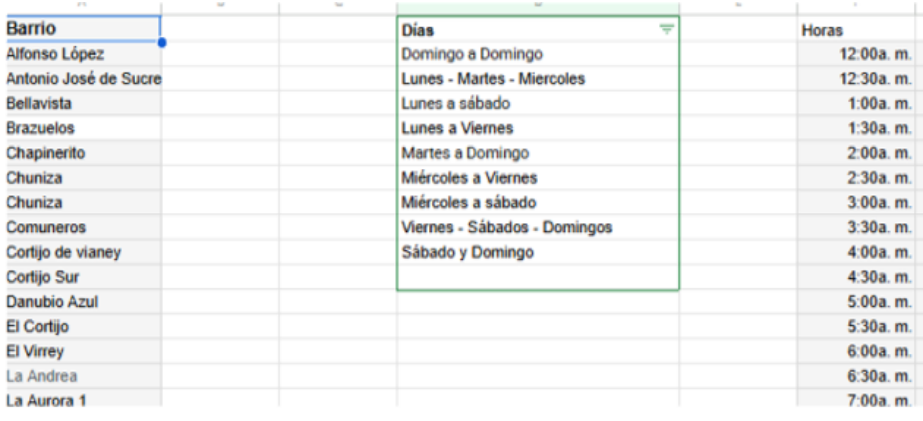
Ficha Técnica y Evaluación del Archivo CRM / Hoja "Categorías"

Campo de la ficha	Contenido
Fuente	Archivo "CRM" almacenado en Google Sheets, hoja denominada categoría
Formato	CSV, delimitado por comas, codificación UTF-8.
Variables clave	ID_Categoria_L1, Name_Categoria_L1 ID_Categoria_L1, Name_Categoria_L1 Etiquetas
Herramienta de descarga	Exportación manual vía botón Export → CSV desde la hoja activa
Problemas detectados	En una sola hoja conviven tres estructuras de datos distintas (categorías de primer nivel, segundo nivel y etiquetas), lo cual complica el análisis y dificulta su integración directa en una base de datos relacional.
Acción correctiva planificada	Durante la fase de transformación se migrarán las estructuras a hojas independientes, se renombran las columnas, se aplicarán reglas de validación entre jerarquías y se cargará en una tabla relacional normalizada en MySQL y se asignará su ID único.

Nota. Esta tabla resume los elementos técnicos, problemáticos y correctivos asociados a la fuente de datos "categorías" dentro del archivo CRM en Google Sheets, como parte del diagnóstico en la fase de comprensión de datos

Figura 2

Barrios y Rangos de Horarios Asociados a los Negocios Registrados



Barrio	Días	Horas
Alfonso López	Domingo a Domingo	12:00a. m.
Antonio José de Sucre	Lunes - Martes - Miercoles	12:30a. m.
Bellavista	Lunes a sábado	1:00a. m.
Brazuelos	Lunes a Viernes	1:30a. m.
Chapinerito	Martes a Domingo	2:00a. m.
Chuniza	Miércoles a Viernes	2:30a. m.
Chuniza	Miércoles a sábado	3:00a. m.
Comuneros	Viernes - Sábados - Domingos	3:30a. m.
Cortijo de vianey	Sábado y Domingo	4:00a. m.
Cortijo Sur		4:30a. m.
Danubio Azul		5:00a. m.
El Cortijo		5:30a. m.
El Virrey		6:00a. m.
La Andrea		6:30a. m.
La Aurora 1		7:00a. m.

Nota. Captura extraída de Google Sheets del DirectorioUsme.com del archivo CRM / hoja "Barrios"

Tabla 10

Ficha Técnica y Evaluación del Archivo CRM / Hoja "Barrios"

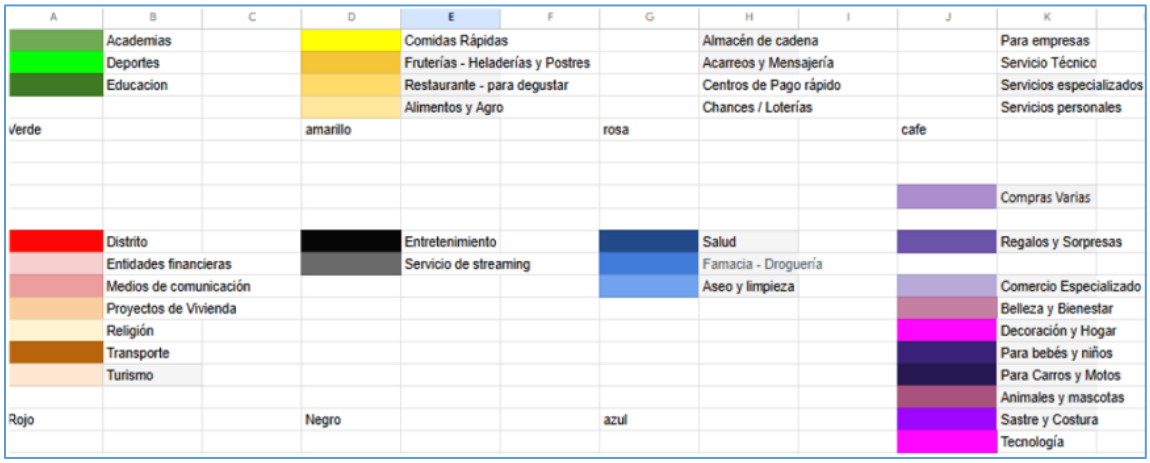
Campo de la ficha	Contenido
Fuente	Archivo "CRM" almacenado en Google Sheets, hoja denominada Barrio
Formato	CSV, delimitado por comas, codificación UTF-8.
Variables clave	Barrio, Días, Horas
Frecuencia de actualización	Mensual (día 1 de cada mes)
Herramienta de descarga	Exportación manual vía botón Export → CSV desde la hoja activa
Problemas detectados	La hoja Barrio contiene múltiples estructuras de datos sin separación clara (barrios, días de atención, horas), lo que complica el análisis y dificulta su normalización. Esta situación impide aplicar reglas de validación estructuradas, limita el análisis y no cumple con los principios de una base de datos relacional

Campo de la ficha	Contenido
Problemas detectados	Como la definición de claves primarias o la segmentación en entidades distintas.
Acción correctiva planificada	Durante la fase de transformación, estas estructuras serán migradas a hojas independientes o tablas separadas. Se asignarán claves primarias (primary keys), se aplicarán reglas de validación entre entidades (horarios, barrios), y posteriormente se integrarán en una base de datos relacional (MySQL), respetando los principios de normalización

Nota. Esta tabla resume los elementos técnicos, problemáticos y correctivos asociados a la fuente de datos “Barrios” dentro del archivo CRM en Google Sheets, como parte del diagnóstico en la fase de comprensión de datos

Figura 3

Asignación de Colores por Categoría



Nota. Captura extraída de Google Sheets del DirectorioUsme.com del archivo CRM / hoja “Colores”

Tabla 11

Ficha Técnica y Evaluación del Archivo CRM / Hoja "Colores"

Campo de la ficha	Contenido
Fuente	Archivo "CRM" almacenado en Google Sheets, hoja denominada Colores
Formato	CSV, delimitado por comas, codificación UTF-8.
Variables clave	No tiene definidas, solo celdas con texto libre; ej. "academias – verde oscuro"
Frecuencia de actualización	Semestral (1 de enero / 1 de julio) se modifica solo si se crea una categoría nueva
Herramienta de descarga	Exportación manual vía botón Export → CSV desde la hoja activa
Problemas detectados	- Falta de variables definidas (no hay columnas explícitas). - Los colores están pintados en las celdas en lugar de un código estándar (por ejemplo: #A8E05F). - No existe clave primaria que relacione la categoría con su color, no se puede hacer JOIN desde SQL o Power BI.
Acción correctiva planificada	- Eliminar la hoja colores. - Agregar a la / tabla Categoria_L1 una columna color_hex (VARCHAR 7) con el código hexadecimal oficial.

Nota. Esta tabla presenta la evaluación técnica de la hoja "Colores", la cual contiene los nombres de las categorías y sus colores asignados visualmente. Dado que no posee un formato estandarizado ni codificación hexadecimal, se plantea su transformación durante el proceso ETL para integrarla correctamente al modelo relacional y mantener consistencia visual en los reportes

Figura 4

Vista de los Registros de Negocios en la Hoja "CRM"

DATOS PERSONALES											CATEGORÍA
Nombres	Apellidos	Nombre Comercial	Dirección	Correo electrónico	Celular	Teléfono	Barrio	CATEGORIA PADRI	Subcategoría	Subcategoría	
[redacted]	[redacted]	Academia de baile y salon de eventos	Magia Internacional	[redacted]	32[redacted]		Santa Librada	ACADEMIAS	Academia de Arte, Baile	Salón de Eventos	
[redacted]	[redacted]	Colectivo Navek: corazón y alma	Calle 81 sur #9A-54	[redacted]	316[redacted]	6613174	Yomasa	ACADEMIAS	Academia de Arte, Baile		
[redacted]	[redacted]	Distribuidora de Pollo DISTRIPOLI	Cl. 69f Sur #2a-16 a 2a-6	[redacted]	313[redacted]	7	La Aurora 1	ABARROTES Y AL	Pollo y Carnes	Salsamentaria	
[redacted]	[redacted]	Supermercado SAN VICENTE	carrera 14 #74c 25 sur	[redacted]	320[redacted]	2002828	El Cortijo	ABARROTES Y AL	Minimercado		
[redacted]	[redacted]	Princes_dan_ el nayib	sin local	[redacted]	310[redacted]		Brazuelos	BELLEZA Y BIENE	Manicure / Pedicure / Ma		
[redacted]	[redacted]	Barber stiveenzon	Calle 72bis sur14v58	[redacted]	313[redacted]		Las Quintas	BELLEZA Y BIENE	Barbería	Manicure / Pedicure / Maquilladori	
[redacted]	[redacted]	Fade barber shop	Cl 136 a bis sur #14-70	[redacted]	315[redacted]		Usme Centro	BELLEZA Y BIENE	Barbería	Manicure / Pedicure / Maquilladori	
[redacted]	[redacted]	Clases particulares de excel desde	Calle 91 A sur No 3-55	[redacted]	316[redacted]		Bellavista Alta	SERVICIOS Y/O C	Cursos y/o Clases Partic		
[redacted]	[redacted]	Maestro de Construcción JS	Carrera 14g #95-25	[redacted]	311[redacted]		Montebianco	SERVICIOS Y/O C	Maestro en Costrucción		
[redacted]	[redacted]	Mariachi Elegancia Mexicana			311[redacted]		Santa Librada	SERVICIOS Y/O C	Mariachi / Grupos musico		
[redacted]	[redacted]	Service extintores	Calle 64 sur #1a -09	[redacted]	305[redacted]		La Fiscala	SERVICIOS Y/O C	Recarga de Extintores		
[redacted]	[redacted]	Tapicería Jaime	Calle 115 a 3 c 22	[redacted]	319[redacted]		Antonio José de	SERVICIOS Y/O C	Tapicería		
[redacted]	[redacted]	Tatuador - Legends_tattoo_art	Kr 14a # 74b-78 casa 2	[redacted]	315[redacted]		Miravalle	SERVICIOS Y/O C	Tatuajes / Piercings		
[redacted]	[redacted]	Empanadas Punto Amarillo	Cra 144#101a-44sur	[redacted]	312[redacted]		Montebianco	COMIDAS RÁPIDA	Arepas - Empanadas - C		
[redacted]	[redacted]	TU PIZZA YA	Cll 75#12-55 sur	[redacted]	311[redacted]		Santa Librada	COMIDAS RÁPIDA	Pizzería		
[redacted]	[redacted]	TU PIZZA YA	Carrera 9A #79-16, Bogot	[redacted]	319[redacted]	7	La Andrea	COMIDAS RÁPIDA	Pizzería		
[redacted]	[redacted]	Pizzería Mangé's pizza	CRA 14 L 71 b13 Sur	[redacted]	312[redacted]	902 - 346 7	La Aurora 2	COMIDAS RÁPIDA	Pizzería		
[redacted]	[redacted]	Pizzas brayan	Car 1 d # 92 b 03 sur	[redacted]	313[redacted]		Comuneros	COMIDAS RÁPIDA	Pizzería		

Nota. Captura extraída de Google Sheets del DirectorioUsme.com del archivo CRM / hoja "CRM" Esta hoja contiene los datos personales y comerciales registrados por los negocios a través del formulario del sitio web DirectorioUsme.com

Tabla 12

Ficha Técnica y Evaluación del Archivo CRM / Hoja "CRM"

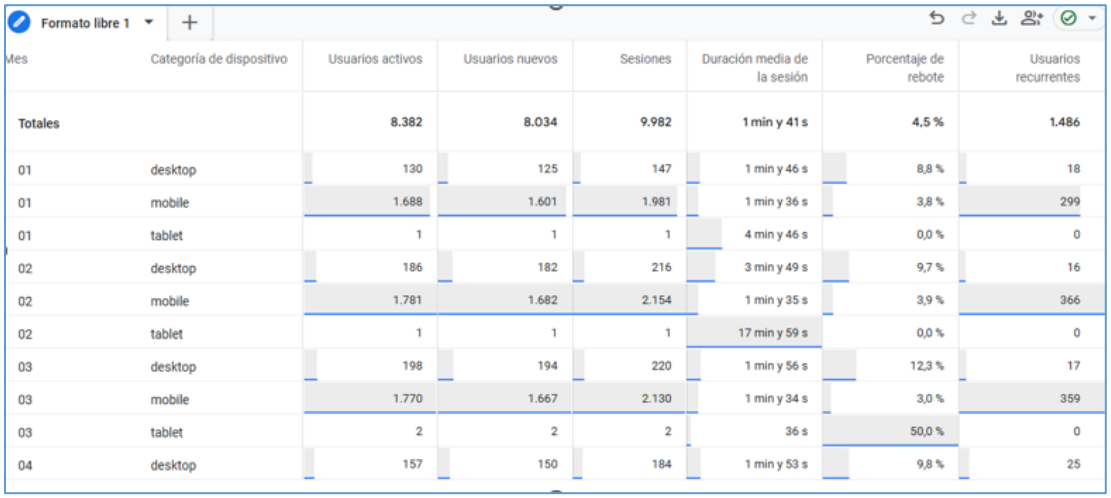
Campo de la ficha	Contenido
Fuente	Archivo "CRM" almacenado en Google Sheets, hoja denominada CRM
Formato	CSV, delimitado por comas, codificación UTF-8.
Variables clave	Datos personales (nombre, apellido, correo, celular) Datos comerciales (Nombre comercial, Celular, Barrio, Categoría, métodos de pago, horarios, enlace url)
Frecuencia de actualización	Quincenal (día 15 y 30 de cada mes)
Herramienta de descarga	Exportación manual vía botón Export → CSV desde la hoja activa
Problemas detectados	- No se dispone de un sistema de claves primarias (ID) para identificar de forma única a los negocios registrados o a las personas que realizan el registro. Las columnas están mezcladas, lo que viola los principios de normalización de datos (una hoja incluye tanto información personal como comercial).

Campo de la ficha	Contenido
Problemas detectados	- No hay trazabilidad temporal (no se registra la fecha de creación ni de modificación del registro). (hay datos redundantes y sin referencia cruzada) - No se cumple con principios de Anonimización y minimización de datos personales requeridos por la Ley 1581 de 2012. - No existen claves foráneas hacia las tablas de categoría, barrio, o métodos de pago, lo que impide relaciones y consultas eficientes. - No se registran fechas de última modificación, lo que impide llevar un control de actualizaciones y auditoría
Acción correctiva planificada	Durante la fase de transformación, se aplicará un proceso de normalización de los datos para dividir esta hoja en al menos tres tablas independientes: 1) persona_registro: donde se almacenarán datos anonimizados de la persona que realiza el registro, incluyendo un id_persona como clave primaria. 2) negocio: tabla con el id_negocio, nombre, barrio, categoría, horarios, descripción, formas de pago y URL personalizada, enlazada por una clave foránea a persona_registro y a categoría. 3)categoría y subcategoria: tablas de referencia ya existentes a las que se conectará cada negocio por su identificador. Se generará también una columna fecha_registro (extraída o estimada) y se agregará fecha_ultima_actualizacion. Los nombres de los propietarios serán anonimizados mediante codificación hash u otra técnica, y los correos/teléfonos serán almacenados con restricción de acceso solo para los administradores con rol autorizado.

Nota. Esta tabla presenta las características técnicas y evaluación de la hoja “CRM”, que consolida la información de negocios registrados en el sitio web, incluyendo datos personales, comerciales y categorías asignadas. Esta estructura requiere transformación para cumplir con principios de normalización, integridad y Anonimización de datos personales

Figura 5

Informe Mensual del Comportamiento de los Usuarios



Formato libre 1

Mes	Categoría de dispositivo	Usuarios activos	Usuarios nuevos	Sesiones	Duración media de la sesión	Porcentaje de rebote	Usuarios recurrentes
Totales		**8.382**	**8.034**	**9.982**	**1 min y 41 s**	**4,5 %**	**1.486**
01	desktop	130	125	147	1 min y 46 s	8,8 %	18
01	mobile	1.688	1.601	1.981	1 min y 36 s	3,8 %	299
01	tablet	1	1	1	4 min y 46 s	0,0 %	0
02	desktop	186	182	216	3 min y 49 s	9,7 %	16
02	mobile	1.781	1.682	2.154	1 min y 35 s	3,9 %	366
02	tablet	1	1	1	17 min y 59 s	0,0 %	0
03	desktop	198	194	220	1 min y 56 s	12,3 %	17
03	mobile	1.770	1.667	2.130	1 min y 34 s	3,0 %	359
03	tablet	2	2	2	36 s	50,0 %	0
04	desktop	157	150	184	1 min y 53 s	9,8 %	25

Nota.: Captura de la herramienta Google Analytics del 1 de mayo al 22 de mayo de 2025 del "Informe mensual del comportamiento de los usuarios" La imagen muestra métricas clave como usuarios activos, usuarios nuevos, sesiones y porcentaje de rebote organizadas por mes y categoría de dispositivo. Esta estructura permite analizar el comportamiento de los usuarios

Tabla 13

Ficha Técnica del Informe Mensual del Comportamiento de los Usuarios

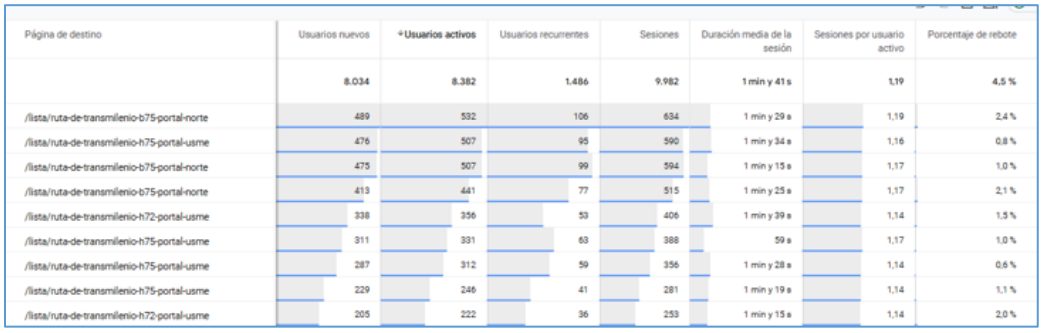
Campo de la ficha	Contenido
Fuente	Informes personalizados de Google Analytics 4 → Exploración "KPIs x Mes". Exportados como archivos individuales (uno por mes) desde la vista que muestra métricas por categoría de dispositivo (mobile, desktop, tablet).
Formato	CSV, delimitado por comas, codificación UTF-8.
Variables clave	Categoría dispositivo, usuarios activos, nuevos, sesiones, porcentaje de rebote, mes.

Campo de la ficha	Contenido
Herramienta de descarga	Exportación manual desde Google Analytics 4 mediante botón Export → CSV desde la exploración activa.
Problemas detectados	- Actualmente, no se tiene una carpeta estructurada para almacenar los archivos históricos por mes, lo que dificulta su control, versión y automatización. - El informe descargado no incluye automáticamente el nombre del mes; esta información debe ser agregada manualmente como una columna adicional antes del análisis o carga a la base de datos. - Si se renombra el archivo o se omite el mes en el nombre, se pierde trazabilidad.
Campo de la ficha	Contenido
Acción correctiva planificada	- Estandarizar el formato del nombre del archivo como analytics_KPIs_YYYY-MM.csv. - Agregar la columna mes en el script de preparación de datos (pandas) antes de cargarlos a la base de datos. - Consolidar todos los archivos mensuales en una tabla única dentro del sistema de gestión de bases de datos, permitiendo análisis históricos y comparativos entre dispositivos y períodos. - Documentar el proceso para que pueda repetirse o escalarse (incluso en futuro con Google Analytics Data API v1).

Nota. La tabla presenta las características técnicas y operativas del "Informe mensual del comportamiento de los usuarios", el cual se construye a partir de datos extraídos directamente de Google Analytics 4 mediante la funcionalidad de Exploraciones personalizadas

Figura 6

Reporte del Comportamiento Páginas Vistas por Mes



Página de destino	Usuarios nuevos	↓ Usuarios activos	Usuarios recurrentes	Sesiones	Duración media de la sesión	Sesiones por usuario activo	Porcentaje de rebote
	8.034	8.382	1.486	9.982	1 min y 41 s	1,19	4,5 %
/lista/ruta-de-transmilenio-b75-portal-norte	489	532	106	634	1 min y 29 s	1,19	2,4 %
/lista/ruta-de-transmilenio-h75-portal-usme	476	507	95	590	1 min y 34 s	1,16	0,8 %
/lista/ruta-de-transmilenio-b75-portal-norte	475	507	99	594	1 min y 15 s	1,17	1,0 %
/lista/ruta-de-transmilenio-b75-portal-norte	413	441	77	515	1 min y 25 s	1,17	2,1 %
/lista/ruta-de-transmilenio-h72-portal-usme	338	356	53	406	1 min y 39 s	1,14	1,5 %
/lista/ruta-de-transmilenio-h75-portal-usme	311	331	63	388	59 s	1,17	1,0 %
/lista/ruta-de-transmilenio-h75-portal-usme	287	312	59	356	1 min y 28 s	1,14	0,6 %
/lista/ruta-de-transmilenio-h75-portal-usme	229	246	41	281	1 min y 19 s	1,14	1,1 %
/lista/ruta-de-transmilenio-h72-portal-usme	205	222	36	253	1 min y 15 s	1,14	2,0 %

Nota. Captura de la herramienta Google Analytics del informe creado para visualizar el comportamiento de las páginas mensualmente permitiendo identificar cuáles generan mayor tráfico, evaluar el desempeño de categorías o negocios específicos y establecer prioridades para la ubicación de contenido estratégico o publicitario

Tabla 14

Ficha Técnica del Informe del Comportamiento Páginas Vistas por Mes

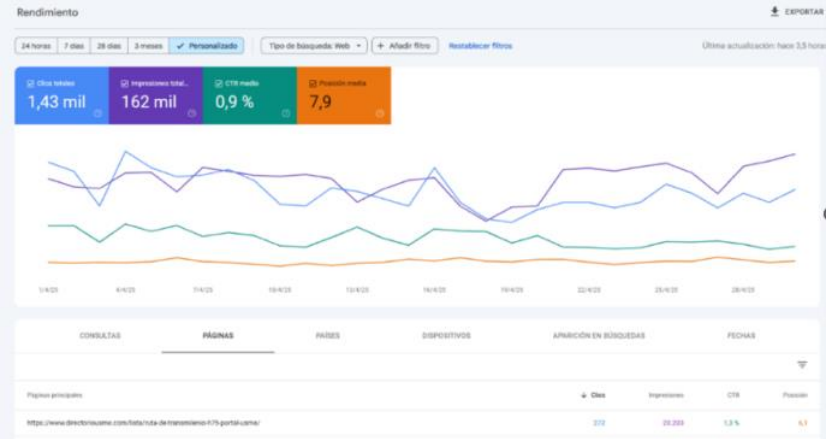
Campo de la ficha	Contenido
Fuente	Informes personalizados de Google Analytics 4 → Exploración "Páginas x Mes". Exportado como archivo individual (uno por mes) desde la vista que muestra métricas por página.
Campo de la ficha	Contenido
Formato	CSV, delimitado por comas, codificación UTF-8.
Variables clave	Página destino, Usuarios nuevos, activos, recurrentes, sesiones, duración, Sesiones por usuario, porcentaje de rebote.
Frecuencia de actualización	Mensual (día 5 de cada mes, una vez cerrado el mes anterior)

Campo de la ficha	Contenido
Herramienta de descarga	Exportación manual desde Google Analytics 4 mediante botón Export → CSV desde la exploración activa.
Problemas detectados	- El informe descargado no incluye automáticamente el nombre del mes; esta información debe ser agregada manualmente como una columna adicional antes del análisis o carga a la base de datos. - Si se renombra
Acción correctiva planificada	- Estandarizar el formato del nombre del archivo como analytics_PaginaDestino_YYYY-MM.csv. - Automatizar la inclusión de la columna mes en el script de preparación de datos (pandas) antes de cargarlos a la base de datos. - Consolidar todos los archivos mensuales en una tabla única dentro del sistema de gestión de bases de datos, permitiendo análisis históricos y comparativos entre dispositivos y períodos. - Documentar el proceso para que pueda repetirse o escalarse (incluso en futuro con Google Analytics Data API v1).

Nota. Esta tabla presenta las características técnicas del informe "*páginas vistas por mes*", que contiene métricas clave de rendimiento por URL. Esta fuente permite analizar la evolución del tráfico por página, detectar secciones con bajo desempeño, y tomar decisiones para mejorar la experiencia del usuario o la estrategia publicitaria del sitio

Figura 7

Métricas Mensuales de Tráfico Orgánico



Nota. Captura de la herramienta Google Search Console de las métricas generadas por el comportamiento orgánico de los usuarios, métricas relevantes como las impresiones en el Search de Google, clics, CTR y la posición media del sitio web. Esto permite entender como encuentran el sitio web y tomar decisiones en las estrategias de contenido SEO

Tabla 15

Ficha Técnica de las Métricas Mensuales de Tráfico Orgánico

Campo de la ficha	Contenido
Fuente	Exportaciones mensuales desde Google Search Console → Menú Rendimiento → Exportar → Hojas de cálculo/CSV. Cada export incluye cinco hojas: Consultas, Páginas, Países, Dispositivos, Fechas.
Variables clave	Todas las hojas contienen las métricas: clicks, impressions, ctr, position. • Consultas: consulta_texto, • Páginas: url, • Países: país, • Dispositivos: dispositivo • Fechas: fecha

Campo de la ficha	Contenido
Frecuencia de actualización	Mensual (día 3 de cada mes).
Herramienta de descarga	Exportación manual (o futura API Search Analytics) y guardado en /data/raw/search_console/.
Campo de la ficha	Contenido
Problemas detectados	Las métricas vienen agregadas por dimensión individual, sin posibilidad de cruces nativos. - Si se usa la exportación “Excel”, hay que descomponerla en CSV por hoja. - No existe columna mes; la fecha está implícita en el nombre del archivo. - Posibles registros duplicados si se repite la exportación.
Acción correctiva planificada	Guardar cada hoja en CSV nombrado search_console {dimension}_YYYY-MM.csv. 2. Crear tablas Stalin para cada dimensión. 3. Normalizar a un esquema (tablas intermedias) 4. Registrar la fecha de extracción y usarla para auditoría.

Nota. Esta tabla describe la estructura que tendría un reporte descargado desde Google Search Console, con indicadores mensuales como clics, impresiones, CTR y posición promedio. Esta fuente permite analizar el comportamiento orgánico del sitio DirectorioUsme.com en buscadores, identificar palabras clave efectivas y evaluar la visibilidad de las páginas más relevantes

Figura 8

Métricas Mensuales por Ads

BLOQUE DE ANUNCIOS	↓ Impresiones	Clics	CTR
Todo	**13.947**	**53**	**0,38 %**
Media	*232*	*0*	–
Negocio_1_Display	5.895	38	0,64 %
Negocio_1_1_Display	4.139	11	0,27 %
Negocio_2_Feed	2.374	4	0,17 %
Negocio_2_1_Feed	1.075	0	0,00 %

Nota. Captura de Google Adsense de métricas e ingresos generados en el sito web

Tabla 16

Ficha Técnica de Métricas e Ingresos Mensuales por Ads

Campo de la ficha	Contenido
Fuente	Plataforma Google AdSense → Informe personalizado: desglose por Unidad de anuncios y Fecha. Exportado manualmente.
Campo de la ficha	Contenido
Formato	CSV, delimitado por comas, codificación UTF-8.
Variables clave	anuncio, fecha, impresiones, impresiones, CTR. (se reserva información e ingresos)
Herramienta de descarga	Exportación manual (o futura API Search Analytics) y guardado en /data/raw/search_console/.

Campo de la ficha	Contenido
Problemas detectados	- Las fechas se muestran en diferente formato (ej. "22 may 2025" vs. "May 22, 2025"). - El informe no incluye una clave única por fila; hay que crear un ID compuesto (unidad_anuncio + fecha).
Problemas detectados	- No hay relación directa entre el bloque AdSense y una página del sitio web (page_slug). - Normalizar el nombre de cada unidad de anuncio y establecer un id_unidad_anuncio fijo para seguimiento.
Acción correctiva planificada	- Convertir los ingresos a tipo decimal y eliminar prefijo “USD” al importar (pandas. replace). Dejar el prefijo en una columna aparte - Asegurar formato YYYY-MM-DD para la fecha. - Crear una tabla de hechos en la base de datos, con claves foráneas.

Nota. Esta tabla presenta los indicadores clave extraídos de los informes mensuales de Google AdSense, incluyendo ingresos estimados, impresiones, clics en anuncios y RPM. Estos datos permiten evaluar el desempeño publicitario del sitio DirectorioUsme.com, identificar temporalidades con mayor monetización

Descripción de Datos

Una vez completado el proceso de comprensión de datos, en el cual se confirmaron las fuentes y tablas disponibles en el sistema del DirectorioUsme.com, se procede a describir los datos recolectados en términos de tipo, formato y volumen.

Tipos de datos: Como parte del diagnóstico técnico, se realizó la identificación y categorización de los principales tipos de datos presentes en las distintas fuentes utilizadas por el proyecto. Esta clasificación se basa en la tipología de datos compatibles en diferentes herramientas de gestión de base de datos, con el objetivo de garantizar un almacenamiento eficiente, coherente y normalizado.

Tabla 17

Tipos de Datos

Tipo de dato (MySQL)	Descripción	Ejemplo	Fuente ejemplo
INT	Entero con o sin signo, se usa para IDs, contadores o claves primarias.	id_categoria, clics, impresiones	Google Analytics, Search Console
BOOLEAN	Almacena valores 0 (falso) o 1 (verdadero). En MySQL es un alias de TINYINT (1).	tiene_domicilio, reserva	CRM – Etiquetas
FLOAT	Números con decimales de precisión media. Útil para métricas como CTR o tasas.	ctr, rpm, Porcentaje_Rebote	Analytics, Search Console
DOUBLE	Similar a FLOAT, pero con mayor precisión.	duracion_promedio	Google Adsense, Analytics
Tipo de dato (MySQL)	Descripción	Ejemplo	Fuente ejemplo

Tipo de dato (MySQL)	Descripción	Ejemplo	Fuente ejemplo
VARCHAR(n)	Cadena de texto de longitud variable, hasta n caracteres. Ideal para nombres, correos, barrios, etc.	nombre_negocio, correo, barrio	CRM
TEXT	Almacena grandes cantidades de texto. Útil para descripciones largas.	descripcion_servicio	CRM
DATE	Almacena solo fecha (formato YYYY-MM-DD).	fecha_registro, fecha_visita	CRM, Analytics
Tipo de dato (MySQL)	Descripción	Ejemplo	Fuente ejemplo
DATETIME	Almacena fecha y hora (formato YYYY-MM-DD HH:MM:SS).	creado_en, actualizado_en	CRM, eventos de tráfico
TIMESTAMP	Similar a DATETIME, con actualización automática si se configura.	modificado_en	Logs, bitácoras técnicas
YEAR	Almacena solo el año (entre 1901 y 2155).	año_reporte	Analytics, Adsense
ENUM	Lista de valores posibles. Útil para campos cerrados como tipo de dispositivo o método de pago.	móvil', 'escritorio', 'tableta'	Analytics, CRM
CHAR(n)	Texto de longitud fija. para códigos o indicadores estandarizados.	cod_categoria, sexo (M/F)	CRM

Nota. Esta tabla presenta los tipos de datos definidos en MySQL que se utilizarán en el diseño del modelo relacional. La selección se basa en la naturaleza de las variables recolectadas desde diversas fuentes como el CRM, Google Analytics y Adsense, con el fin de garantizar precisión, eficiencia en consultas y compatibilidad con los procesos de modelado y análisis posteriores

Formato de los Datos: Los datos se obtienen y almacenan en los siguientes formatos, varían de acuerdo con la herramienta o fuente.

Tabla 18

Formatos de los Archivos de Datos Utilizados

Fuente	Formato de archivo	Detalles
Archivos de Google Sheets	.csv, .xlsx	Exportado manualmente, codificación UTF-8
Google Analytics	.xlsx, .csv	Informes descargados mensualmente
Google Search Console	.xlsx	Datos agrupados por consultas, país, etc.
Google Adsense	.csv	Reporte mensual de ingresos y clics

Nota. Esta tabla describe los formatos de archivo empleados para almacenar y procesar los datos extraídos de las fuentes del proyecto. Incluye detalles sobre el tipo de archivo, su clasificación

Volumen Aproximado de Datos

Tabla 19

Volumen Estimado de Registros por Fuente de Datos

Fuente	Registros estimados	Observaciones
CRM	~500 registros	Negocios registrados en el sitio
Categorías	~30 categorías	Incluye categorías de nivel 1 (L1), nivel 2 (L2)
Etiquetas	Variable	Métodos de pago, servicios extra y beneficios comerciales
Barrios	~40 barrios	Datos estables, no sujetos a cambios frecuentes
Horarios	Variable	Rangos de días y horas de atención declarados por negocio

Fuente	Registros estimados	Observaciones
Google Analytics (usuarios)	+ 2.000 usuarios al mes	Datos del comportamiento de usuarios
Google Analytics (páginas)	~135 registros al mes	Datos del tráfico de las páginas
Search Console (KPIs)	Variable	Clics, impresiones, CTR y posición promedio varían de acuerdo con el tráfico orgánico mes x mes
Google AdSense	~3 registros al mes	Ingresos, clics, RPM por mes

*Nota***.** Esta tabla resume la cantidad aproximada de registros disponibles por cada fuente de datos utilizada en el proyecto. La estimación de volumen permite anticipar requerimientos de almacenamiento, definir la estructura del modelo relacional y establecer criterios para escalar el sistema a futuro conforme crece el DirectorioUsme.com

Exploración de Datos

En esta etapa se realiza un análisis preliminar de los datos recolectados, mediante la aplicación de técnicas de estadística descriptiva orientadas a identificar la estructura general del conjunto de datos, detectar posibles inconsistencias y/o patrones relevantes. Entre los procedimientos aplicados se encuentran el conteo de valores únicos, la identificación de registros nulos, la detección de duplicados y el cálculo de métricas como promedios, frecuencias y distribuciones por categoría.

Estos análisis permiten anticipar transformaciones futuras y contribuyen a garantizar la calidad de los datos y la consistencia de las tablas que se integrarán al sistema.

Los archivos utilizados en esta exploración fueron previamente descargados desde Google Drive, organizados en una carpeta local denominada *data_cruda/* y procesados a través de la herramienta *Jupyter Notebook,* utilizando bibliotecas como *pandas, numpy y matplotlib.*

A continuación, se describe el proceso de carga inicial del archivo principal CMR.xlsx, el cual contiene múltiples hojas correspondientes a los distintos componentes del DirectorioUsme.com.

Figura 9

Importación del Archivo CMR en Jupyter

```
import pandas as pd
```

```
# ruta del archivo
ruta_archivo = r"C:\Users\Usuario\Desktop\UNAD\5_Proyecto de Grado\DATA_CRUDA\CMR.xlsx"

# Carga del archivo de Excel
archivo = pd.ExcelFile(ruta_archivo)

# Mostrar los nombres de las hojas del archivo
print(archivo.sheet_names)
```

```
['CMR', 'Categoria', 'Barrio', 'Colores']
```

Nota. Código utilizado para la importación del archivo Excel “CMR” en la herramienta Jupyter y la visualización de las hojas que se encuentran en este archivo

Exploración Inicial – Archivo CRM / hoja “categorías”

Figura 10

Exploración Archivo CRM / Hoja “Categorías”

```
# Inicio del analisis hoja "categoria"
df_categoria = archivo.parse('Categoria')
df_categoria.columns
```

```
Index(['CATEGORIA L1', 'Unnamed: 1', 'SUBCATEGORIA  L2', 'Unnamed: 3',
       'etiquetas', 'Unnamed: 5'],
      dtype='object')
```

```
# vista primeras filas
df_categoria.head()
```

	CATEGORIA L1	Unnamed: 1	SUBCATEGORIA L2	Unnamed: 3	etiquetas	Unnamed: 5
0	ACADEMIAS\n/335	NaN	Academia de arte, baile y música	NaN	Daviplata	Cita previa
1	ALIMENTOS PREPARADOS / RESTAURANTES\n/394	NaN	Academias de belleza	NaN	Nequi	Contraentrega
2	ABARROTES Y ALIMENTOS VARIOS\n/337	NaN	Academias de conducción	NaN	Pago con Tarjeta	Domicilios
3	BANCOS, CORRESPONSAL, PAGOS, GIROS\n/380	NaN	Academias de seguridad privada	NaN	Pago en efectivo	Reservas
4	BELLEZA y BIENESTAR\n/339	NaN	Asadero / Piqueteadero / Pollo Asado	NaN	NaN	NaN

```
# Contar valores únicos por columna
print("Valores únicos por columna:\n", df_categoria.nunique())
```

```
Valores únicos por columna:
 CATEGORIA L1         25
Unnamed: 1            0
SUBCATEGORIA  L2    201
Unnamed: 3            0
etiquetas            16
Unnamed: 5            4
dtype: int64
```

```
# Análisis de campos vacíos o mal estructurados
print("Cantidad de valores nulos por columna:\n", df_categoria.isnull().sum())
```

```
Cantidad de valores nulos por columna:
 CATEGORIA L1        177
Unnamed: 1           202
SUBCATEGORIA  L2      1
Unnamed: 3           202
etiquetas            186
Unnamed: 5           198
dtype: int64
```

```
# Detección de duplicados
duplicados = df_categoria.duplicated().sum()
print(f"Filas duplicadas encontradas: {duplicados}")
```

```
Filas duplicadas encontradas: 0
```

Nota. Código utilizado en la herramienta Jupyter para explorar en la hoja de “categoria” columnas, valores únicos, valores mal estructurados o duplicados

La primera hoja analizada que es nombrada "categorías", donde se concentra la jerarquía de clasificación del sistema de las diferentes categorías de primer nivel (L1), y de las subcategorías (L2) y etiquetas asociadas. El análisis permitió identificar las siguientes características:

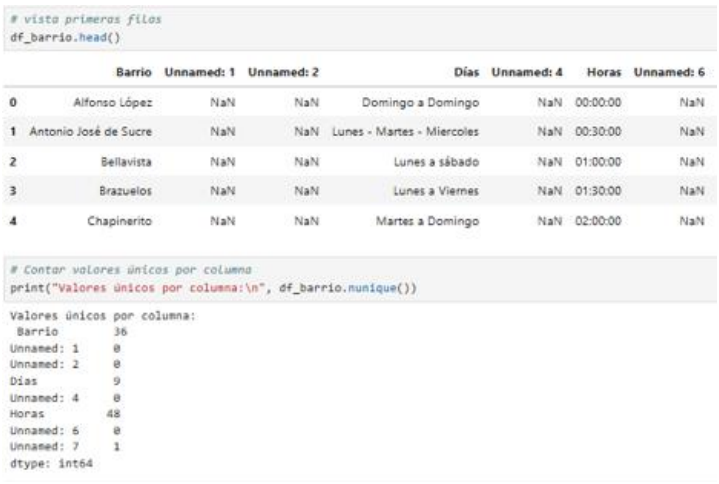
- Total, de categorías L1 únicas: 25
- Total, de subcategorías L2 únicas: 201
- Total, de etiquetas únicas: 16
- Filas duplicadas encontradas: 0
- Presencia significativa de valores nulos en campos de categorías (L1), y etiquetas
- Columnas vacías o innecesarias detectadas (Unnamed: 1, Unnamed: 3, Unnamed:
- Estos hallazgos indican que la hoja confirma lo visto en la etapa de recolección de

datos que esta hoja contiene múltiples estructuras combinadas, lo que dificulta su uso directo en una base de datos relacional. Por tanto, se plantea como acción correctiva su separación en tres entidades independientes, respetando la jerarquía y aplicando reglas de integridad referencial.

Exploración inicial – Archivo CRM / hoja “barrio”

Figura 11

Exploración Archivo CRM / Hoja “Barrio”



```
# vista primeras filas
df_barrio.head()
```

	Barrio	Unnamed: 1	Unnamed: 2	Días	Unnamed: 4	Horas	Unnamed: 6
0	Alfonso López	NaN	NaN	Domingo a Domingo	NaN	00:00:00	NaN
1	Antonio José de Sucre	NaN	NaN	Lunes - Martes - Miercoles	NaN	00:30:00	NaN
2	Bellavista	NaN	NaN	Lunes a sábado	NaN	01:00:00	NaN
3	Brazuelos	NaN	NaN	Lunes a Viernes	NaN	01:30:00	NaN
4	Chapinerito	NaN	NaN	Martes a Domingo	NaN	02:00:00	NaN

```
# Contar valores únicos por columna
print("Valores únicos por columna:\n", df_barrio.nunique())
```

```
Valores únicos por columna:
 Barrio        36
Unnamed: 1     0
Unnamed: 2     0
Días           9
Unnamed: 4     0
Horas         48
Unnamed: 6     0
Unnamed: 7     1
dtype: int64
```

Nota. Código utilizado en la herramienta Jupyter para explorar en la hoja de “Barrio” columnas, valores únicos, valores mal estructurados o duplicados

La hoja “barrios” del archivo CMR.xlsx contiene información geográfica y operativa de los sectores que conforman la localidad de Usme, incluyendo los nombres de los barrios, los días de atención y los rangos horarios declarados por los negocios registrados.

Resultados del análisis:

- Total, de barrios únicos identificados: 36
- Total, de combinaciones de horarios distintos: 9
- Total, de horas únicas: 48
- Cantidad de filas duplicadas: 0
- Campos con mayor cantidad de valores nulos:
- Barrio: 22 nulos
- Días: 49 nulos
- Horas: 10 nulos

Adicionalmente, se identificaron múltiples columnas sin etiquetar (Unnamed), las cuales carecen de información útil, creadas para espaciar atributos, lo que es una mala práctica. Asimismo, se detectó una mezcla de dimensiones en la misma hoja (como geografía para el barrio y horarios de atención), lo que genera la presencia de valores nulos en los campos correspondientes, indicando registros incompletos o una estructura inadecuada que debería ser abordada mediante la separación de estas entidades.

Exploración inicial – Archivo CRM / hoja “Colores”

La hoja de cálculo “colores” presenta una estructura no estandarizada, en la cual la relación entre cada categoría y su color asociado está representada únicamente mediante el formato de relleno de celda, sin estar codificada de manera explícita como texto o valor hexadecimal. Esta condición impide realizar un análisis exploratorio técnico automatizado, ya

que los colores no pueden ser interpretados ni procesados directamente por herramientas como Excel, Python o SQL.

No obstante, se deja constancia de que esta información será recuperada y transformada durante la fase de transformación (Fase 3), con el fin de integrarla correctamente a la futura tabla de categorías L1. Allí se añadirá una variable adicional denominada color_categoria, que permitirá mantener una línea gráfica coherente para análisis y visualizaciones posteriores.

Exploración inicial – Archivo CRM / hoja “CRM”

Figura 12

Exploración Archivo CRM / Hoja “CRM” Parte 1

```
# Inicio del análisis hoja "Barrio"
df_cmr = archivo.parse('CMR', header=1)

# Mostrar los nombres de las columnas
print(df_cmr.columns)
```

```
Index(['Nombres', 'Apellidos', 'Cedula', 'Nombre Comercial', 'Dirección',
       'Correo electrónico', 'Celular ', 'Celular 2', 'Teléfono', 'Barrio',
       'CATEGORIA PADRE', 'Subcategoria', 'Subcategoria.1', 'Subcategoria.2',
       'Subcategoria.3', 'Subcategoria.4', 'Unnamed: 16', 'Unnamed: 17',
       'Unnamed: 18', 'Unnamed: 19', 'Pago en efectivo', 'Pago con Tarjeta',
       'Domicilios', 'Daviplata', 'Nequi', 'Corresponsal', 'Pago de Recibos',
       'Pagatodo', 'Recarga de Minutos', 'Reservas', 'Cita previa',
       'Transferencia', 'Transferencia.1', 'Recarga Tu Llave', 'Contraentrega',
       'Movid Red', 'Dias ', 'Apertura', 'Cierre', 'Observacion',
       'Unnamed: 40', 'Daviplata.1', 'Domicilios.1', 'Nequi.1',
       'Pago con Tarjeta.1', 'Pago con Efectivo', 'Pagatodo.1', 'Reservas.1',
       'Cita previa.1', 'Corresponsal.1', 'Transferencias', 'Recarga Minutos',
       'Recarga Tu Llave.1', 'Pago de Recibos.1', 'Contraentrega.1',
       'Movi red', 'Horario', 'Url', 'Fecha', 'Unnamed: 59', 'Unnamed: 60'],
      dtype='object')
```

```
filas, columnas = df_cmr.shape
print(f"El DataFrame tiene {filas} filas y {columnas} columnas.")
```

```
El DataFrame tiene 63 filas y 61 columnas.
```

```
# vista primeras filas
df_cmr.head(2)
```

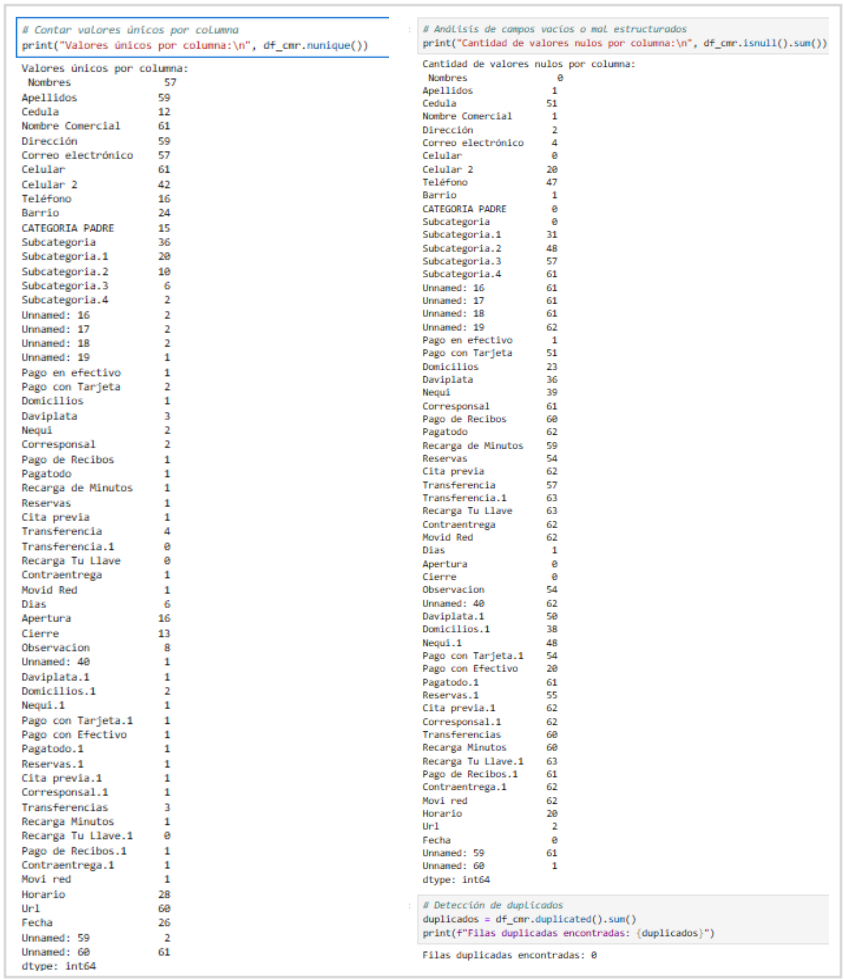
	Nombres	Apellidos	Cedula	Nombre Comercial	Dirección	Correo electrónico	Celular	Celular 2	Teléfono	Barrio	CATEGORIA PADRE	Subcategoria	Subcategoria.1	Su
0	Olga Rocio	Salas	NaN	Academia de baile y salon de eventos Magia Int...	NaN	magiainternacional@hotmail.com	320 - 828 40 33\n	311 - 472 42 41	NaN	Santa Librada	ACADEMIAS	Academia de Arte, Baile y musica	Salón de Eventos	
1	Sandra	Gómez Muñoz	NaN	Colectivo Navek corazón y alma	Calle 81 sur #9A-54	navek.danza@gmail.com	316 - 332 25 62	310 - 314 67 87	6613174	Yomasa	ACADEMIAS	Academia de Arte, Baile y musica	NaN	

Nota. Código utilizado en la herramienta Jupyter para identificar las variables de la hoja “CRM” y conocer la cantidad de variables y filas

La hoja **“**CRM” representa la base de datos principal de negocios registrados en DirectorioUsme.com. Contiene tanto información personal (nombres, teléfonos, correos) como información comercial (nombre del negocio, dirección, categoría, subcategorías, métodos de pago, servicios adicionales, horarios y fechas de registro). Se aplicó un análisis exploratorio estructural mediante Python y pandas, con el objetivo de evaluar la calidad de los datos y su nivel de organización.

Figura 13

Exploración Archivo CRM / Hoja “CRM” Parte 2



```
# Contar valores únicos por columna
print("Valores únicos por columna:\n", df_cmr.nunique())
```

```
Valores únicos por columna:
 Nombres                57
Apellidos               59
Cedula                  12
Nombre Comercial        61
Dirección               59
Correo electrónico      57
Celular                 61
Celular 2               42
Teléfono                16
Barrio                  24
CATEGORIA PADRE         15
Subcategoria            36
Subcategoria.1          20
Subcategoria.2          10
Subcategoria.3           6
Subcategoria.4           2
Unnamed: 16              2
Unnamed: 17              2
Unnamed: 18              2
Unnamed: 19              1
Pago en efectivo         1
Pago con Tarjeta         2
Domicilios               1
Daviplata                3
Nequi                    2
Corresponsal             2
Pago de Recibos          1
Pagatodo                 1
Recarga de Minutos       1
Reservas                 1
Cita previa              1
Transferencia            4
Transferencia.1          0
Recarga Tu Llave         0
Contraentrega            1
Movid Red                1
Dias                     6
Apertura                16
Cierre                  13
Observacion              8
Unnamed: 40              1
Daviplata.1              1
Domicilios.1             2
Nequi.1                  1
Pago con Tarjeta.1       1
Pago con Efectivo        1
Pagatodo.1               1
Reservas.1               1
Cita previa.1            1
Corresponsal.1           1
Transferencias           3
Recarga Minutos          1
Recarga Tu Llave.1       0
Pago de Recibos.1        1
Contraentrega.1          1
Movi red                 1
Horario                 28
Url                     60
Fecha                   26
Unnamed: 59              2
Unnamed: 60             61
dtype: int64
```

```
# Análisis de campos vacíos o mal estructurados
print("Cantidad de valores nulos por columna:\n", df_cmr.isnull().sum())
```

```
Cantidad de valores nulos por columna:
 Nombres                 0
Apellidos                1
Cedula                  51
Nombre Comercial         1
Dirección                2
Correo electrónico       4
Celular                  0
Celular 2               20
Teléfono                47
Barrio                   1
CATEGORIA PADRE          0
Subcategoria             0
Subcategoria.1          31
Subcategoria.2          48
Subcategoria.3          57
Subcategoria.4          61
Unnamed: 16             61
Unnamed: 17             61
Unnamed: 18             61
Unnamed: 19             62
Pago en efectivo         1
Pago con Tarjeta        51
Domicilios              23
Daviplata               36
Nequi                   39
Corresponsal            61
Pago de Recibos         60
Pagatodo                62
Recarga de Minutos      59
Reservas                54
Cita previa             62
Transferencia           57
Transferencia.1         63
Recarga Tu Llave        63
Contraentrega           62
Movid Red               62
Dias                     1
Apertura                 0
Cierre                   0
Observacion             54
Unnamed: 40             62
Daviplata.1             50
Domicilios.1            38
Nequi.1                 48
Pago con Tarjeta.1      54
Pago con Efectivo       20
Pagatodo.1              61
Reservas.1              55
Cita previa.1           62
Corresponsal.1          62
Transferencias          60
Recarga Minutos         60
Recarga Tu Llave.1      63
Pago de Recibos.1       61
Contraentrega.1         62
Movi red                62
Horario                 20
Url                      2
Fecha                    0
Unnamed: 59             61
Unnamed: 60              1
dtype: int64
```

```
# Detección de duplicados
duplicados = df_cmr.duplicated().sum()
print(f"Filas duplicadas encontradas: {duplicados}")
```

```
Filas duplicadas encontradas: 0
```

Nota. Código utilizado en la herramienta Jupyter para contar valores únicos por columna y si hay campos vacíos en la hoja “CRM”

Resultados generales

- Cantidad total de registros (filas): 63
- Cantidad total de columnas: 61
- Filas duplicadas: 0
- Campos con valores únicos elevados:
- Nombre Comercial: 61
- Celular: 61
- URL: 60
- Campos con gran cantidad de valores nulos:
- Cedula: 51 nulos
- Teléfono: 47 nulos
- Subcategoria.2: más del 75% vacíos
- Más de 15 columnas tipo Unnamed, no estructuradas

Observaciones específicas. Múltiples campos Subcategorias.1, 2,3,4 corresponden a etiquetas no jerarquizadas. La estructura actual mezcla relaciones muchos-a-muchos sin una separación lógica entre dimensiones (categoría, beneficios, métodos de pago, etc.).

Columnas irrelevantes o corruptas: Al menos 15 columnas no tienen encabezado (Unnamed) o contienen información vacía o inconsistente. Estas serán descartadas o fusionadas durante la fase de transformación.

Valores inconsistentes en medios de pago y servicios: Existen múltiples columnas para el mismo tipo de atributo (Domicilios, Domicilios.1; Daviplata, Daviplata.1), indicando una duplicación innecesaria o una estructura mal interpretada en el ingreso de datos. Ausencia de claves primarias explícitas: No existe un campo id_negocio o id_persona, lo que limita la trazabilidad y

la integridad referencial dentro del modelo relacional futuro. Datos personales sensibles no anonimizados: Se identificaron campos con cédulas, correos electrónicos y teléfonos. Esta situación conlleva a aplicar procesos de Anonimización para cumplir con la Ley 1581 de 2012.

Exploración en Google Analytics / Informe del comportamiento de los usuarios

Figura 14

Consolidado Enero a Mayo 2025 "Informe del Comportamiento de los Usuarios"

```
# Ruta del archivo CSV
data_analytics_usuarios = r"C:\Users\Usuario\Desktop\UNAD\5_Proyecto de Grado\DATA_CRUDA\data_analytics_usuarios_enero_a_mayo_2025.csv"

# Carga del archivo CSV
df_usuarios = pd.read_csv(data_analytics_usuarios)

# Mostrar nombres de las columnas
print("Columnas en el DataFrame:", df_usuarios.columns)

Columnas en el DataFrame: Index(['Mes', 'Categoria de dispositivo', 'Usuarios activos',
       'Usuarios nuevos', 'Sesiones', 'Duracion media de la sesion',
       'Porcentaje de rebote', 'Usuarios recurrentes'],
      dtype='object')

# vista primeras filas
df_usuarios.head(3)
```

	Mes	Categoria de dispositivo	Usuarios activos	Usuarios nuevos	Sesiones	Duracion media de la sesion	Porcentaje de rebote	Usuarios recurrentes	Usuarios activos (%)	Mes_num
0	1	desktop	130	125	147	106.219885	0.088435	18	1.530492	1
1	1	mobile	1688	1601	1981	96.858732	0.037860	299	19.872851	1
2	1	tablet	1	1	1	286.032444	0.000000	0	0.011773	1

Nota. Código utilizado en la herramienta Jupyter para la lectura del "*Informe mensual del comportamiento de los usuarios"* consolidando los primeros 5 meses del año 2025

Figura 15

Estadísticas Descriptivas Generales

```
estadisticas = df_usuarios.describe()
print(estadisticas)

             Mes  Usuarios activos  Usuarios nuevos     Sesiones  \
count  14.000000         14.000000       14.000000    14.000000
mean    2.928571        606.714286      573.857143   719.142857
std     1.491735        737.168119      694.888022   881.926027
min     1.000000          1.000000        1.000000     1.000000
25%     2.000000         34.000000       32.750000    38.250000
50%     3.000000        176.000000      170.500000   200.000000
75%     4.000000       1244.500000     1170.000000  1465.250000
max     5.000000       1781.000000     1682.000000  2154.000000

       Duracion media de la sesion  Porcentaje de rebote  Usuarios recurrentes
count                    14.000000             14.000000             14.000000
mean                    188.814355              0.085064            110.857143
std                     264.161502              0.126493            145.764751
min                      36.907606              0.000000              0.000000
25%                      95.214064              0.030644              4.000000
50%                     104.143971              0.039789             18.000000
75%                     127.551068              0.097675            219.000000
max                    1079.404349              0.500000            366.000000
```

Nota. Código aplicado en la herramienta Jupyter utilizando .describe() para realizar un análisis estadístico descriptivo de las métricas clave del sitio web

Alta dispersión de datos: Las desviaciones estándar son elevadas en casi todas las métricas, lo cual indica una alta variabilidad entre los registros probablemente por diferencias entre meses al tener celebraciones o eventos específicos.

Usuarios activos y nuevos: El promedio mensual de usuarios activos fue de 606, con un máximo de 1.781, lo cual sugiere que en algunos meses hubo campañas exitosas o un aumento orgánico significativo. El número de usuarios nuevos muestra una tendencia similar, lo que indica que una gran parte del tráfico corresponde a usuarios que llegan por primera vez.

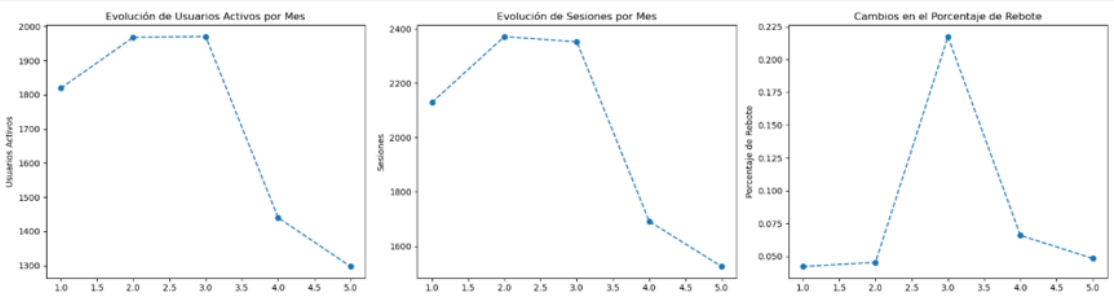
Sesiones: El número de sesiones por mes tiene un promedio de 719, pero el valor máximo llega a 2.154, lo que refuerza la hipótesis de que algunos meses tuvieron un tráfico muy superior al promedio.

Duración media de la sesión: El tiempo promedio de permanencia fue de 188 segundos (≈ 3 min 9 s). Se deben investigar estos valores extremos, ya que podrían deberse a errores de medición, sesiones anómalas o un comportamiento poco común.

Porcentaje de rebote: Aunque el promedio general fue bajo (8,5 %), hubo registros de hasta 50 %, lo que indica que en ciertos casos los usuarios abandonaron el sitio sin interactuar.

Figura 16

Tendencia Enero a Mayo 2025 "Informe del Comportamiento de los Usuarios"



Nota. Código utilizado en la herramienta Jupyter para graficar el comportamiento de los "usuarios activos", "usuarios nuevos" y el "porcentaje de rebote "durante los primeros 5 meses del año 2025

Comportamiento de Usuarios Activos por Mes. El gráfico muestra una tendencia creciente en los primeros tres meses del año, alcanzando el pico en marzo. Sin embargo, abril y mayo reflejan una caída significativa, lo cual puede deberse a factores estacionales, reducción en campañas de promoción o cambios en el posicionamiento SEO.

Comportamiento de Sesiones por Mes. Al igual que los usuarios activos, las sesiones presentan un comportamiento ascendente hasta marzo y una caída pronunciada en los meses siguientes.

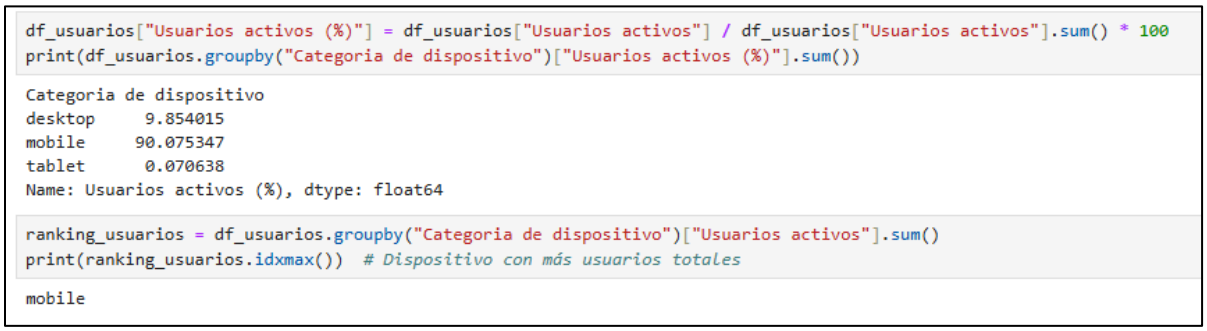
Esto indica una correlación directa entre la cantidad de usuarios y la frecuencia con la que interactúan en el sitio web, lo cual reafirma la importancia de mantener estrategias constantes de contenido y relevancia.

Cambios en el Porcentaje de Rebote. El porcentaje de rebote se mantiene bajo entre enero y marzo, pero se dispara en abril, alcanzando su punto más alto, para luego disminuir en mayo.

Es un indicador clave para investigar qué tipo de contenido, páginas o dispositivos presentan mayor tasa de abandono.

Figura 17

Relevancia de Categoría de Dispositivo



```
df_usuarios["Usuarios activos (%)"] = df_usuarios["Usuarios activos"] / df_usuarios["Usuarios activos"].sum() * 100
print(df_usuarios.groupby("Categoria de dispositivo")["Usuarios activos (%)"].sum())
```

```
Categoria de dispositivo
desktop     9.854015
mobile     90.075347
tablet      0.070638
Name: Usuarios activos (%), dtype: float64
```

```
ranking_usuarios = df_usuarios.groupby("Categoria de dispositivo")["Usuarios activos"].sum()
print(ranking_usuarios.idxmax())  # Dispositivo con más usuarios totales
```

```
mobile
```

Nota. Código utilizado en la herramienta Jupyter para identificar el comportamiento de la categoria de dispositivo y detectar cual tiene más usuarios, sesiones y menor porcentaje de rebote

La categoría de mobile. Representa más del 90% del tráfico total, lo que indica que los usuarios acceden predominantemente desde celulares.

Esto nos permite entender que la estrategia digital debe priorizar la optimización móvil.

Además de ser el más usado, es el que presenta mayor retención, lo que refuerza la importancia de diseñar una experiencia fluida y atractiva para este tipo de dispositivo.

Exploración en Google Analytics / Informe: KPIs por Mes “páginas”

Figura 18

Consolidado Enero a Mayo 2025 de KPIs “Páginas”

```
# Ruta del archivo CSV
data_analytics_usuarios = r"C:\Users\Usuario\Desktop\UNAD\5_Proyecto de Grado\DATA_CRUDA\data_analytics_paginas_enero_a_mayo_2025.csv"

# Carga del archivo CSV
df_paginas = pd.read_csv(data_analytics_usuarios)
```

```
# Mostrar nombres de las columnas
print("Columnas en el DataFrame:", df_paginas.columns)
```

```
Columnas en el DataFrame: Index(['Mes', 'Página de destino', 'Usuarios nuevos', 'Usuarios activos',
       'Usuarios recurrentes', 'Sesiones', 'Duración media de la sesión',
       'Sesiones por usuario activo', 'Porcentaje de rebote'],
      dtype='object')
```

```
# vista primeras filas
df_paginas.head(3)
```

	Mes	Página de destino	Usuarios nuevos	Usuarios activos	Usuarios recurrentes	Sesiones	Duración media de la sesión	Sesiones por usuario activo	Porcentaje de rebote
0	2	/lista/ruta-de-transmilenio-b75-portal-norte	489	532	106	634	89.628292	1.191729	0.023659
1	2	/lista/ruta-de-transmilenio-h75-portal-usme	476	507	95	590	94.779360	1.163708	0.008475
2	3	/lista/ruta-de-transmilenio-b75-portal-norte	475	507	99	594	75.380710	1.171598	0.010101

Nota. Código utilizado en la herramienta Jupyter para la lectura del informe de KPIs “páginas” consolidando los primeros 5 meses del año 2025

Figura 19

Estadísticas Descriptivas Generales (Consolidado Enero a Mayo 2025)

```
estadisticas = df_paginas.describe()
print(estadisticas)
```

```
              Mes  Usuarios nuevos  Usuarios activos  Usuarios recurrentes  \
count  649.000000       649.000000        649.000000            649.000000
mean     3.009245        12.379045         13.593220              2.513097
std      1.395509        46.254342         49.629998              9.844951
min      1.000000         0.000000          1.000000              0.000000
25%      2.000000         1.000000          1.000000              0.000000
50%      3.000000         2.000000          3.000000              0.000000
75%      4.000000         6.000000          7.000000              1.000000
max      5.000000       489.000000        532.000000            106.000000

          Sesiones  Duración media de la sesión  Sesiones por usuario activo  \
count  649.000000                   649.000000                   649.000000
mean    15.513097                   141.852983                     1.091003
std     57.770674                   362.988641                     0.221948
min      1.000000                     0.000000                     1.000000
25%      1.000000                    19.617945                     1.000000
50%      3.000000                    70.956499                     1.000000
75%      9.000000                   153.146825                     1.096774
max    634.000000                  7216.430723                     4.000000

       Porcentaje de rebote
count            649.000000
mean               0.030773
std                0.128914
min                0.000000
25%                0.000000
50%                0.000000
75%                0.000000
max                1.000000
```

Nota. Código aplicado en la herramienta Jupyter utilizando .describe() para realizar un análisis estadístico descriptivo de las métricas relacionadas al comportamiento de las páginas del sitio web

Se observa un elevado promedio de usuarios activos por página (13,6) y de usuarios nuevos (12,4), lo que indica una buena capacidad del sitio para atraer tráfico inicial. Sin embargo, este flujo se contrasta con un bajo promedio de usuarios recurrentes (apenas 2,5) y un promedio de sesiones por usuario activo cercano a 1,09 (con la mayoría en 1,0), sugiriendo una limitada fidelización y poco retorno al mismo contenido. En cuanto a la duración de las sesiones, la media general es de 141 segundos, pero la alta desviación estándar y un máximo de 7.216 segundos muestran una gran variabilidad, con algunas páginas logrando una alta retención

mientras que muchas otras experimentan visitas muy breves. En conclusión, el DirectorioUsme.com atrae eficazmente, pero enfrenta el desafío de mejorar la retención de usuarios y la consistencia en la calidad de las interacciones a través de sus diversas páginas, lo cual requerirá un enfoque basado en métricas específicas por contenido.

Figura 20

Ranking de Páginas del Sitio DirectorioUsme.com (Enero a Mayo 2025)

```
# Páginas con más usuarios activos acumulados
print(df_paginas.groupby("Página de destino")["Usuarios activos"].sum().sort_values(ascending=False).head(5))
```

```
Página de destino
/lista/ruta-de-transmilenio-b75-portal-norte    1780
/lista/ruta-de-transmilenio-h75-portal-usme     1583
/lista/ruta-de-transmilenio-h72-portal-usme     1088
/lista/banco-caja-social-usme-santa-librada      369
(not set)                                        221
Name: Usuarios activos, dtype: int64
```

```
#  Páginas con mayor duración media de sesión
print(df_paginas.groupby("Página de destino")["Duración media de la sesión"].mean().sort_values(ascending=False).head(5))
```

```
Página de destino
/lista/239-desayunos-sorpresa-kaprichitos                    7216.430723
/lista/supermercado-san-vicente-el-cortijo                   1916.733891
/como-reclamar-queja-por-servicios-mas-caros-de-lo-normal     1504.424977
/impresiona-aqui-encuentras                                   581.660060
/lista-categoria/servicios-de-oficias-de-bancos-en-usme-381   485.302290
Name: Duración media de la sesión, dtype: float64
```

```
#Páginas con mayor porcentaje de rebote
print(df_paginas.groupby("Página de destino")["Porcentaje de rebote"].mean().sort_values(ascending=False).head(5))
```

```
Página de destino
/lista/perfil-de-ejemplo-directorio-usme                              1.000000
(not set)                                                             0.983233
/blog-de-usme                                                         0.833333
/lista/ruta-de-transmilenio-j76-universidades                         0.333333
/lista/miscelanea-y-papeleria-tienda-de-detalles-tutilandia-el-virrey 0.333333
Name: Porcentaje de rebote, dtype: float64
```

```
# Páginas con más usuarios recurrentes
print(df_paginas.groupby("Página de destino")["Usuarios recurrentes"].sum().sort_values(ascending=False).head(5))
```

```
Página de destino
/lista/ruta-de-transmilenio-b75-portal-norte    346
/lista/ruta-de-transmilenio-h75-portal-usme     290
(not set)                                       215
/lista/ruta-de-transmilenio-h72-portal-usme     164
/lista/banco-caja-social-usme-santa-librada      64
Name: Usuarios recurrentes, dtype: int64
```

Nota. Código aplicado en la herramienta Jupyter para examinar los rankings de comportamiento de las páginas del sitio DirectorioUsme.com

Páginas con mayor cantidad de usuarios activos: Se observa un claro dominio de las páginas relacionadas con Transmilenio (rutas B75, H75, H72) en términos de usuarios activos, lo que indica que el contenido informativo sobre transporte público es el de mayor tráfico y valor para la audiencia.

Páginas con mayor duración media de sesión: Las páginas de "Desayunos sorpresa" y "Supermercado San Vicente" destacan por generar una duración media de sesión significativamente alta.

Las páginas con alta duración de sesión demuestran un fuerte interés o lectura profunda por parte del usuario.

Páginas con mayor porcentaje de rebote: Genera una alarma que el 100% de rebote en la "página de perfil de ejemplo" y en algunas rutas específicas.

Asimismo, la sección "/blog-de-usme" presenta un elevado 83% de rebote, lo que indica un problema significativo en la atracción o retención de usuarios en estas áreas.

Páginas con Mayor Cantidad de Usuarios Recurrentes**:** Las páginas que logran mayor recurrencia de usuarios coinciden con las de mayor actividad: las rutas de Transmilenio y las secciones de bancos. Esto sugiere una oportunidad para desarrollar estrategias de actualización constante de esta información

Exploración de Datos – Google Search Console (enero a mayo 2025)

Figura 21

Archivo Consolidado de Enero a Mayo de 2025 del Tráfico Orgánico

```
# Cargar el archivo de Excel
ruta_archivo = "C:/Users/Usuario/Desktop/UNAD/5_Proyecto de Grado/DATA_CRUDA/data_console_enero_a_mayo_2025.xlsx"
archivo = pd.ExcelFile(ruta_archivo)

# Mostrar nombres de hojas disponibles
print("Hojas en el archivo:", archivo.sheet_names)
```

```
Hojas en el archivo: ['Consultas', 'Páginas', 'Países', 'Dispositivos', 'Aparición en búsquedas', 'Fechas', 'Filtros']
```

Nota. Código aplicado en la herramienta Jupyter para examinar las hojas que contiene el archivo descargado de Google Search Console con los datos orgánicos del sitio DirectorioUsme.com

Figura 22

Estadísticas Descriptivas del Tráfico Orgánico (Enero a Mayo 2025)

```
df_consultas = archivo.parse("Consultas")

# Estadísticas descriptivas
print(df_consultas.describe())
```

	Clics	Impresiones	CTR	Posición
count	1000.00000	1000.000000	1000.000000	1000.000000
mean	5.62600	544.667000	0.020606	7.823530
std	24.03951	1630.549816	0.069683	4.207231
min	0.00000	1.000000	0.000000	1.000000
25%	0.00000	96.750000	0.000000	5.427500
50%	1.00000	159.000000	0.006100	7.795000
75%	3.00000	325.250000	0.020725	9.670000
max	385.00000	19519.000000	1.000000	67.150000

Nota. Código aplicado en la herramienta Jupyter para examinar la hoja “Consultas” del archivo consolidado descargado de Google Search Console

Se analizaron 1.000 términos de búsqueda (consultas) que dirigieron tráfico orgánico al sitio DirectorioUsme.com. Los principales hallazgos estadísticos son:

Consultas con alto volumen, pero baja conversión: Aunque el promedio de impresiones es alto (544), el promedio de clics es bajo (5.6), y la mediana es apenas 1. Esto indica que muchas búsquedas muestran el sitio, pero no generan interacción. Esto se ve reforzado por un CTR promedio de solo 2.06%.

Distribución desigual del tráfico: La desviación estándar en clics e impresiones es muy alta, lo que significa que unas pocas consultas concentran la mayoría del tráfico, mientras que muchas tienen poca o nula interacción.

Posicionamiento orgánico relativamente bueno: La posición promedio es 7.8, lo que ubica al sitio en la primera página de resultados de Google en muchos términos. No obstante, hay valores extremos de hasta posición 67, que deben ser analizados para evaluar si son páginas poco optimizadas o irrelevantes.

Clics = 0: En varias consultas el 25% inferior (Q1) tiene 0 clics y 0% de CTR. Estas son oportunidades de mejora SEO, ya que el sitio está apareciendo, pero no es lo suficientemente atractivo para obtener clics (tal vez por el título, meta descripción o intención de búsqueda no alineada).

Figura 23

Palabras Clave con más Clics (Consolidado Enero a Mayo 2025)

```
# palabras clave con más clics
print(df_consultas.sort_values(by="Clics", ascending=False).head(10))
```

	Consultas principales	Clics	Impresiones	CTR	Posición
0	b75 paradas	385	18494	0.0208	5.84
1	paradas b75	326	13402	0.0243	5.56
2	paradas h75	293	12563	0.0233	5.33
3	h75 paradas	283	19519	0.0145	5.56
4	h72 paradas	177	13035	0.0136	4.55
5	paradas h72	172	9796	0.0176	4.90
6	paradas del b75	139	7384	0.0188	5.79
7	ruta b75	117	4965	0.0236	5.12
8	ruta h75	98	4740	0.0207	4.81
9	paradas del h75	90	4956	0.0182	5.58

Nota. Código aplicado en la herramienta Jupyter para examinar la hoja “Páginas” del archivo consolidado descargado de Google Search Console para encontrar el ranking del top 10 de palabras que más tienen clics en los resultados de búsquedas orgánicas

Las consultas más relevantes están centradas en rutas de Transmilenio (B75, H75, H72). Además, están en primeras posiciones (top 5) en Google, pero tienen CTR bajo (< 3%), lo cual indica que los títulos o descripciones no están llamando la atención.

Por lo que se debería realizar una revisión en los títulos y metadescripciones para aumentar el CTR en búsquedas donde ya se tiene buen posicionamiento.

Figura 24

Páginas con más Clics (Consolidado Enero a Mayo 2025)

```
# Páginas con más clics
print(df_paginas.sort_values(by="Clics", ascending=False).head(10))
```

	Páginas principales	Clics	Impresiones	CTR	Posición
0	https://www.directoriousme.com/lista/ruta-de-t...	2024	100669	0.0201	6.62
1	https://www.directoriousme.com/lista/ruta-de-t...	1746	101491	0.0172	6.25
2	https://www.directoriousme.com/lista/ruta-de-t...	1149	80554	0.0143	5.83
3	https://www.directoriousme.com/lista/banco-caj...	398	5638	0.0706	13.61
4	https://www.directoriousme.com/lista/ruta-de-t...	208	47264	0.0044	9.38
5	https://www.directoriousme.com/lista/ruta-de-t...	173	62116	0.0028	8.95
6	https://www.directoriousme.com/lista/ruta-de-t...	151	40742	0.0037	8.74
7	https://www.directoriousme.com/lista/cai-en-el...	132	2683	0.0492	7.34
8	https://www.directoriousme.com/lista/cai-barri...	131	2165	0.0605	6.75
9	https://www.directoriousme.com/lista/cai-en-el...	104	3877	0.0268	8.14

Nota. Código aplicado en la herramienta Jupyter para examinar la hoja “Páginas” del archivo consolidado descargado de Google Search Console para encontrar el ranking del top 10 de páginas que más tienen clics en los resultados de búsquedas orgánicas

Estas páginas lideran el tráfico y deben ser priorizadas en mantenimiento y actualización, se deben revisar para entender su optimo resultado y usarlas como modelo para el diseño de otras páginas similares.

Figura 25

Páginas con Mejor CTR (Consolidado Enero a Mayo 2025)

```
# Páginas con mejor CTR
print(df_paginas.sort_values(by="CTR", ascending=False).head(10))
```

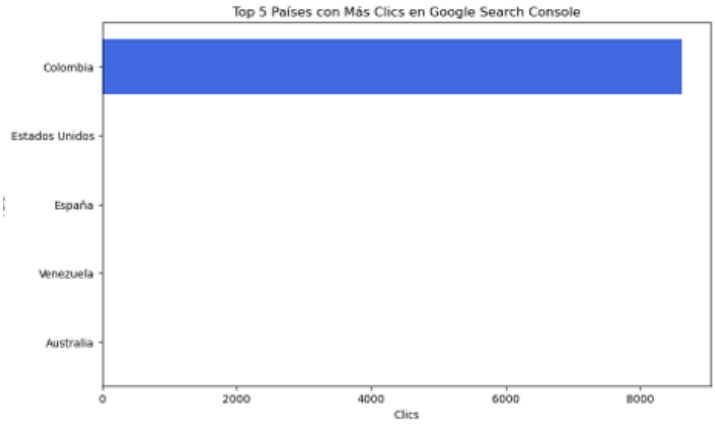
```
                              Páginas principales  Clics  Impresiones  \
176  https://www.directoriousme.com/lista-categoria...      1            1
175  https://www.directoriousme.com/lista-categoria...      1            4
174  https://www.directoriousme.com/servicio-o-come...      1            4
115          https://www.directoriousme.com/panalera/      4           23
116  https://www.directoriousme.com/lista/421-maest...      4           23
173  https://www.directoriousme.com/abarrotes/pollo...      1            6
114  https://www.directoriousme.com/lista-categoria...      4           27
172  https://www.directoriousme.com/lista-categoria...      1            8
149  https://www.directoriousme.com/lista-categoria...      2           18
150  https://www.directoriousme.com/servicio-o-come...      2           18

        CTR  Posición
176  1.0000     28.00
175  0.2500     38.00
174  0.2500      6.75
115  0.1739      5.26
116  0.1739      8.48
173  0.1667      9.33
114  0.1481     15.59
172  0.1250     10.00
149  0.1111     11.94
150  0.1111     12.28
```

Nota. Código aplicado en la herramienta Jupyter para examinar la hoja “Páginas” del archivo consolidado descargado de Google Search Console para encontrar el ranking del top 10 de páginas con mejor CTR en los resultados de búsquedas orgánicas

Aquí se identifican páginas con CTR superiores al 10%, incluso del 100%, aunque con pocas impresiones: estas páginas son ejemplos de alto rendimiento de conversión. Aunque tienen poca visibilidad, cada vez que aparecen generan clics. Se debe potenciar el posicionamiento de esas URLs (tienen buena estructura, pero poca visibilidad).

Figura 26

Top 5 Países con más Clics (Consolidado Enero a Mayo 2025)



Nota. Código aplicado en la herramienta Jupyter para examinar la hoja "Países" del archivo consolidado descargado de Google Search Console para visualizar los 5 países que tienen más relevancia para el sitio web

El sitio web al ser específico para una localidad de Bogotá no se espera que otros países tengan tráfico relevante, es coherente que el foco este concentrado en Colombia.

Figura 27

Clics por Categoria por Dispositivo (Consolidado Enero a Mayo 2025)

```
df_dispositivos = archivo.parse("Dispositivos")

# Sumar clics por tipo de dispositivo
print(df_dispositivos.groupby("Dispositivo")["Clics"].sum())
```

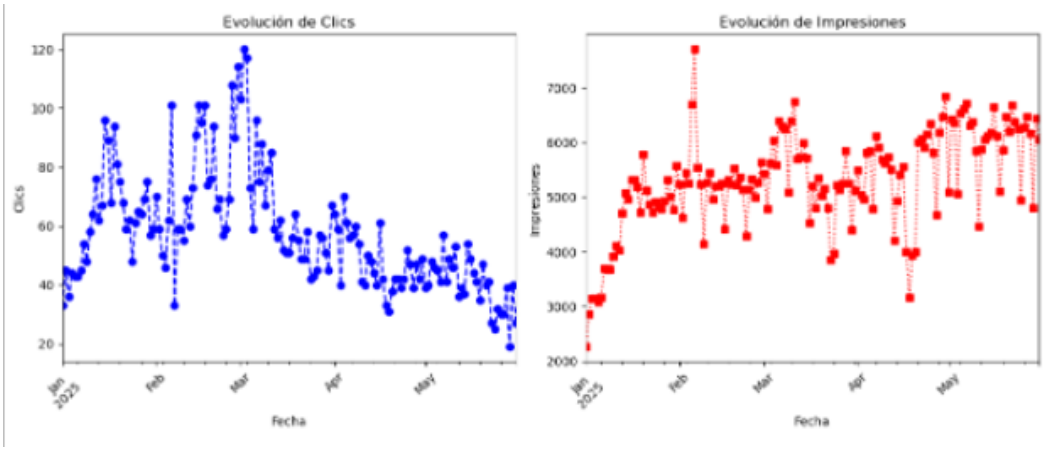
```
Dispositivo
Móviles      8133
Ordenador     524
Tablet          8
Name: Clics, dtype: int64
```

Nota. Código aplicado en la herramienta Jupyter para examinar la hoja "Dispositivos" del archivo consolidado descargado de Google Search Console para identificar la relevancia en clics por tipo de dispositivo

El sitio es altamente móvil (> 90% del tráfico), lo que se evidencio en análisis anteriores con la herramienta de Google Analytics y refuerza la necesidad de mantener un diseño responsivo óptimo, monitorear la velocidad en mobile y adaptar los textos.

Figura 28

Comportamiento de Clics e Impresiones (Consolidado Enero a Mayo 2025)



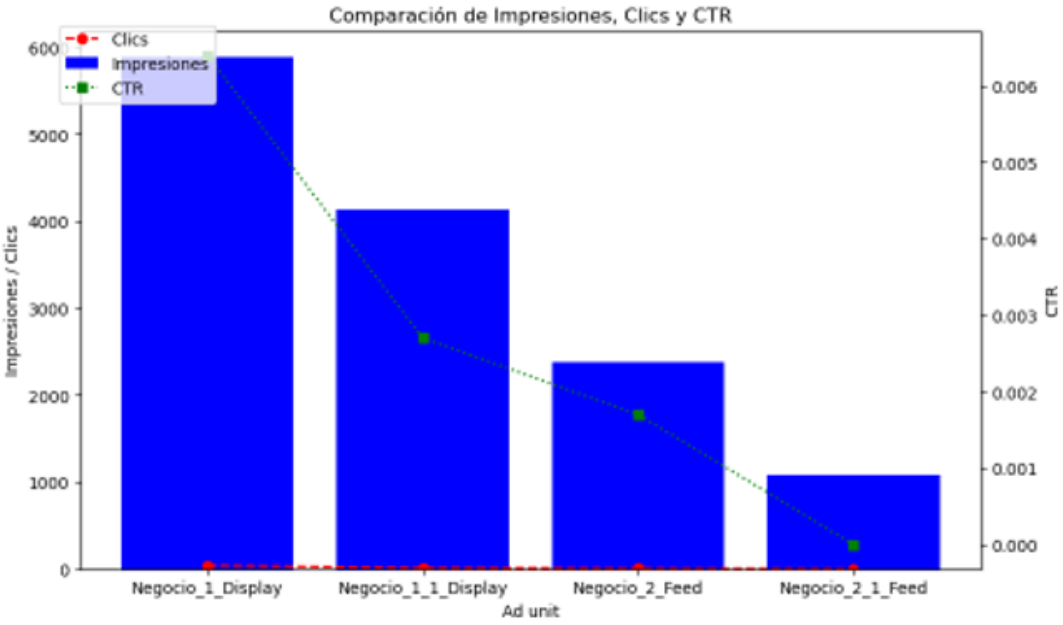
Nota. Código aplicado en la herramienta Jupyter para examinar la hoja “Fecha” del archivo consolidado descargado de Google Search Console para visualizar su evolución en clics e impresiones en lo transcurrido de los primeros 5 meses del año 2025

Clics e impresiones: En el gráfico de clics se observa una tendencia creciente entre enero y marzo, con un pico cercano a los 120 clics diarios. A partir de abril comienza una caída sostenida, llegando a menos de 30 clics diarios en mayo.

Las impresiones siguen una curva similar, aunque se mantienen más estables, con picos sobre las 7.000 impresiones. Esto indica que el sitio sigue siendo visible en los resultados, pero se está haciendo menos clic en las páginas.

Figura 29

Rendimiento de los Anuncios en Google Adsense



Nota. Código aplicado en la herramienta Jupyter para examinar el reporte descargado de Google Adsense donde se tiene información de enero hasta mayo de 2025 de impresiones, clics y CTR de los espacios de anuncio ubicados en el home del sitio web

El gráfico presenta el rendimiento de los formatos de anuncios ubicados en el home del sitio web por Adsense desde el 1 de enero hasta el 31 de mayo de 2025. Se visualizan las siguientes métricas:

Hay baja efectividad en todos los formatos analizados, con una relación impresiones/clics muy baja, lo que afecta directamente el rendimiento de los anuncios. El formato Display ubicado en “Negocio_1” genera más visibilidad, pero no logra captar el interés para obtener clics. El CTR más bajo en los formatos Feed puede indicar que estos están mal ubicados, ignorados visualmente o cargando de forma poco atractiva en dispositivos móviles.

Para este proyecto no se mostrarán datos relacionados a RPM o ingresos de los anuncios ya que se quiere mantener esta información reservada

Verificación de Calidad

Durante la exploración de los datos, se detectaron diversas problemáticas de calidad que afectan la integridad, coherencia y utilidad de la información:

Valores nulos. Columnas clave como *"Cedula"*, *"Subcategoría.1"* y *"Categoría L1"* presentan altos porcentajes de valores faltantes, lo que limita su uso inmediato sin transformación.

Formatos inconsistentes. En los campos de tiempo (*Duración media de sesión*) se identificaron formatos mixtos (minutos/segundos y segundos totales), lo que requiere estandarización previa al análisis.

Datos duplicados. Aunque la mayoría de las hojas no presentaron duplicados, se detectó riesgo de valores redundantes en etiquetas o subcategorías que requieren validación.

Columnas sin uso o irrelevantes. En varios archivos y principalmente en el archivo CRM, se identificaron múltiples columnas *"Unnamed"* con valores vacíos o sin relevancia directa para el análisis, las cuales serán descartadas en la etapa de preparación.

Estas observaciones serán abordadas durante la Fase 3 con procesos de limpieza, transformación y validación, garantizando que los datos sean consistentes, completos y confiables para su posterior análisis e integración en el sistema relacional.

Fase 3 Preparación de los Datos

El desarrollo de la fase de preparación de los datos representa un pilar fundamental dentro del ciclo analítico, ya que permite transformar la información recolectada a partir del inventario de fuentes, las fichas técnicas y los hallazgos del análisis exploratorio, en un insumo limpio y estructurado.

Estandarización de Nomenclatura

El primer paso es establecer una convención clara y coherente para la nomenclatura de las tablas y actuales y dejen base para las futuras tablas que se necesiten crear. Esta estandarización busca facilitar la lectura y jerarquía entre entidades. Las principales reglas aplicadas son:

Claridad y legibilidad. Se emplean guiones bajos (_) para separar las palabras dentro de un identificador, evitando confusiones semánticas.

Compatibilidad. Se evita el uso de mayúsculas, guiones medios (-), tildes, espacios y caracteres especiales, los cuales podrían causar errores en entornos SQL o dificultar la escritura de consultas.

Esta nomenclatura será adoptada de manera uniforme en todas las tablas, tanto en las existentes como en las que se creen a futuro, garantizando consistencia estructural en todo el sistema.

A continuación, en la Tabla 20 se presentan algunos ejemplos de buenas y malas prácticas en la asignación de nombres a tablas y variables.

Tabla 20

Algunos Ejemplos Correctos de la Nomenclatura de Tablas

Regla	Ejemplo correcto	Ejemplo para evitar
Primera letra en mayúscula	Belleza_Bienestar	BELLEZA_y_Bienestar
Palabras cortas en mayúscula que sean descriptivas	CTR	clicksthroughrateporcentaje
Sin tildes, ñ ni caracteres raros	Cafeteria	Cafetería
No iniciar con número	Academias_L1	2L_Academias
Prefijos coherentes	Data_GA_Usuarios	kpiAnalyticsData

Nota. Esta tabla da ejemplos de usos correctos y de los usos a evitar al nombrar una tabla, para así mantener una coherencia en la nomenclatura y evitar malas prácticas al ejecutar consultas o procesos en lenguaje SQL

Nomenclaturas de Tablas. Con el fin de mantener el orden, la trazabilidad y la consistencia entre las distintas fuentes internas y externas, la siguiente nomenclatura sigue una estructura lógica basada en el origen y el contenido de los datos.

Nomenclaturas de tablas

<prefijo>_<fuente>_<tema >

Se define una clasificación de prefijos que permite identificar el propósito general de cada tabla dentro del sistema. Estos prefijos se anteponen al nombre completo de la tabla, facilitando su trazabilidad y comprensión tanto para desarrolladores como para analistas.

En la siguiente Tabla 21 presenta los prefijos establecidos y su respectiva descripción.

Tabla 21

Clasificación de Tablas para Nombrar el <Prefijo>

Prefijo	Nombre	Descripción
Cat	Categoría	Utilizado para tablas que almacenan jerarquías de categorías y subcategorías de los diferentes negocios o servicios registrados.
Etq	Etiquetas	Aplicado a tablas que contienen información complementaria sobre beneficios, medios de pago, servicios adicionales u otras características.
Zn	Zona	Tabla estable. No está sujeta a cambio frecuentes.
Reg	Registros	Prefijo utilizado en tablas que almacenan datos provenientes de personas, negocios, empleados o proveedores.
Prefijo	Nombre	Descripción
Data	Métricas y KPIs	Correspondiente a tablas que contienen información recolectada desde herramientas de analítica web o publicidad digital.
Time	Tiempo	Utilizado para tablas de referencia que almacenan información sobre horarios y días de atención al público.
Inter	Intermedia	Prefijo asignado a tablas intermedias, generalmente utilizadas para representar relaciones muchos a muchos entre entidades.

*Nota***.** La tabla presenta los prefijos a utilizar en la nomenclatura de las tablas de la base de datos del proyecto. Estos prefijos permiten clasificar las tablas según su función y contenido, facilitando la organización, comprensión y mantenimiento del modelo relacional

Además del prefijo, cada tabla incorpora un descriptor que indica la fuente de origen de los datos. Esta segunda parte del nombre facilita la identificación del sistema o documento del cual proviene la información, ya sea interno (como el CRM) o externo (como herramientas de Google). La Tabla 22 resume las principales fuentes utilizadas y la etiqueta asignada a cada una.

Tabla 22

Clasificación para Identificar los Orígenes de los Datos <Fuente>

Fuente	Descripción
Categoría	Datos estructurados que definen las categorías (nivel 1) y subcategorías (nivel 2) utilizadas para clasificar los negocios o servicios registrados.
Métodos	Registro de los métodos de pago que pueden ofrecer los negocios, como Nequi, Daviplata, efectivo, entre otros.
Beneficios	Lista de atributos que representan ventajas para los usuarios, tales como reservas, domicilios.
Servicios	Información complementaria sobre servicios especiales ofrecidos por los negocios, como corresponsal bancario o recargas móviles.
Ubicación	Listado fijo de barrios y sectores de la localidad de Usme, útil para procesos de georreferenciación y segmentación territorial.
Datos	Información recolectada con datos personales, comerciales, empleados y de proveedores.
GA	Datos extraídos desde Google Analytics, relacionados con el tráfico del sitio, comportamiento de usuarios, duración de sesiones, entre otros.
GSC	Información proveniente de Google Search Console, enfocada en el rendimiento de búsqueda orgánica, consultas, páginas más visitadas y clics.
GADSE	Datos extraídos de Google AdSense, vinculados con la monetización del sitio: ingresos, impresiones, clics en anuncios y RPM.

Nota. La tabla presenta las principales fuentes de datos utilizadas en el proyecto. Estas fuentes abarcan datos estructurados, operativos y analíticos

Clasificación de tablas por <tema>: Además de identificar la fuente de origen de los datos, es necesario clasificar las tablas según el tema específico al que hacen referencia. Esto permite mejor organización lógica dentro del modelo relacional, ya que cada fuente puede descomponerse en múltiples componentes temáticos. Por ejemplo, si la fuente de datos es denominada "Datos", esta puede subdividirse en temas como datos personales, comerciales, proveedores o empleados, dependiendo según la información que contenga cada tabla. Esta categorización facilita la comprensión del sistema, la trazabilidad de la información y la estructuración eficiente de las relaciones entre entidades.

Importante: No se podrá utilizar dentro de las nomenclaturas de las tablas y campos palabras como OLD, NEW, V1, V2, Vn, MESH, TEMP, TMP, BCK, PRUEBA, TEST, etc. No se podrán utilizar nombres propios en las tablas, o “dueños” de los datos

Transformación y normalización. Este proceso de transformación y normalización garantizará que la información extraída del CRM esté debidamente organizada, estandarizada y alineada con las buenas prácticas de modelado relacional, facilitando su posterior análisis, integración y mantenimiento dentro del sistema.

Transformación y Normalización del Archivo CRM / hoja “categorías”

Categorías L1: Como parte del proceso de transformación de los datos provenientes del archivo CRM, se identificó que la hoja correspondiente a las “categorías” incluía registros sin estructura uniforme, donde algunos campos combinaban identificadores y nombres en una misma celda y le hacía falta información complementaria como la asignación de color por categoria, fecha de creación y de actualización.

Como se crea una nueva tabla se asigna el nombre con la nomenclatura correspondiente: “Cat_Categoria_L1”

Tabla 23

Estructura de la Nueva Tabla para las Categorías - "Cat_Categoria_L1"

Atributo	Tipo de dato	Descripción funcional
Id_Categoria_L1	INT – NO NULL - (PK)	Identificador único de la categoría principal
Nombre_Categoria_L1	VARCHAR (100)	Nombre visible de la categoría nivel 1
Id_URL_Paginas_FK	IN (FK)	Clave foránea que se relaciona con la tabla dim_pagina, para asociar una URL pública
Color_Nombre	VARCHAR (50)	Nombre del color asignado a esta categoría (opcional)
Atributo	Tipo de dato	Descripción funcional
Color_Hexadecimal	VARCHAR (7)	Código hexadecimal del color asociado para línea gráfica
Estado	TINYINT	Estado del registro: 1 = Activo, 0 = Inactivo
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro
Url_Categoria	VARCHAR (150)	Identificador URL amigable para SEO

Nota. Esta tabla es la guía para comprender los atributos que va a tener la tabla *"Cat_Categoria_L1"* y el tipo de dato correspondiente que será asociado en el sistema de gestión de base de datos

Como parte de la transformación de la tabla de categorías, se aplicó un proceso de normalización que incluyó la estandarización de nombres bajo el campo Nombre_Categoria_L1,

donde se capitalizó la primera letra de cada palabra, se reemplazaron espacios por guiones bajos y se eliminaron tildes o caracteres especiales.

Adicionalmente, se asignaron colores distintivos a cada categoría mediante los campos Color_Nombre y Color_Hex, con el fin de facilitar su visualización en reportes analíticos.

Por último, se incorporó la columna Url_Categoria_L1, para almacenar las URLs que se asigna a cada categoria y que en procesos más adelanté servirán como referencia para realizar cruces de información o de métricas con fuentes externas

Categorias L2: La tabla de subcategorías, originalmente ubicada en la hoja "categorías" del archivo CRM, fue aislada y normalizada como una tabla independiente, nombrada siguiendo la nomenclatura establecida *"*Cat_Subcategoria_L2*"*.

Tabla 24

Estructura de la Nueva Tabla para las Subcategorías - "Cat_Subcategoria_L2"

Atributo	Tipo de dato	Descripción funcional
Id_Subcategoria_L2	INT – NO NULL - (PK)	Identificador único de la subcategoría.
Nombre_Subcategoria_L2	VARCHAR (100)	Nombre estandarizado de la subcategoría.
Id_URL_Paginas_FK	IN (FK)	Clave foránea que se relaciona con la tabla dim_pagina, para asociar una URL pública
Id_Categoria_L1_FK	INT (FK)	Referencia al ID de la categoría principal.
Estado	TINYINT	1 = activo, 0 = inactivo

Atributo	Tipo de dato	Descripción funcional
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro
Url_Subcategoria_L2	VARCHAR (180)	Identificador URL amigable para SEO

Nota. Esta tabla es la guía para comprender los atributos que va a tener la tabla *"Cat_Subcategoria_L2"* y el tipo de dato correspondiente que será asociado en el sistema de gestión de base de datos y la llave foránea que va a tener para relacionarse con la tabla *"Cat_Categoria_L1"*

En el proceso de transformación de la tabla Cat_Subcategoria_L2, se separaron valores compuestos que contenían tanto el nombre como el ID en una misma celda, permitiendo extraer de forma estructurada el identificador correspondiente. Posteriormente, se incorporó la columna Id_Categoria_L1_FK para establecer la relación jerárquica con la categoría de primer nivel. Finalmente, se añadió el campo Url_Subcategoria_L2, diseñado para facilitar el cruce con otras tablas y optimizar su uso en estrategias de posicionamiento SEO y análisis de métricas de tráfico web.

Etiquetas: La información de etiquetas, originalmente ubicada en la hoja "categorías" del archivo CRM, fue aislada, dividida en tres tablas independientes y normalizadas. La división se realizó teniendo en cuenta su fuente (Métodos, Beneficios y Servicios). Aplicando la nomenclatura establecida las tablas quedan con los siguientes nombres:

- "Etqt_Metodos_Pago"
- "Etq_Beneficios_Comerciales"

- “Etq_Servicios_Extra”

Tabla 25

Estructura para las Tablas Relacionadas con el Prefijo “Etq” – Etiquetas

Atributo	Tipo de dato	Descripción funcional
Id_Etiqueta_PK	INT – NO NULL - (PK, AI)	Identificador único de la etiqueta (PK)
Nombre_Etiqueta	VARCHAR (100)	Ejemplo: Pago_Efectivo, Nequi
Estado	TINYINT	1 = Activa, 0 = Inactiva
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro

Nota. Esta tabla es la guía para comprender los atributos, el tipo de dato que van a tener las tablas que se vayan generando que estén relacionadas con el prefijo “Etq”, las fuentes relacionadas son Métodos de pago, Beneficios comerciales, Servicios extra.

Para las tablas con prefijo ETQ se definió su ID como clave primaria. El estado se definió como Activas (valor 1), Inactivas (valor 0).

Las fechas de creación se dejaron NULL por falta de historial, y la Fecha_Actualizacion se fijó en 2025-06-01 (fecha de procesamiento).

Transformación Normalización del Archivo CRM / hoja “barrio”

Como parte del proceso de transformación de los datos provenientes del archivo CRM, se identificó que la hoja correspondiente a “barrio” hay tres entidades que deberían estar separadas

ay que no tienen una relación que justifiqué unirlas en una sola entidad, adicional ninguna tiene asignado un identificador único tampoco su estado (activo / inactivo).

Para iniciar la transformación cada tabla será aislada, serán tres tablas independientes y normalizadas.

Aplicando la nomenclatura establecida a las tres tablas que se generan quedan con los siguientes nombres:

- "Zn_Ubicacion_Barrio"
- "Time_Atencion_Dia"
- "Time_Atencion_Inicio"

Barrio: La tabla con la información de los diferentes barrios no se provee que tenga demasiados ajustes y cambios ya que esta variable son de barrios que no tienen una probabilidad alta de que desparezcan o se creen nuevos barrios. La tabla se fija con los atributos de ID único como clave primaria (PK), estado activo (valor 1), inactivo (valor 0) y se agregan dos campos para tener trazabilidad de la fecha de creación o cargue en el sistema y un campo de fecha de actualización para tener el registro de los ajustes que se realicen,

Tabla 26

Estructura para la Tabla de Barrios "Zn_Ubicacion_Barrio"

Atributo	Tipo de dato	Descripción funcional
Id_Barrio_PK	INT (PK, AI)	Identificador único del barrio
Nombre_Barrio	VARCHAR (100)	Nombre oficial del barrio
Atributo	Tipo de dato	Descripción funcional
Estado	TINYINT	1 = Activo, 0 = Inactivo

Atributo	Tipo de dato	Descripción funcional
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro

Nota. Esta tabla es la guía para comprender los atributos, el tipo de dato, estado; fecha de creación o modificación que va a tener la tabla con la información de los barrios.

Días de atención: Para transformar, estandarizar y normalizar la tabla referente a los días de atención *"Zn_Atencion_Dia"* se realizaron los siguientes pasos clave:

Se le asigno el atributo de ID único como clave primaria (PK), a la combinación de días que se generen, adicional el atributo de estado activo (valor 1), inactivo (valor 0). Se agregan dos campos para tener trazabilidad de la fecha de creación o cargue en el sistema y un campo de fecha de actualización para tener el registro de los ajustes que se realicen.

Tabla 27

Estructura para la Tabla con los Días de Atención "Time_Atencion_Dia"

Atributo	Tipo de dato	Descripción funcional
Id_Atencion_Dia_PK	INT (PK, AI)	Identificador único del patrón de atención
Dias_Rango	VARCHAR (100)	Rango de días en los que se atiende (ej. "lunes a sábado", "Domingo a Domingo")
Estado	TINYINT	1 = Activo, 0 = Inactivo
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro

Atributo	Tipo de dato	Descripción funcional
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro

Nota. Esta tabla es la guía para comprender los atributos, el tipo de dato, estado; que va a tener la tabla con la información días de atención por parte de los negocios y servicios registrados

Horas de atención: Para transformar, estandarizar y normalizar la tabla referente a las horas en que un negocio atiende se crea la tabla “Zn_Atencion_Hora” la cual se aplicaron los siguientes pasos clave:

Se le asigno el atributo de ID único como clave primaria (PK), a las diferentes horas de inicio o fin en que un negocio o servicio presta su servicio. Se agregó el atributo de estado activo (valor 1), inactivo (valor 0).

Tabla 28

Estructura para la Tabla con los Días de Atención “Time_Atencion_Hora”

Atributo	Tipo de dato	Descripción funcional
Id_Atencion_Hora_PK	INT (PK, AI)	Identificador único de la hora
Hora_Inicio	TIME	Hora en formato 24h (ej. 12:00:00, 13:30:00). Se elimina formato con "a.m./p.m."
Atributo	Tipo de dato	Descripción funcional
Hora_Fin	TIME	Hora en formato 24h (ej. 12:00:00, 13:30:00). Se elimina formato con "a.m./p.m."
Estado	TINYINT	1 = Activo, 0 = Inactivo
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro

Atributo	Tipo de dato	Descripción funcional
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro

Nota. Esta tabla es la guía para comprender los atributos, el tipo de dato, estado; que va a tener la tabla con la información de horas de atención por parte de los negocios y servicios registrados.

Transformación Normalización de Archivos de Herramientas de Google

Como parte del proceso de estandarización de los datos provenientes de herramientas como Google, se establece la estructura que van a tener las tablas y su transformación para así poder ser enviadas o cargadas en el sistema de gestión de base de datos.

Aplicando la nomenclatura definida en este proyecto, las tablas que se descargan de las diferentes fuentes de Google reciben los siguientes nombres:

- “Data_GA_Usuarios”
- “Data_GA_Paginas”
- “Data_GADSE_Metricas_Ads”

El informe que se descarga de Google Search Console contiene 5 hojas (Consultas, Páginas, Países, Dispositivos y Fechas), por lo tanto, cada hoja será una entidad independiente y nombradas así:

- “Data_GSC_Consulta”
- “Data_GSC_Pagina”
- “Data_GSC_Pais”
- “Data_GSC_Dispositivo”
- “Data_GSC_Fecha”

La estructura de las tablas provenientes de las herramientas de Google, esta tabla no tiene ajustes relevantes ya que desde la descarga o en la integración por una API, los datos bien optimizados para ser utilizados en hojas de cálculo o cargarlos en el sistema de gestión de bases.

Tabla 29

Estructura para la Tabla de "Data_GA_Usuarios"

Atributo	Tipo de dato	Descripción funcional
Id_Metrica_PK	INT	Clave primaria, autoincremental
Anio	SMALLINT	Número del año (2025)
Atributo	Tipo de dato	Descripción funcional
Mes	TINYINT	Número del mes (1 a 12)
Categoria_Dispositivo	VARCHAR (20)	Tipo de dispositivo (desktop, mobile, tablet)
Usuarios_Activos	INT	Total de usuarios activos en el mes y dispositivo
Usuarios_Nuevos	INT	Total de usuarios nuevos
Sesiones	INT	Número de sesiones
Duracion_Media_Segundos	INT	Duración media de sesión en segundos (convertida)
Porcentaje_Rebote	DECIMAL (5,2)	Tasa de rebote en porcentaje
Usuarios_Recurrentes	INT	Número de usuarios que regresaron

Nota. Esta tabla almacena información diaria agrupada por tipo de dispositivo (mobile, desktop, tablet) proveniente de Google Analytics. Permite analizar el comportamiento de los usuarios por canal de acceso, diferenciando entre nuevos, recurrentes y sesiones. Se incluye la duración media de la sesión en segundos y la tasa de rebote, facilitando evaluaciones de retención y engagement.

Tabla 30

Estructura para la Tabla de "Data_GA_Paginas"

Atributo	Tipo de dato	Descripción funcional
Id_Metrica_Pagina_PK	INT	Clave primaria autoincremental
Anio	SMALLINT	Número del año (2025)
Mes	TINYINT	Número del mes (1–12)
Pagina_Destino	VARCHAR (255)	URL relativa de la página (ej.: /lista/ruta-de-transmilenio...)
Id_URL_Paginas_FK	IN (FK)	Clave foránea que se relaciona con la tabla dim_pagina, para asociar una URL pública
Usuarios_Nuevos	INT	Número de usuarios nuevos
Usuarios_Activos	INT	Total de usuarios activos
Usuarios_Recurrentes	INT	Número de usuarios recurrentes
Sesiones	INT	Total de sesiones en la página
Duracion_Media_Segundos	INT	Duración promedio de la sesión (convertido a segundos)
Sesiones_Usuario_Activo	DECIMAL (4,2)	Promedio de sesiones por usuario activo
Porcentaje_Rebote	DECIMAL (5,2)	Porcentaje de rebote

Nota. Contiene métricas por página de destino del sitio web, permitiendo identificar las URLs más efectivas en términos de visitas, tiempo de permanencia y comportamiento del usuario. Es especialmente útil para contrastar el rendimiento de diferentes secciones del portal (como rutas, comercios o contenido institucional). Se incluye el promedio de sesiones por usuario activo y la tasa de rebote específica por página

Tabla 31

Estructura para la Tabla de "Data_GADSE_Metricas_Ads"

Campo	Tipo de Dato	Descripción
Id_Ingreso (PK)	INT	Identificador único
Anio	SMALLINT	Número del año (2025)
Mes	TINYINT	Número del mes (1–12)
Tipo_Anuncio	VARCHAR (100)	Banner, video, cuadro, etc.
Impresiones	INT	Veces que se mostró el anuncio
Clics	INT	Clics sobre el anuncio
CTR	DECIMAL (5,2)	Porcentaje de rebote

Nota. Registra datos extraídos de Google AdSense clasificados por tipo de anuncio. Esta tabla es clave para evaluar el rendimiento económico del sitio a través de métricas como impresiones, clics y CTR, facilitando el cálculo del retorno por formato publicitario. Se plantea incluir esta tabla dentro de los indicadores comerciales del sitio

Tabla 32

Estructura para la Tabla de "Data_GSC_Consulta"

Campo	Tipo de Dato	Descripción
Id_Consulta (PK)	INT	Identificador único
Anio	SMALLINT	Número del año (2025)
Mes	TINYINT	Número del mes (1–12)
Consulta	VARCHAR (255)	Palabra clave usada por el usuario

Campo	Tipo de Dato	Descripción
Clics	INT	Veces que recibió clic
Impresiones	INT	Impresiones totales de esa página
CTR	DECIMAL (5,2)	Clics / Impresiones * 100
Posicion_Promedio	DECIMAL (5,2)	Posición media en resultados

Nota. Agrupa información por consulta (palabra clave) registrada en el buscador de Google, indicando cuántas veces apareció el sitio como resultado (impresiones), cuántos clics recibió y su posición promedio. Es esencial para evaluar el posicionamiento SEO del sitio y ajustar contenido o etiquetas de manera estratégica.

Tabla 33

Estructura para la Tabla de "Data_GSC_Pagina"

Campo	Tipo de Dato	Descripción
Id_Pagina (PK)	INT	Identificador único
Anio	SMALLINT	Número del año (2025)
Mes	TINYINT	Número del mes (1–12)
Pagina_Destino	VARCHAR (255)	URL de la página del sitio
Id_URL_Paginas_FK	IN (FK)	Clave foránea que se relaciona con la tabla dim_pagina, para asociar una URL pública
Clics	INT	Veces que recibió clic
impresiones	INT	Impresiones totales de esa página

Campo	Tipo de Dato	Descripción
CTR	DECIMAL (5,2)	CTR de esa página
Posicion_Promedio	DECIMAL (5,2)	Posición promedio en buscador

Nota. Relaciona métricas específicas por URL del sitio tal como aparecen en Google Search Console. Permite comparar rendimiento orgánico por página, facilitando la identificación de contenidos mejor posicionados y aquellos que requieren optimización. También permite analizar CTR y posición media por página de destino.

Tabla 34

Estructura para la Tabla de "Data_GSC_Pais"

Campo	Tipo de Dato	Descripción
Id_Pais (PK)	INT	Identificador único
Anio	SMALLINT	Número del año (2025)
Mes	TINYINT	Número del mes (1–12)
Pais	VARCHAR (100)	Nombre del país
Clics	INT	Veces que recibió clic
Impresiones	INT	Impresiones totales de esa página
CTR	DECIMAL (5,2)	Clics / Impresiones * 100
Posicion_Promedio	DECIMAL (5,2)	Posición promedio

Nota. Organiza las interacciones según el país desde el cual se realizaron las búsquedas, lo que permite detectar audiencias internacionales o comportamiento local. Ideal para identificar mercados emergentes o ajustar campañas segmentadas según geografía. Las métricas incluidas permiten evaluar clics, impresiones y efectividad de resultados

Tabla 35

Estructura para la Tabla de "Data_GSC_Dispositivo"

Campo	Tipo de Dato	Descripción
Id_Dispositivo (PK)	INT	Identificador único
Anio	SMALLINT	Número del año (2025)
Mes	TINYINT	Número del mes (1–12)
Dispositivo	VARCHAR (50)	mobile / desktop / tablet
Clics	INT	Impresiones desde ese tipo de dispositivo
impresiones	INT	Clics desde ese tipo de dispositivo
CTR	DECIMAL (5,2)	Clics / Impresiones * 100
Posicion_Promedio	DECIMAL (5,2)	Posición promedio

Nota. Contiene estadísticas separadas por tipo de dispositivo (desktop, mobile, tablet), lo que permite comparar el rendimiento de la presencia del sitio en función del medio de acceso. Útil para decisiones relacionadas con diseño responsivo, experiencia de usuario (UX) y adaptaciones móviles. Incluye CTR y posición promedio

Tabla 36

Estructura para la Tabla de "Data_GSC_Fecha".

Campo	Tipo de Dato	Descripción
Fecha (PK)	DATE	Fecha del registro
Clics	INT	Impresiones desde ese tipo de dispositivo
impresiones	INT	Clics desde ese tipo de dispositivo

Campo	Tipo de Dato	Descripción
CTR	DECIMAL (5,2)	Clics / Impresiones * 100
Posicion_Promedio	DECIMAL (5,2)	Posición promedio

Nota. Registra métricas consolidadas por fecha, proporcionando una vista temporal del rendimiento orgánico del sitio. Es fundamental para detectar picos de tráfico, estacionalidad, campañas efectivas o caídas de posicionamiento.

Transformación y Normalización del Archivo CRM / hoja "CRM"

Como parte del proceso de transformación de los datos provenientes del archivo CRM, se identificó que la hoja denominada "CRM" contiene una mezcla de datos personales, comerciales y operativos, lo cual contraviene los principios de normalización en bases de datos. Por esta razón, se decidió dividir dicha información en entidades separadas.

Como resultado, se definieron dos tablas: la primera almacena los datos personales de la persona que realiza el registro del negocio o servicio; la segunda contiene la información comercial del negocio registrado. Esta última incluirá múltiples claves foráneas (FK), dado que funcionará como una de las tablas principales de consulta para el modelo relacional.

Aplicando la nomenclatura definida en este proyecto, las tablas derivadas reciben los siguientes nombres:

Reg_Datos_Personales - Reg_Datos_Comerciales

Datos personales: Para transformar, estandarizar y normalizar la tabla únicamente con los datos que se obtienen de las personas que registran un negocio se crea la tabla siguiendo la guía de la nomenclatura “Reg_Datos_Personales”

Tabla 37

Estructura para la Tabla de Datos Personales “Reg_Datos_Personales”

Atributo	Tipo de Dato	Descripción funcional
Id_Persona_PK	INT (PK, AI)	Identificador único de la persona que registra el negocio.
Nombres	VARCHAR (100)	Nombre(s) de la persona que realiza el registro.
Apellidos	VARCHAR (100)	Apellidos de la persona que realiza el registro.
Cedula	VARCHAR (20) NULL	Número de identificación personal (se debe anonimizar o cifrar).
Atributo	Tipo de Dato	Descripción funcional
Correo_Electronico	VARCHAR (150) NULL	Correo electrónico de contacto.
Contacto_Celular	VARCHAR (10)	Número de celular principal de contacto.
Contacto_Telefono	VARCHAR (20) NULL	Teléfono fijo (opcional).
Estado	TINYINT	Estado del registro: 1 = Activo, 0 = Inactivo.
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro

Atributo	Tipo de Dato	Descripción funcional
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro

Nota. Esta tabla es la guía de los atributos que debe tener la tabla donde se almacena los datos personales de quien realiza el registro del negocio, sin datos comerciales ni operativos

Se extrajeron únicamente los campos relevantes relacionados con datos personales, tales como nombres, apellidos, documento de identidad, correo electrónico, contacto celular y teléfono fijo (opcional). Además, se incorporó un nuevo campo denominado Estado, con el propósito de llevar control sobre la condición del registro, indicando si el negocio ha solicitado su retiro, ha cambiado de persona de contacto o si permanece activo (1) o inactivo (0). También se añadieron los campos Fecha_Creacion y Fecha_Actualizacion, que permiten llevar trazabilidad sobre el tiempo de permanencia en el sistema y las modificaciones realizadas. En este contexto, se creó la tabla Reg_Datos_Comerciales, la cual representa gran relevancia del sistema, al contener la información estructurada de los comercios registrados en el sitio web DirectorioUsme.com. Esta información, originalmente desorganizada en la hoja “CRM” del archivo CRM, fue sometida a un riguroso proceso de transformación y normalización, permitiendo su conversión en una tabla relacional optimizada, apta para consultas, integraciones y visualizaciones.

Datos comerciales: El núcleo del sistema contiene la información estructurada de comercios registrados. Se realizo un proceso riguroso de transformación y normalización, que permitió organizar y optimizar, lista para consultas, integraciones y visualización.

Tabla 38

Estructura para la Tabla de Datos Personales "Reg_Datos_Comerciales"

Atributo	Tipo de Dato	Descripción funcional
Id_Comercial_PK	INT (PK, AI)	Identificador único del negocio.
Id_Persona_FK	INT (FK)	Relación con la persona que registró el negocio tabla (Reg_Datos_Personales).
Nombre_Comercial	VARCHAR (120)	Nombre del negocio como se muestra en el directorio.
Estado	TINYINT	Estado: 1 = Activo, 0 = Inactivo.
Direccion	VARCHAR (150)	Dirección física del negocio.
Correo_Electronico	VARCHAR (150)	Correo electrónico del negocio.
Contacto_Celular_1	VARCHAR (10)	Celular principal de contacto.
Contacto_Celular_2	VARCHAR (10)	Celular adicional (opcional).
Contacto_Telefono	VARCHAR (10)	Teléfono fijo del negocio (opcional).
Id_Barrio_PK	INT (FK)	Clave foránea que relaciona el barrio (Zn_Ubicacion_Barrio).
Id_Categoria_L1_FK	INT (FK)	Relación con la categoría principal (Cat_Categoria_L1).
Id_URL_Paginas_FK	IN (FK)	Clave foránea que se relaciona con la tabla dim_pagina, para asociar una URL pública
Descripcion	TEXT	Campo para descripción del negocio.

Atributo	Tipo de Dato	Descripción funcional
Observacion	TEXT	Campo para observaciones internar (opcional)
Fecha_Creacion	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro

Nota. Esta tabla es la guía de los atributos que debe tener la tabla donde se almacena los datos comerciales de los negocios o servicios registrados, se destaca que es de las tablas más extensas y relevantes de todo el proyecto y se incluyen varias claves foráneas.

Como parte del proceso de normalización de la tabla Reg_Datos_Comerciales, se aplicaron los principios hasta la Tercera Forma Normal (3FN).

En la Primera Forma Normal (1FN), se eliminaron datos redundantes y columnas con valores compuestos, separando, por ejemplo, los números de contacto en campos individuales (Contacto_Celular_1, Contacto_Celular_2 y Contacto_Telefono).

En la Segunda Forma Normal (2FN), se garantizó que todos los atributos dependieran únicamente de la clave primaria Id_Comercial_PK, mientras que en la Tercera Forma Normal (3FN), se reemplazaron dependencias transitivas utilizando claves foráneas para enlazar con otras entidades como Cat_Categoria_L1, Zn_Ubicacion_Barrio y Reg_Datos_Personales

Asimismo, se realizaron labores de depuración para corregir errores tipográficos, estandarizar formatos y eliminar caracteres no válidos en campos clave como correos, teléfonos y nombres comerciales.

Se incorporaron también metadatos como Estado, Fecha_Creacion y Fecha_Actualizacion para permitir trazabilidad y control de cada registro, y se añadió el campo Url_Slug para facilitar la generación de enlaces amigables con fines de SEO. Este proceso dejó como resultado una tabla robusta, relacional y alineada con las buenas prácticas de diseño de base de datos.

Cierre Fase 3: El desarrollo de la Fase 3 ha sido fundamental para el éxito del proyecto, al permitir transformar datos dispersos, inconsistentes y poco estructurados en un modelo relacional claro, coherente y escalable. A lo largo de esta etapa, se identificaron y depuraron atributos clave, se definieron los tipos de datos apropiados para cada variable y se diseñaron tablas independientes que reflejan adecuadamente las entidades del dominio. Como resultado, se obtuvo una base sólida sobre la cual será posible construir relaciones lógicas, generar vistas analíticas confiables y ejecutar procesos de integración y modelado sin redundancia, ambigüedad ni pérdida de información.

Esta estructura, además de optimizar la eficiencia del sistema, facilita futuras expansiones y garantiza la calidad de los datos que alimentarán la toma de decisiones estratégicas en el DirectorioUsme.com.

Fase 4 Modelado

En esta fase se consolida el diseño lógico y físico del sistema de gestión de bases de datos del proyecto DirectorioUsme.com, con el fin de estructurar formalmente las entidades, atributos y relaciones que representarán la información previamente depurada.

Selección de Técnica de Modelado

Luego de haber identificado y preparado las tablas necesarias en la fase anterior, se procedió a seleccionar la técnica de modelado más adecuada para representar y gestionar los

datos de manera estructurada. Considerando que la información del sitio DirectorioUsme.com se compone de registros tabulares interrelacionados como datos personales, comerciales, categorías, subcategorías, barrios, horarios y etiquetas, se determinó que la técnica más apropiada era el **modelo relacional**, respaldado por un Sistema de Gestión de Bases de Datos (SGBD).

El modelo relacional permite organizar la información en tablas interconectadas mediante claves primarias y foráneas, lo que garantiza la integridad referencial, la consistencia de los datos y la escalabilidad del sistema. Esta técnica resulta especialmente adecuada para entornos donde la información debe categorizarse, jerarquizarse y consultarse de forma eficiente, como es el caso del DirectorioUsme.com.

En este contexto, el uso de un **Sistema de Gestión de Bases de Datos (SGBD)** cobra especial relevancia, ya que actúa como un intermediario entre el usuario y la base de datos, operando como una interfaz que facilita el acceso de la información de manera eficiente y segura (Soberón & Jesús, 2020). Este tipo de sistema resulta indispensable cuando la escala del proyecto prevé manejar grandes volúmenes de datos relacionados con categorías, subcategorías y registros de negocios locales.

Tal como lo indica (Torres, 2021), un SGBD se vuelve crítico cuando la escala del sistema alcanza “centenares de gigabytes y millares de usuarios”, escenarios que pueden ser aplicables en sistemas comunitarios en crecimiento como DirectorioUsme.com.

Con base en este enfoque, se definieron los siguientes lineamientos para el modelado de datos:

Aplicación de normalización hasta la Tercera Forma Normal (3FN), con el fin de evitar redundancias, mejorar el almacenamiento y asegurar una estructura lógica coherente.

Construcción de un modelo entidad-relación (ER) como representación gráfica del sistema, para validar las relaciones antes de su implementación técnica.

Elección del SGBD **MySQL Workbench**, debido a su compatibilidad con herramientas de visualización de métricas como Power BI, su integración fluida con Python para procesos ETL automatizados, y su soporte robusto para SQL estándar, lo que facilita tanto la gestión como la consulta de datos.

Estos lineamientos permitieron establecer la base técnica del sistema de gestión de datos desarrollado, asegurando que la estructura relacional implementada cumpla con los requisitos de calidad, rendimiento y análisis definidos desde la comprensión del negocio.

Diagrama Entidad Relación (DER): Este diagrama permite visualizar de manera clara las entidades principales, sus atributos, y las relaciones que existen entre ellas, sirviendo como base para la posterior implementación técnica en el SGBD MySQL. En la siguiente página la Figura 30 se observa la composición y relación de todas las tablas

Figura 30

Diagrama Entidad-Relación (DER) del Sistema de Gestión de Datos

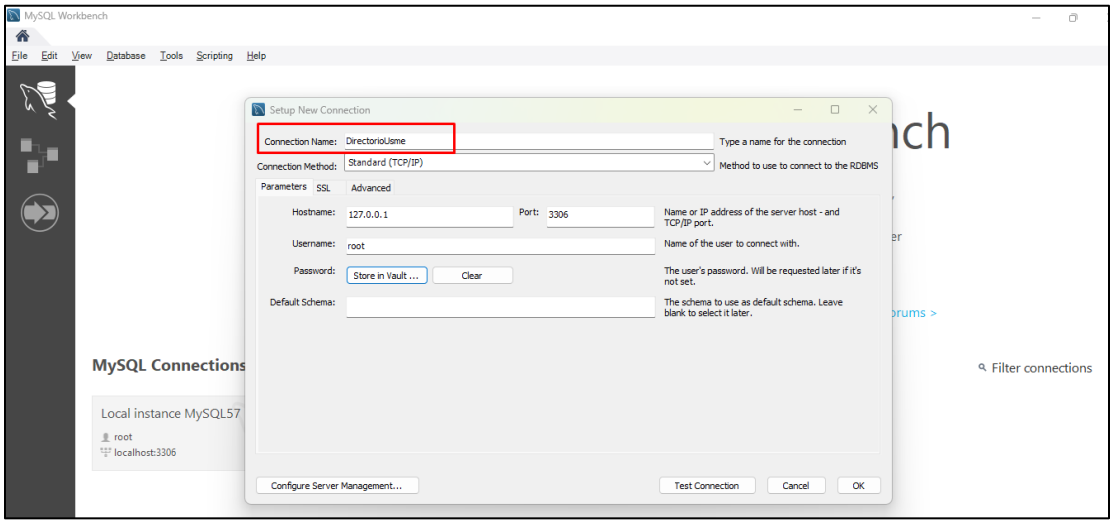
Nota. El diagrama muestra la estructura lógica de la base de datos relacional del DirectorioUsme.com, con sus respectivas entidades, atributos, claves primarias y claves foráneas. Elaboración propia a partir de la normalización y modelado del sistema. Esto se realizó en la aplicación dbdiagram.io

El DER incluye tanto las tablas principales que almacenan datos personales y comerciales, como las tablas categóricas. Además, se identificó la necesidad de agregar tablas intermedias, dado que existen relaciones de tipo muchos a muchos entre varias entidades. Asimismo, el diagrama refleja las claves primarias, claves foráneas y la cardinalidad de las relaciones, lo cual facilita la validación de la integridad referencial y el cumplimiento de las reglas de normalización adoptadas en el modelo.

Las entidades en el sistema de gestión de bases de datos MySQL *Workbench* se realizó tomando como base la estructura definida en la preparación de los datos y en el modelo relacional (DER). A continuación, se muestra el proceso de creación de la base de datos que inicia con la construcción física del modelo, en cumplimiento de lo diseñado conceptualmente en fases anteriores.

Figura 31

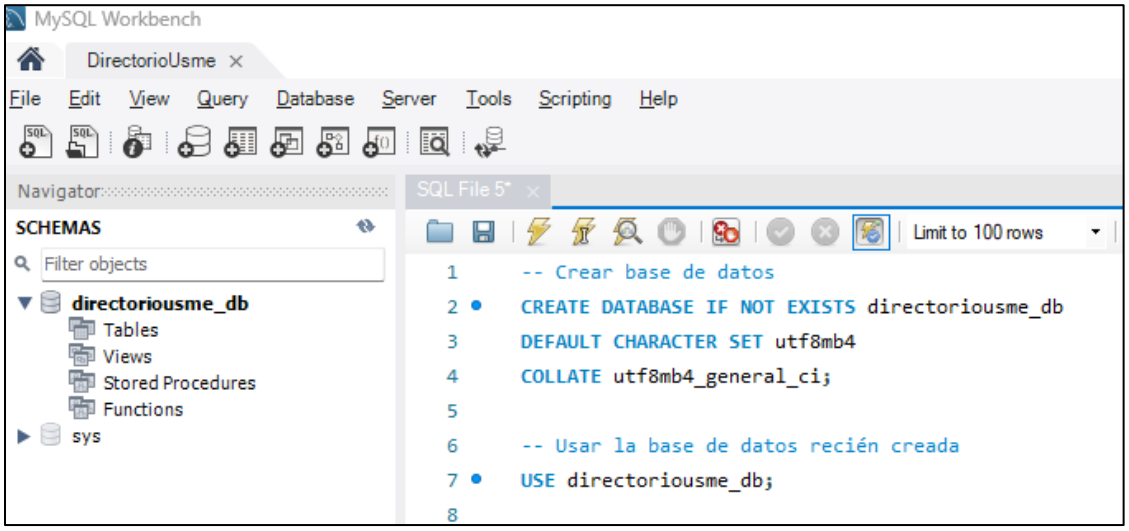
Creación de la Base de Datos Mediante Interfaz Gráfica en Mysql Workbench



Nota. Elaboración propia, se asigna a la base de datos el nombre de DirectorioUsme

Figura 32

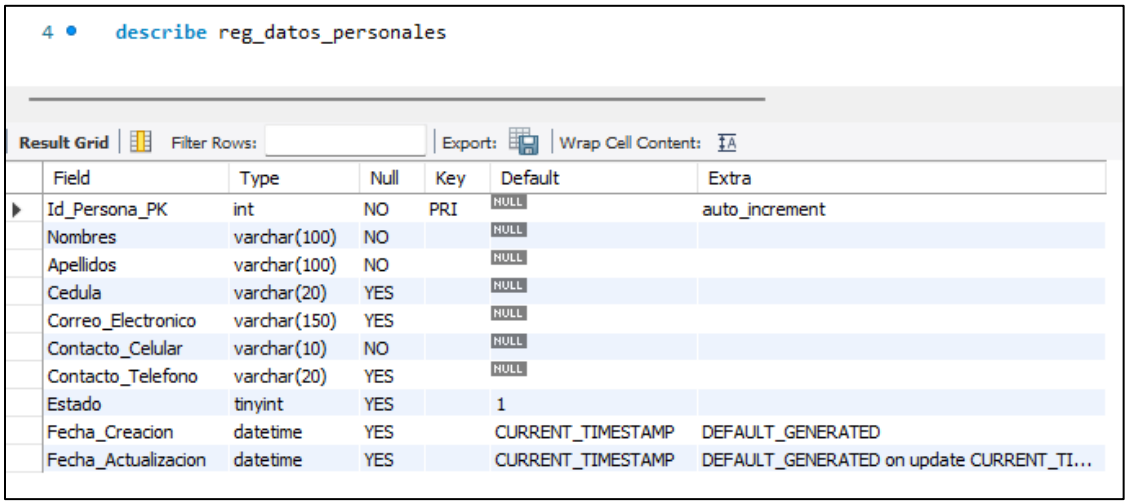
Sentencia SQL de Creación Base de Datos



Nota. Sentencia SQL utilizada para la creación de la base de datos directoriousme_db

Figura 33

Estructura de la Tabla Implementada - Reg_Datos_Personales



Field	Type	Null	Key	Default	Extra
Id_Persona_PK	int	NO	PRI	NULL	auto_increment
Nombres	varchar(100)	NO		NULL	
Apellidos	varchar(100)	NO		NULL	
Cedula	varchar(20)	YES		NULL	
Correo_Electronico	varchar(150)	YES		NULL	
Contacto_Celular	varchar(10)	NO		NULL	
Contacto_Telefono	varchar(20)	YES		NULL	
Estado	tinyint	YES		1	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Tabla creada en MySQL con la nomenclatura y estructura previamente establecida

Figura 34

Estructura de la Tabla Implementada - Reg_Datos_Comerciales.

```
5 •    describe reg_datos_comerciales
```

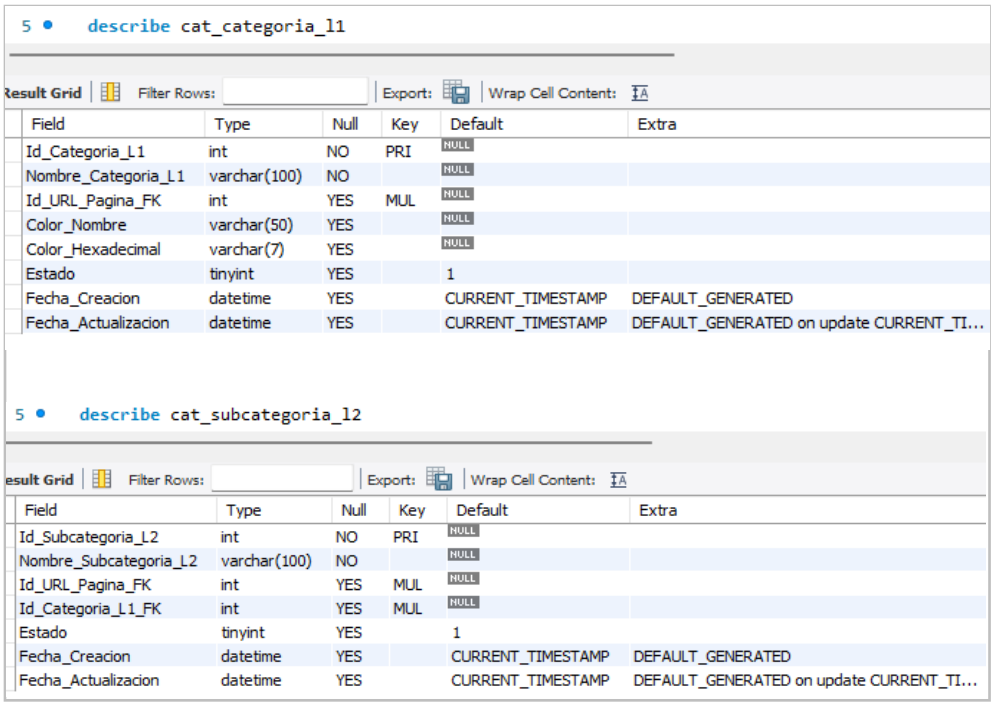
esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Comercial_PK	int	NO	PRI	NULL	auto_increment
Id_Persona_FK	int	YES	MUL	NULL	
Nombre_Comercial	varchar(120)	NO		NULL	
Estado	tinyint	YES		1	
Direccion	varchar(150)	NO		NULL	
Correo_Electronico	varchar(150)	YES		NULL	
Contacto_Celular_1	varchar(10)	NO		NULL	
Contacto_Celular_2	varchar(10)	YES		NULL	
Contacto_Telefono	varchar(10)	YES		NULL	
Id_Barrio_FK	int	YES	MUL	NULL	
Id_Categoria_L1_FK	int	YES	MUL	NULL	
Id_URL_Pagina_FK	int	YES	MUL	NULL	
Descripcion	text	YES		NULL	
Observacion	text	YES		NULL	
Fecha_Registro	datetime	YES		CURR...	DEFAULT_GEN...
Fecha_Actualizacion	datetime	YES		CURR...	DEFAULT_GEN...

Nota. Tabla creada en MySQL con la nomenclatura y estructura previamente establecida. Una de las tablas principales de la base de datos con varias FK, como de persona, barrio, categoría, Url de página

Figura 35

Estructura de las Tablas Implementadas - Categoria L1 y Subcategoria L2



```
5 •  describe cat_categoria_l1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Categoria_L1	int	NO	PRI	NULL	
Nombre_Categoria_L1	varchar(100)	NO		NULL	
Id_URL_Pagina_FK	int	YES	MUL	NULL	
Color_Nombre	varchar(50)	YES		NULL	
Color_Hexadecimal	varchar(7)	YES		NULL	
Estado	tinyint	YES		1	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

```
5 •  describe cat_subcategoria_l2
```

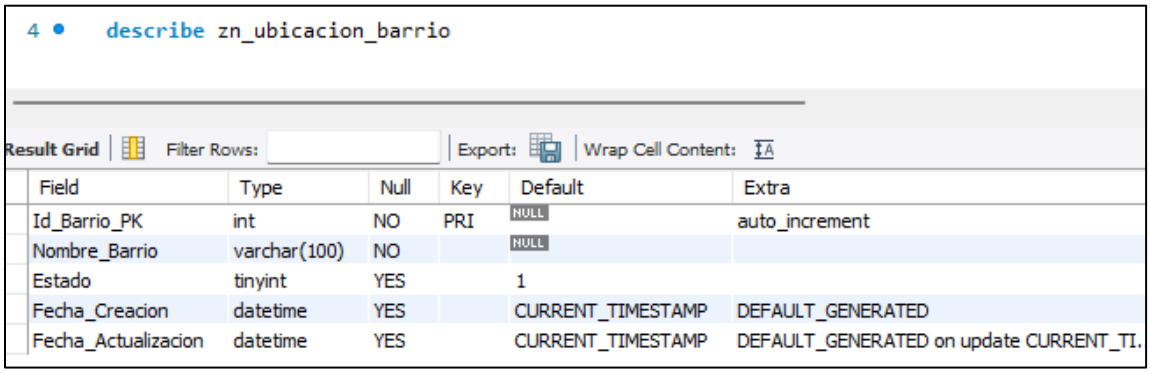
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Subcategoria_L2	int	NO	PRI	NULL	
Nombre_Subcategoria_L2	varchar(100)	NO		NULL	
Id_URL_Pagina_FK	int	YES	MUL	NULL	
Id_Categoria_L1_FK	int	YES	MUL	NULL	
Estado	tinyint	YES		1	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Las tablas fueron creadas con la nomenclatura y estructura previamente establecida, cada una con su correspondiente PK y FK para relacionar la información

Figura 36

Estructura de la Tabla Implementada - Zn_Ubicacion_Barrio.



```
4 •  describe zn_ubicacion_barrio
```

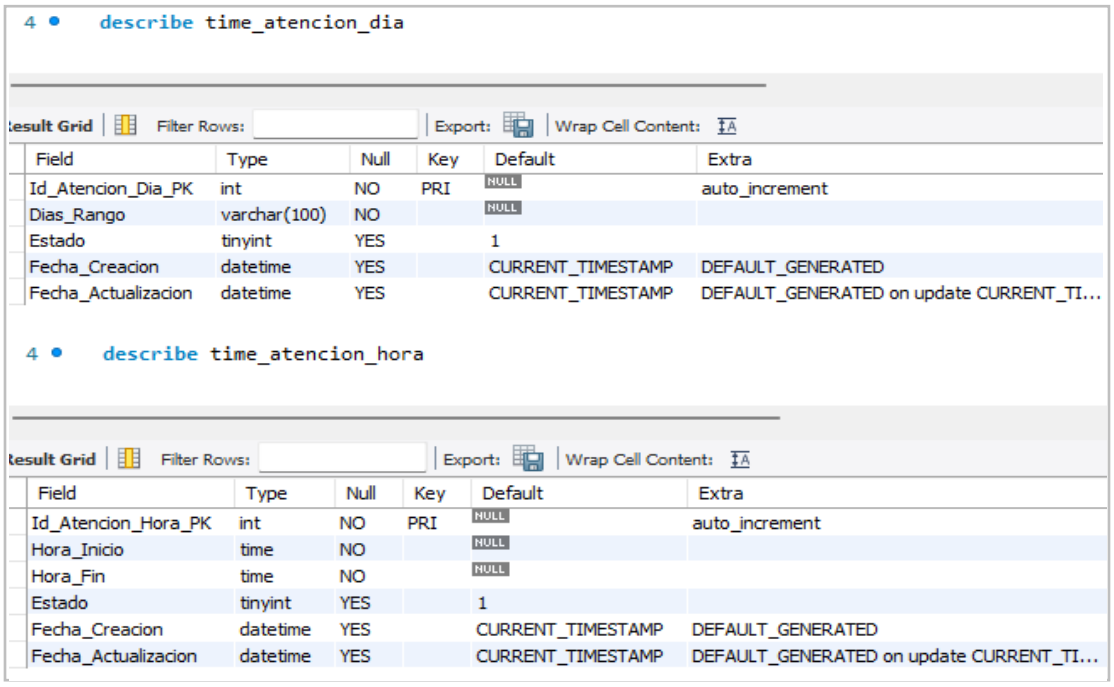
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Barrio_PK	int	NO	PRI	NULL	auto_increment
Nombre_Barrio	varchar(100)	NO		NULL	
Estado	tinyint	YES		1	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI.

Nota. Tabla creada en MySQL con la nomenclatura y estructura previamente establecida

Figura 37

Estructura de Tabla Time_Atencion_Dia y Time_Atencion_Hora



```
4 •  describe time_atencion_dia
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Atencion_Dia_PK	int	NO	PRI	NULL	auto_increment
Dias_Rango	varchar(100)	NO		NULL	
Estado	tinyint	YES		1	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

```
4 •  describe time_atencion_hora
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Atencion_Hora_PK	int	NO	PRI	NULL	auto_increment
Hora_Inicio	time	NO		NULL	
Hora_Fin	time	NO		NULL	
Estado	tinyint	YES		1	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Las tablas fueron creadas con la nomenclatura y estructura previamente establecida

Creación de Tablas Intermedias. Durante la implementación del modelo relacional, se identificó la necesidad de crear tablas intermedias o tablas puente para resolver adecuadamente relaciones de tipo muchos a muchos (N:M) entre los negocios registrados y diversas dimensiones adicionales.

Por ejemplo, un mismo comercio puede ofrecer varios beneficios, aceptar múltiples métodos de pago, o brindar distintos servicios extra. Estas relaciones no pueden representarse directamente en una sola tabla sin generar redundancia, por lo que se diseñaron estructuras independientes que vinculan los registros comerciales con sus correspondientes atributos mediante claves foráneas.

Figura 38

Tabla Resumen de Tablas Intermedias

Nombre de la tabla	Descripción
Inter_Datos_Comerciales_Subcategoria_L2	Relaciona cada comercio con una o más subcategorías del catálogo.
Inter_Datos_Comerciales_Metodos_Pago	Relaciona cada comercio con uno o más métodos de pago aceptados por cada negocio (ej. Nequi, Daviplata, efectivo).
Inter_Datos_Comerciales_Beneficios_Comerciales	Relaciona cada comercio con uno o más beneficios como reservas en línea o domicilios.
Inter_Datos_Comerciales_Servicios_Extra	Relaciona cada comercio con uno o más servicios adicionales como corresponsal bancario o recarga TuLlave.
Inter_Datos_Comerciales_Atencion_Dia_Hora	Relaciona a cada comercio con los días y horas en los que presta atención al público.

Nota. Las tablas intermedias permiten mantener un modelo escalable, reutilizable y preparado para análisis cruzados posteriores

Estas tablas intermedias permiten: Mantener la integridad referencial, evitar la duplicidad de datos, facilitar el análisis multidimensional y escalar fácilmente el modelo ante nuevas categorías o servicios.

Tabla 39

Estructura para una Tabla Intermedia

Atributo	Tipo de dato	Descripción funcional
Id_Comercial_FK	INT	Clave foránea que hace referencia al comercio registrado en Reg_Datos_Comerciales.
Id_<Nombre_Dimension>_FK	INT	Clave foránea que se relaciona con la tabla correspondiente (subcategoría, etiqueta).
Estado	TINYINT	Estado de la relación: 1 = activo, 0 = inactivo.
Fecha_Registro	CURRENT TIMESTAMP	Fecha de creación del registro y almacenas la hora exacta del registro
Fecha_Actualizacion	CURRENT TIMESTAMP	Fecha de creación de modificación y almacenas la hora exacta del registro

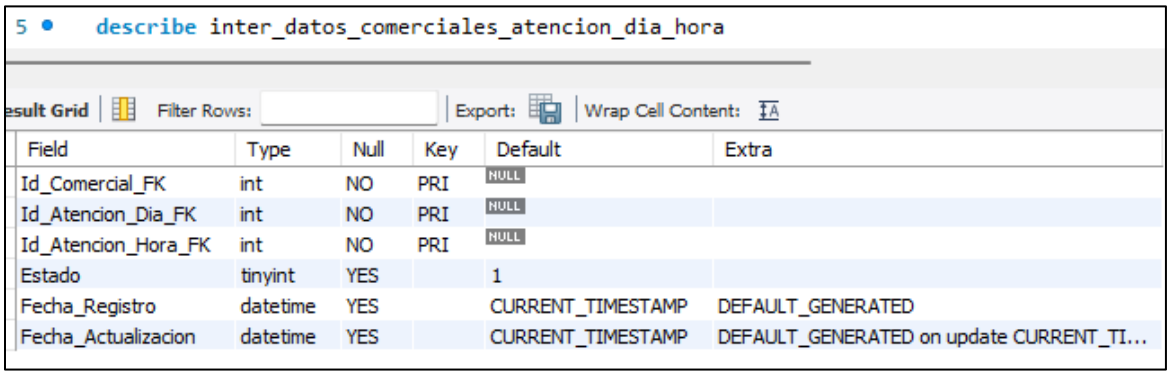
Nota. Esta tabla tiene la estructura que deben tener todas las tablas intermedias o puente para conectar tablas de muchos a muchos (N:M).

La estructura de la tabla es la guía en la composición que se debe tener en las tablas intermedia (relaciones muchos a muchos), solo se debe cambiar el nombre de la dimensión (por ejemplo: subcategoría, método de pago, beneficio o servicio extra) Adicional se crea el uso de clave primaria compuesta: para evitar duplicados silenciosos, asegurar consistencia lógica en

relaciones N:M. Este diseño estandarizado facilita la documentación y el manejo de todas las relaciones N:M en el proyecto, proporcionando una base de datos limpia y eficiente.

Figura 39

Estructura de la Tabla- Inter_Datos_Comerciales_Atencion_Dia_Hora



```
5 •   describe inter_datos_comerciales_atencion_dia_hora
```

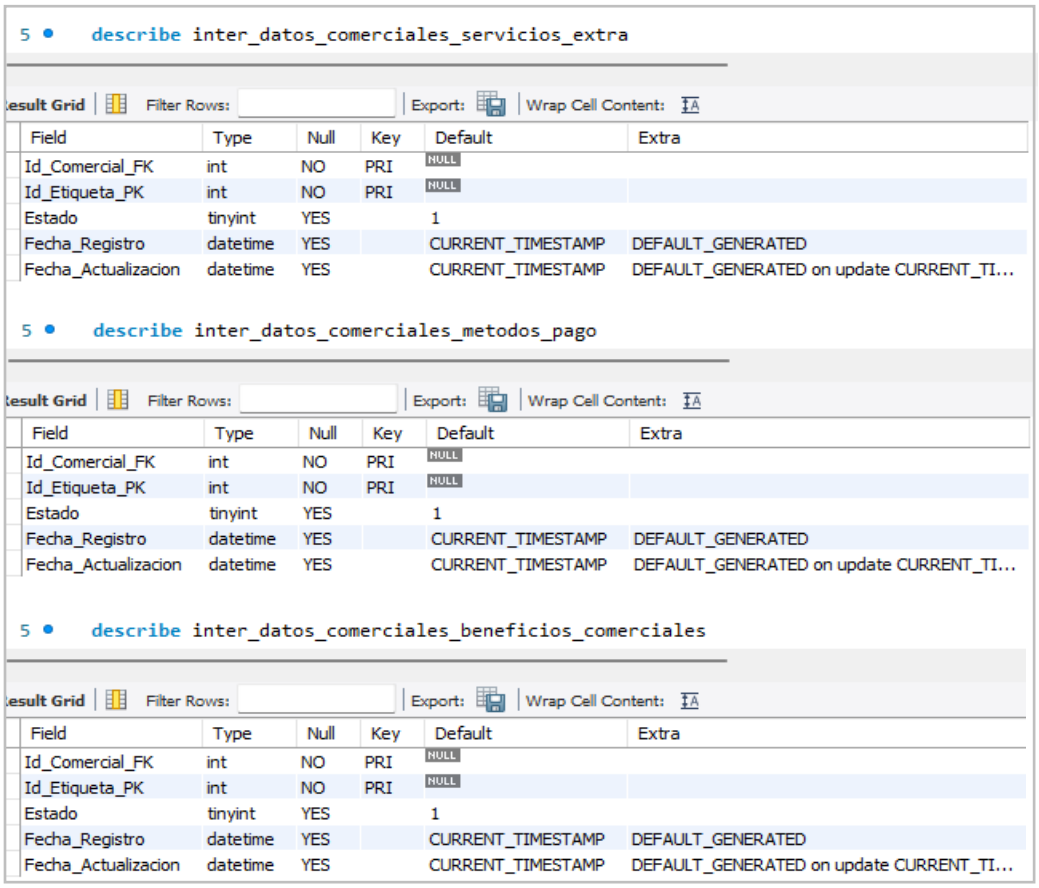
esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Comercial_FK	int	NO	PRI	NULL	
Id_Atencion_Dia_FK	int	NO	PRI	NULL	
Id_Atencion_Hora_FK	int	NO	PRI	NULL	
Estado	tinyint	YES		1	
Fecha_Registro	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Tabla creada en MySQL con la nomenclatura y estructura previamente establecida, esta tabla es el puente para cruzar la información de los días de atención, las horas de atención con los datos comerciales

Figura 40

Estructura de Tablas Catalogadas Como Intermedias Relacionadas a Etiquetas



```
5 •  describe inter_datos_comerciales_servicios_extra
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Comercial_FK	int	NO	PRI	NULL	
Id_Etiqueta_PK	int	NO	PRI	NULL	
Estado	tinyint	YES		1	
Fecha_Registro	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

```
5 •  describe inter_datos_comerciales_metodos_pago
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Comercial_FK	int	NO	PRI	NULL	
Id_Etiqueta_PK	int	NO	PRI	NULL	
Estado	tinyint	YES		1	
Fecha_Registro	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

```
5 •  describe inter_datos_comerciales_beneficios_comerciales
```

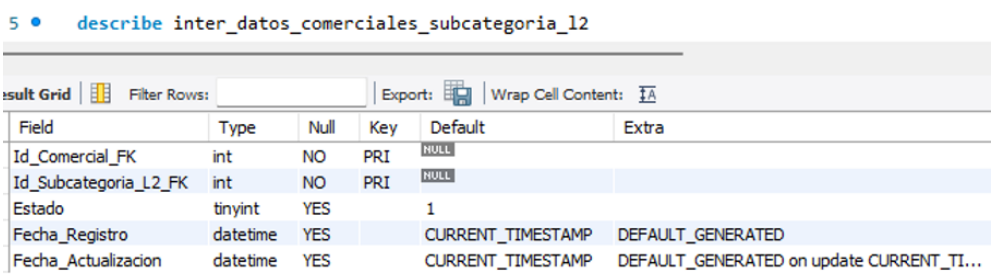
esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Comercial_FK	int	NO	PRI	NULL	
Id_Etiqueta_PK	int	NO	PRI	NULL	
Estado	tinyint	YES		1	
Fecha_Registro	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Las tablas fueron creadas con la nomenclatura y estructura previamente establecida

Figura 41

Estructura de Tabla Implementada - inter_datos_comerciales_sibcategorias_l2.



```
5 •  describe inter_datos_comerciales_subcategoria_l2
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Comercial_FK	int	NO	PRI	NULL	
Id_Subcategoria_L2_FK	int	NO	PRI	NULL	
Estado	tinyint	YES		1	
Fecha_Registro	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Tabla creada en MySQL con la nomenclatura y estructura previamente establecida

Tablas de Fuentes Externas. Se incorporaron tablas destinadas en almacenar información proveniente de fuentes externas, con el objetivo de enriquecer el análisis del comportamiento digital de los negocios registrados en el DirectorioUsme.com.

Figura 42

Estructura de la Tabla Implementada - Data_GA_Paginas

```
5 •   describe data_ga_paginas
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Metrica_Pagina_PK	int	NO	PRI	NULL	auto_increment
Anio	smallint	NO	MUL	NULL	
Mes	tinyint	NO		NULL	
Dia	tinyint	YES		NULL	
Pagina_Destino	varchar(255)	NO		NULL	
Id_URL_Pagina_FK	int	YES	MUL	NULL	
Usuarios_Nuevos	int	YES		0	
Usuarios_Activos	int	YES		0	
Usuarios_Recurrentes	int	YES		0	
Sesiones	int	YES		0	
Duracion_Media_Seg...	int	YES		0	
Sesiones_Usuario_Ac...	decimal(4,2)	YES		0.00	
Porcentaje_Rebote	decimal(5,2)	YES		0.00	
Fecha_Creacion	datetime	YES		CURR...	DEFAULT_GEN...
Fecha_Actualizacion	datetime	YES		CURR...	DEFAULT_GEN...

Nota. Tabla creada en MySQL con la nomenclatura y estructura previamente establecida. Esta tabla contiene la columna Id_Pagina_FK que funciona para aplicar relaciones con otras tablas que también tienen esta columna,

Figura 43

Estructura de la Tabla Implementada - Data_GA_Usuarios

```
5 •   describe data_ga_usuarios
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Metrica_PK	int	NO	PRI	NULL	auto_increment
Anio	smallint	NO	MUL	NULL	
Mes	tinyint	NO		NULL	
Dia	tinyint	YES		NULL	
Categoria_Dispositivo	varchar(20)	NO		NULL	
Usuarios_Activos	int	YES		0	
Usuarios_Nuevos	int	YES		0	
Sesiones	int	YES		0	
Duracion_Media_Segundos	int	YES		0	
Porcentaje_Rebote	decimal(5,2)	YES		0.00	
Usuarios_Recurrentes	int	YES		0	
Fecha_Creacion	datetime	YES		CURR...	DEFAULT_GEN...
Fecha_Actualizacion	datetime	YES		CURR...	DEFAULT_GEN...

Nota. Tabla creada en MySQL siguiendo la nomenclatura y estructura previamente definida. Incluye columnas de año (Anio) y mes (Mes), las cuales permiten establecer relaciones mediante JOIN con otras tablas que comparten estos campos, facilitando así el análisis temporal de métricas como usuarios activos, sesiones o porcentaje de rebote, entre otros

Figura 44

Estructura de la Tabla Implementada – Data_GADSE_Metricas_Ads

```
5 •   describe data_gadse_metricas_ads
```

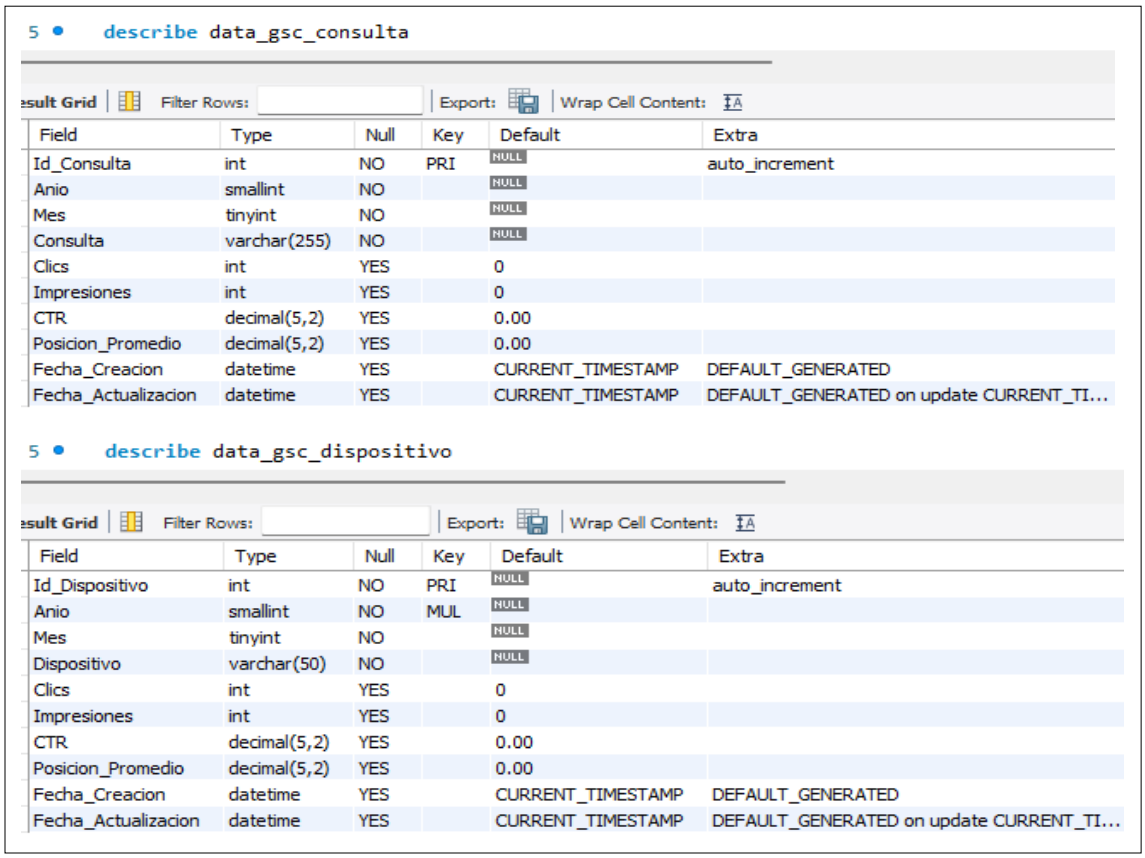
esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Ingreso	int	NO	PRI	NULL	auto_increment
Anio	smallint	NO		NULL	
Mes	tinyint	NO		NULL	
Tipo_Anuncio	varchar(100)	NO		NULL	
Impresiones	int	YES		0	
Clics	int	YES		0	
CTR	decimal(5,2)	YES		0.00	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Tabla creada en MySQL siguiendo la nomenclatura y estructura previamente definida. Incluye columnas de año (Anio) y mes (Mes), las cuales permiten establecer relaciones mediante JOIN con otras tablas que comparten estos campos, facilitando así el análisis temporal

Figura 45

Estructura de las Tablas Data_GSC_Consulta y Data_GSC_Dispositivo



```
5 •  describe data_gsc_consulta
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Consulta	int	NO	PRI	NULL	auto_increment
Anio	smallint	NO		NULL	
Mes	tinyint	NO		NULL	
Consulta	varchar(255)	NO		NULL	
Clics	int	YES		0	
Impresiones	int	YES		0	
CTR	decimal(5,2)	YES		0.00	
Posicion_Promedio	decimal(5,2)	YES		0.00	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

```
5 •  describe data_gsc_dispositivo
```

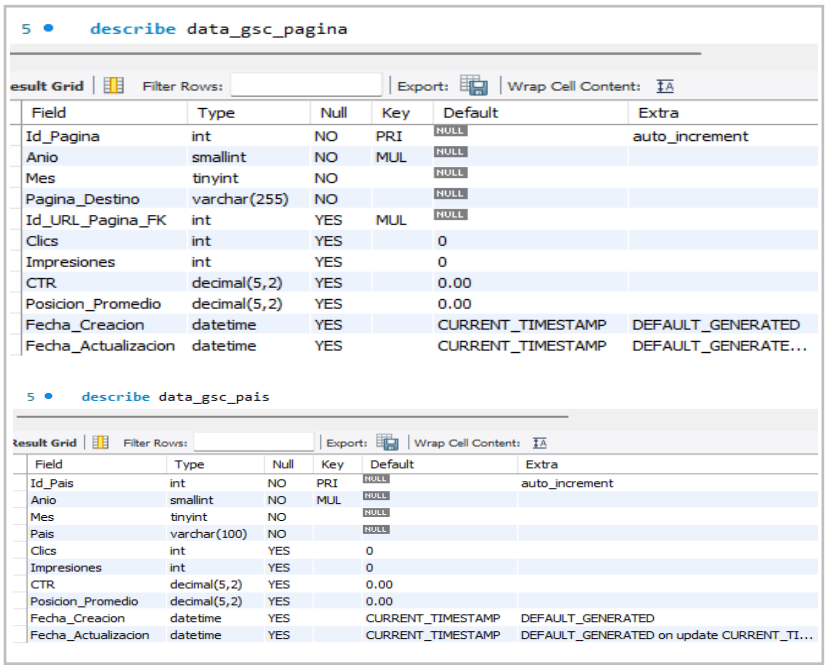
esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Dispositivo	int	NO	PRI	NULL	auto_increment
Anio	smallint	NO	MUL	NULL	
Mes	tinyint	NO		NULL	
Dispositivo	varchar(50)	NO		NULL	
Clics	int	YES		0	
Impresiones	int	YES		0	
CTR	decimal(5,2)	YES		0.00	
Posicion_Promedio	decimal(5,2)	YES		0.00	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Tablas creadas en MySQL siguiendo la nomenclatura y estructura previamente definida. Incluye columnas de año (Anio) y mes (Mes), las cuales permiten establecer relaciones mediante JOIN con otras tablas que comparten estos campos

Figura 46

Estructura de las Tablas Implementadas – Data_Gsc_Pagina y Data_Gsc_Pais



```
5 •  describe data_gsc_pagina
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Pagina	int	NO	PRI	NULL	auto_increment
Anio	smallint	NO	MUL	NULL	
Mes	tinyint	NO		NULL	
Pagina_Destino	varchar(255)	NO		NULL	
Id_URL_Pagina_FK	int	YES	MUL	NULL	
Clics	int	YES		0	
Impresiones	int	YES		0	
CTR	decimal(5,2)	YES		0.00	
Posicion_Promedio	decimal(5,2)	YES		0.00	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATE...

```
5 •  describe data_gsc_pais
```

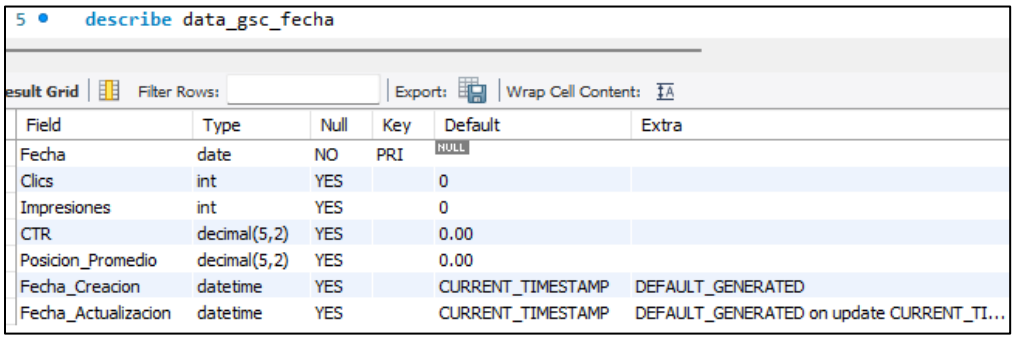
esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Id_Pais	int	NO	PRI	NULL	auto_increment
Anio	smallint	NO	MUL	NULL	
Mes	tinyint	NO		NULL	
Pais	varchar(100)	NO		NULL	
Clics	int	YES		0	
Impresiones	int	YES		0	
CTR	decimal(5,2)	YES		0.00	
Posicion_Promedio	decimal(5,2)	YES		0.00	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Tablas creadas en MySQL siguiendo la nomenclatura y estructura previamente definida. Incluye columnas de año (Anio) y mes (Mes), las cuales permiten establecer relaciones mediante JOIN con otras tablas que comparten estos campos.

Figura 47

Estructura de las Tablas Implementadas – Data_GSC_Fechas.



```
5 •  describe data_gsc_fecha
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Fecha	date	NO	PRI	NULL	
Clics	int	YES		0	
Impresiones	int	YES		0	
CTR	decimal(5,2)	YES		0.00	
Posicion_Promedio	decimal(5,2)	YES		0.00	
Fecha_Creacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
Fecha_Actualizacion	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED on update CURRENT_TI...

Nota. Tablas creadas en MySQL siguiendo la nomenclatura y estructura previamente definida

Validación de relaciones y claves foráneas

Tabla 40

Resumen de Relaciones entre Entidades

Tabla Origen	Atributo Común	Tabla Destino	Comentario técnico
Reg_Datos_Comerciales	Id_Persona_FK	Reg_Datos_Personales	Cada comercio está asociado a una sola persona
Reg_Datos_Comerciales	Id_Barrio_FK	Zn_Ubicacion_Barrio	Relaciona el comercio con su barrio
Reg_Datos_Comerciales	Id_Categoria_L1_FK	Cat_Categoria_L1	Define la categoría principal del comercio
Cat_Subcategoria_L2	Id_Categoria_L1_FK	Cat_Categoria_L1	Subcategorías dependen de una categoría principal
Inter_Datos_Comerciales_Subcategoria_L2	Id_Comercial_FK, Id_Subcategoria_L2_FK	Reg_Datos_Comerciales, Cat_Subcategoria_L2	Asocia comercios con múltiples subcategorías
Inter_Datos_Comerciales_Servicios_Extra	Id_Comercial_FK, Id_Etiqueta_FK	Reg_Datos_Comerciales, Etq_Servicios_Extra	Asocia servicios adicionales a comercios
Inter_Datos_Comerciales_Metodos_Pago	Id_Comercial_FK, Id_Etiqueta_FK	Reg_Datos_Comerciales, Etq_Metodos_Pago	Comercios pueden aceptar múltiples métodos de pago
Inter_Datos_Comerciales_Beneficios_Comerciales	Id_Comercial_FK, Id_Etiqueta_FK	Reg_Datos_Comerciales, Etq_Beneficios_Comerciales	Relación con beneficios especiales ofrecidos
Inter_Datos_Comerciales_Atencion_Dia_Hora	Id_Comercial_FK Id_Atencion_Dia_FK Id_Atencion_Hora_FK	Reg_Datos_Comerciales Time_Atencion_Dia Time_Atencion_Hora	Horarios por día específicos en que el comercio atiende
Data_GA_Paginas	Id_URL_Pagina_FK	Dim_Pagina	Asocia métricas de páginas con una URL única
Search_console_paginas	Id_URL_Pagina_FK	Dim_Pagina	Páginas posicionadas en buscador cruzadas con catálogo de páginas

Tabla Origen	Atributo Común	Tabla Destino	Comentario técnico
Reg_Datos_Comerciales	Id_URL_Pagina_FK	Dim_Pagina	Comercio está vinculado a una URL del sitio
Cat_Categoria_L1	Id_URL_Pagina_FK	Dim_Pagina	Cada categoría está publicada como página
Cat_Subcategoria_L2	Id_URL_Pagina_FK	Dim_Pagina	Subcategorías también representan una página navegable

Nota. La tabla resume las relaciones clave entre entidades del modelo, indicando los atributos comunes utilizados, el tipo de relación establecida y su propósito técnico dentro de la estructura del sistema de base de datos.

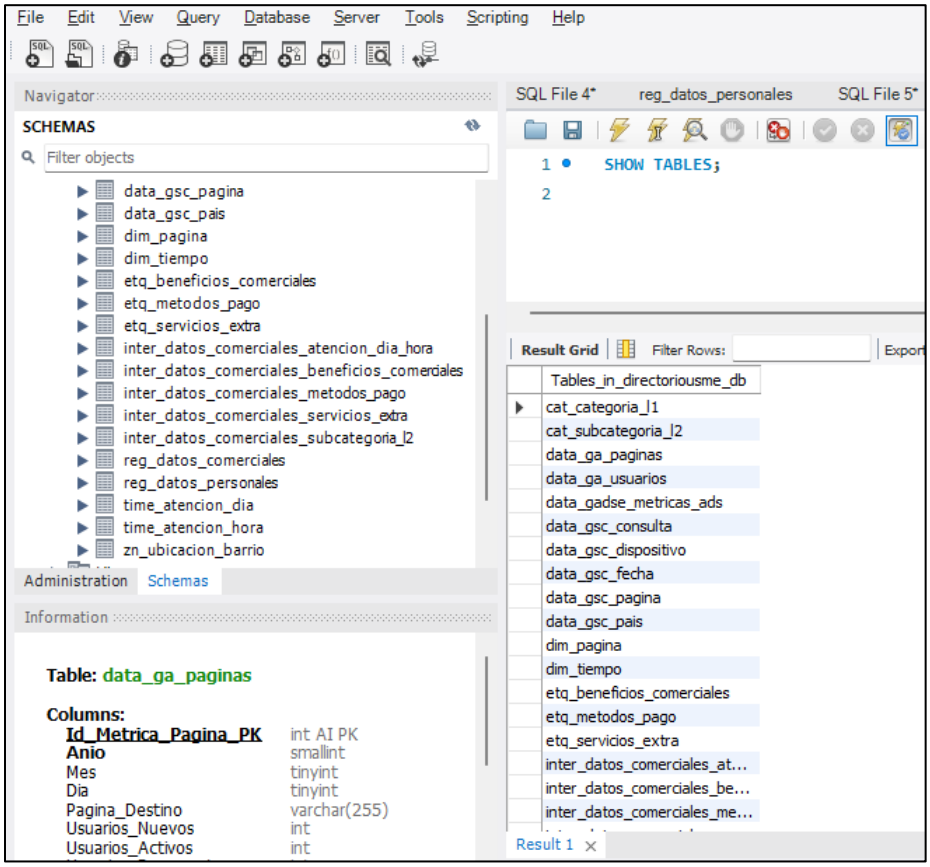
Estas validaciones permiten confirmar que los datos están correctamente estructurados y relacionados, lo cual es fundamental para garantizar su integridad referencial y facilitar su análisis cruzado.

Esto incrementa significativamente el valor analítico de la base de datos, al permitir consultas complejas, generación de reportes consolidados y visualizaciones.

A continuación, se presenta una vista general de todas las tablas creadas dentro del sistema de gestión relacional, como resultado del proceso de modelado implementado en MySQL

Figura 48

Base de Datos del DirectorioUsme.com en MySQL



Nota. Se visualizan todas las tablas mencionadas anteriormente dentro de la base de datos

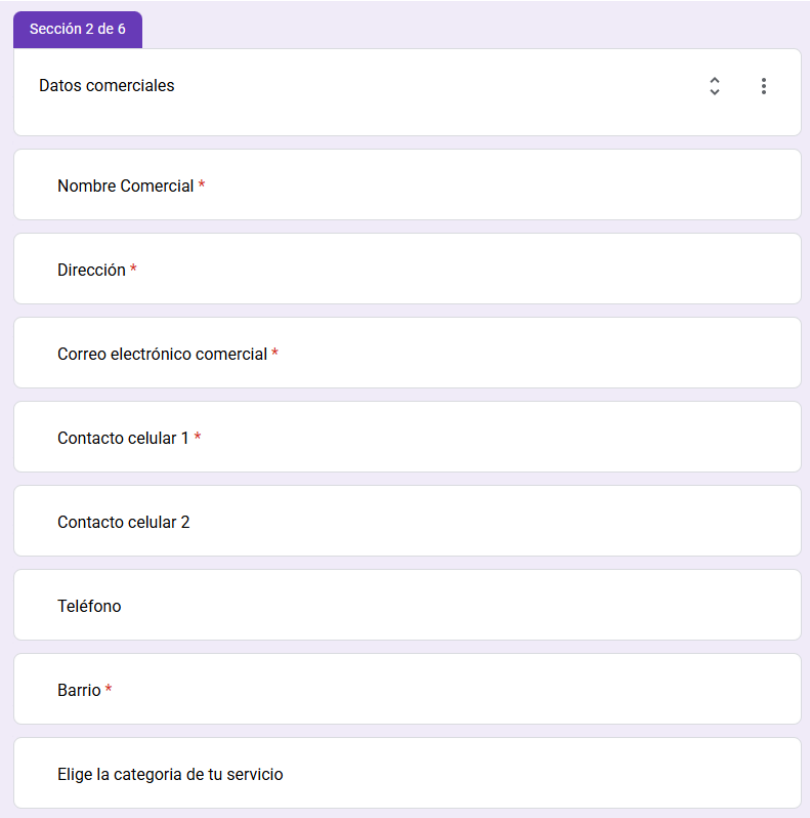
Proceso ETL: Una vez definidas las estructuras relacionales en MySQL, se procedió a diseñar el proceso de carga automatizada de datos mediante un flujo ETL *(Extract, Transform, Load).* Como lo destaca (Encalada Garcia, 2025), la implementación de procesos ETL optimizados, utilizando Python y herramientas como Power BI, permite integrar diversas fuentes de datos y automatizar la consolidación de información en entornos estructurados.

Para este proyecto, se definió la creación de un formulario en Google Forms como mecanismo para permitir que los interesados registren su negocio o servicio de manera práctica y

rápida. Este formulario incluye campos abiertos y listas desplegables que facilitan la selección de opciones, minimizan errores y agilizan el proceso de registro.

Figura 49

Sección 2 de 6 del Formulario Creado para Registro de Negocios Locales



Nota. Elaboración propia. Se visualiza extracto del formulario. URL del formulario

La información recopilada se almacena automáticamente en una hoja de cálculo de Google Sheets, la cual sirve como fuente para el proceso ETL implementado en Python mediante Jupyter Notebook. Este proceso se ejecuta en tres fases principales:

Extracción: El proceso inicia con la importación de todas las librerías necesarias para interactuar con las APIs de Google, manipular los datos, conectarse a Google Sheets, enviar notificaciones por correo electrónico y operar con MySQL.

Posteriormente, se establecen las credenciales de acceso a la hoja de cálculo donde se almacenan las respuestas del formulario, y se extraen los registros utilizando pandas y gspread. Finalmente, se filtran únicamente los registros del día anterior, evitando reprocesamientos.

Figura 50

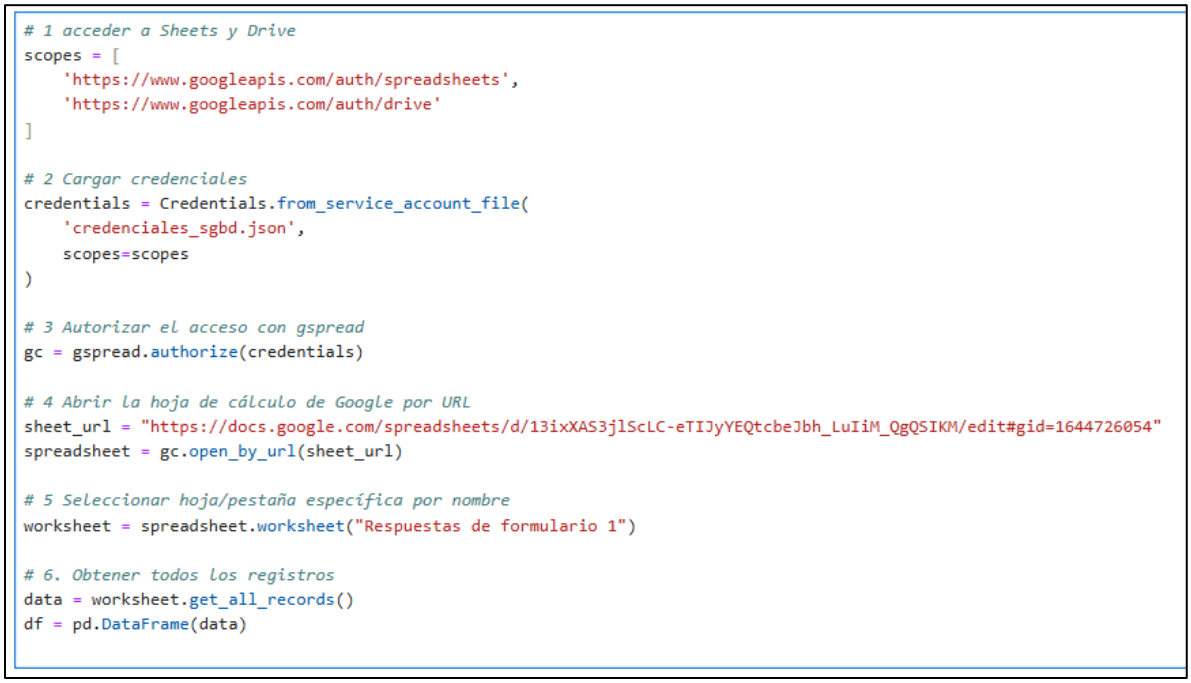
Librerías Necesarias

```python
# 1. librerías necesarias
import pandas as pd
import gspread
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from google.oauth2.service_account import Credentials
from email.message import EmailMessage
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para iniciar la importación de librerías necesarias para el inicio del proceso ETL

Figura 51

Librerías y Autorización de Permisos para Iniciar Extracción



```python
# 1 acceder a Sheets y Drive
scopes = [
    'https://www.googleapis.com/auth/spreadsheets',
    'https://www.googleapis.com/auth/drive'
]

# 2 Cargar credenciales
credentials = Credentials.from_service_account_file(
    'credenciales_sgbd.json',
    scopes=scopes
)

# 3 Autorizar el acceso con gspread
gc = gspread.authorize(credentials)

# 4 Abrir la hoja de cálculo de Google por URL
sheet_url = "https://docs.google.com/spreadsheets/d/13ixXAS3jlScLC-eTIJyYEQtcbeJbh_LuIiM_QgQSIKM/edit#gid=1644726054"
spreadsheet = gc.open_by_url(sheet_url)

# 5 Seleccionar hoja/pestaña específica por nombre
worksheet = spreadsheet.worksheet("Respuestas de formulario 1")

# 6. Obtener todos los registros
data = worksheet.get_all_records()
df = pd.DataFrame(data)
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para autenticarse con Google con credenciales y acceder a la hoja de cálculo del formulario para obtener las respuestas en un *DataFrame*.

Figura 52

Configuración para Enviar Notificaciones al Correo.

```
# Configuración de correo

def enviar_correo(asunto, cuerpo, destino='soydeusme@directoriousme.com '):
    remitente = 'proyectosdigitalesedc@gmail.com'
    password = '[redacted]'


    mensaje = EmailMessage()
    mensaje['Subject'] = asunto
    mensaje['From'] = remitente
    mensaje['To'] = destino
    mensaje.set_content(cuerpo)

    try:
        with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
            smtp.login(remitente, password)
            smtp.send_message(mensaje)
        print("📤 Correo enviado exitosamente.")
    except Exception as e:
        print(f"❌ Error al enviar correo: {e}")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, aquí se configura los parámetros necesarios para enviar por correo notificaciones de los procesos exitosos, fallos o duplicados durante la ejecución *ETL*.

Figura 53

Filtrar Registros del Día Anterior

```
from datetime import datetime, timedelta

# Asegurarte de que la columna tenga formato datetime
df['Marca temporal'] = pd.to_datetime(df['Marca temporal'], format='%d/%m/%Y %H:%M:%S', errors='coerce')

# Filtrar únicamente los registros del día anterior
ayer = (datetime.now() - timedelta(days=1)).date()
df = df[df['Marca temporal'].dt.date == ayer]
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para filtrar únicamente los registros del día anterior para evitar reprocesar datos antiguos

Figura 54

Evitar Duplicados

```
if df.empty:
    print("🔹 No hay registros nuevos para procesar hoy.")
    exit()  # o return si estás dentro de una función
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para finalizar el script si no hay nuevos registros que procesar, ahorrando recursos y evitando errores

Figura 55

Conexión a MySQL.

```
import mysql.connector

conexion = mysql.connector.connect(
    host='localhost',
    user='root',
    password='[redacted]',
    database='directoriousme_db'
)

cursor = conexion.cursor()
print("✅ Conexión exitosa con MySQL.")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para establecer una conexión a la base de datos MySQL donde se almacenarán los datos procesados.

Transformación: En esta etapa, los datos extraídos son limpiados y transformados para ajustarse a las convenciones de la base de datos relacional. Las acciones principales incluyen: Normalización de columnas: se convierten los nombres de columnas a minúsculas, se eliminan tildes y espacios innecesarios, y se ajustan al formato estándar (snake_case) usado en MySQL. Mapeo de nombres: se renombran las columnas del formulario (por ejemplo, “nombre

comercial", "teléfono") a sus equivalentes en la base de datos (nombre_comercial, contacto_telefono), garantizando compatibilidad con las tablas creadas. Depuración de valores sensibles: se eliminan espacios internos en campos como cédula, correo electrónico o celular, y se estandarizan las listas múltiples (como subcategorías, servicios extra, beneficios o métodos de pago). Obtención de llaves foráneas: se consultan los identificadores correspondientes en las tablas de referencia (categoría, barrio, etiquetas) para mantener la integridad relacional.

Figura 56

Limpieza y transformación

```python
# Normalizar nombres de columnas (quita espacios, minúsculas, sin tildes)
import unidecode
df.columns = [unidecode.unidecode(col.strip().lower()) for col in df.columns]

# Eliminar dobles espacios en columnas
for col in df.select_dtypes(include='object').columns:
    df[col] = df[col].apply(lambda x: ' '.join(x.split()) if isinstance(x, str) else x)

# Quitar espacios internos en campos (como cédula, celular, etc.)
campos_sin_espacios = [
    'cedula',
    'correo electronico',
    'contacto celular',
    'contacto telefono',
    'correo electronico comercial',
    'contacto celular 1',
    'contacto celular 2',
    'telefono'
]

for campo in campos_sin_espacios:
    if campo in df.columns:
        df[campo] = df[campo].astype(str).str.replace(r'\s+', '', regex=True)
```

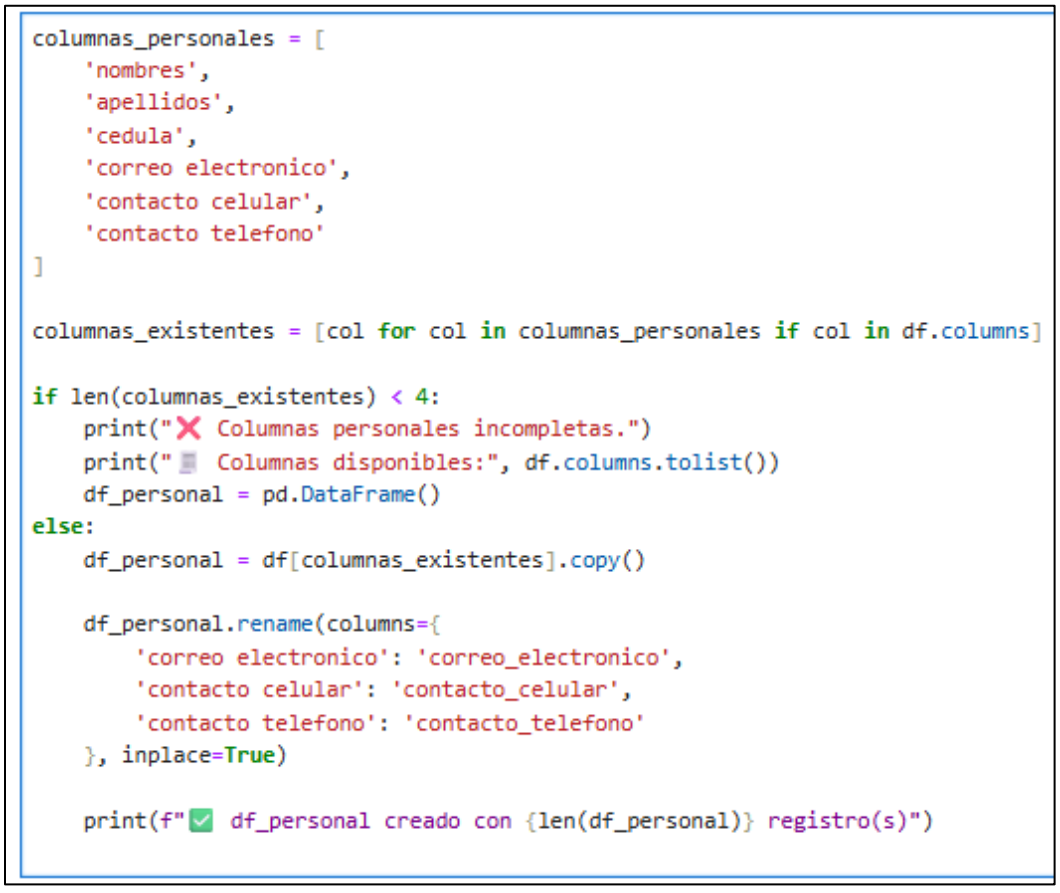
```python
# Mapeo de columnas del formulario a columnas en MySQL
columnas_mapeo = {
    'nombre comercial': 'nombre_comercial',
    'direccion': 'direccion',
    'correo electronico comercial': 'correo_electronico',
    'contacto celular 1': 'contacto_celular_1',
    'contacto celular 2': 'contacto_celular_2',
    'telefono': 'contacto_telefono',
    'barrio': 'nombre_barrio',
    'elige la categoria d tu servicio': 'nombre_categoria',
    'descripcion del servicio': 'descripcion'
}

# Seleccionar columnas disponibles y renómbralas
columnas_presentes = [col for col in columnas_mapeo if col in df.columns]
df_comercial = df[columnas_presentes].copy()
df_comercial.rename(columns={col: columnas_mapeo[col] for col in columnas_presentes}, inplace=True)
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para limpiar y normalizar los datos del formulario (quitar tildes, espacios, estandarizar nombres de columnas) y preparar los datos para insertarlos en la base

Carga: Luego de transformar los datos, se realiza la inserción en las tablas correspondientes. Para cada registro, se verifica si ya existe una combinación duplicada y se evita su inserción en caso necesario. Finalmente, se ejecuta el método commit() sobre la conexión MySQL, lo que asegura que todos los datos transformados se guarden de forma permanente en la base. Se envía un correo con el resumen de registros insertados, duplicados detectados o errores encontrados durante la ejecución.

Figura 57

Seleccionar Datos Personales

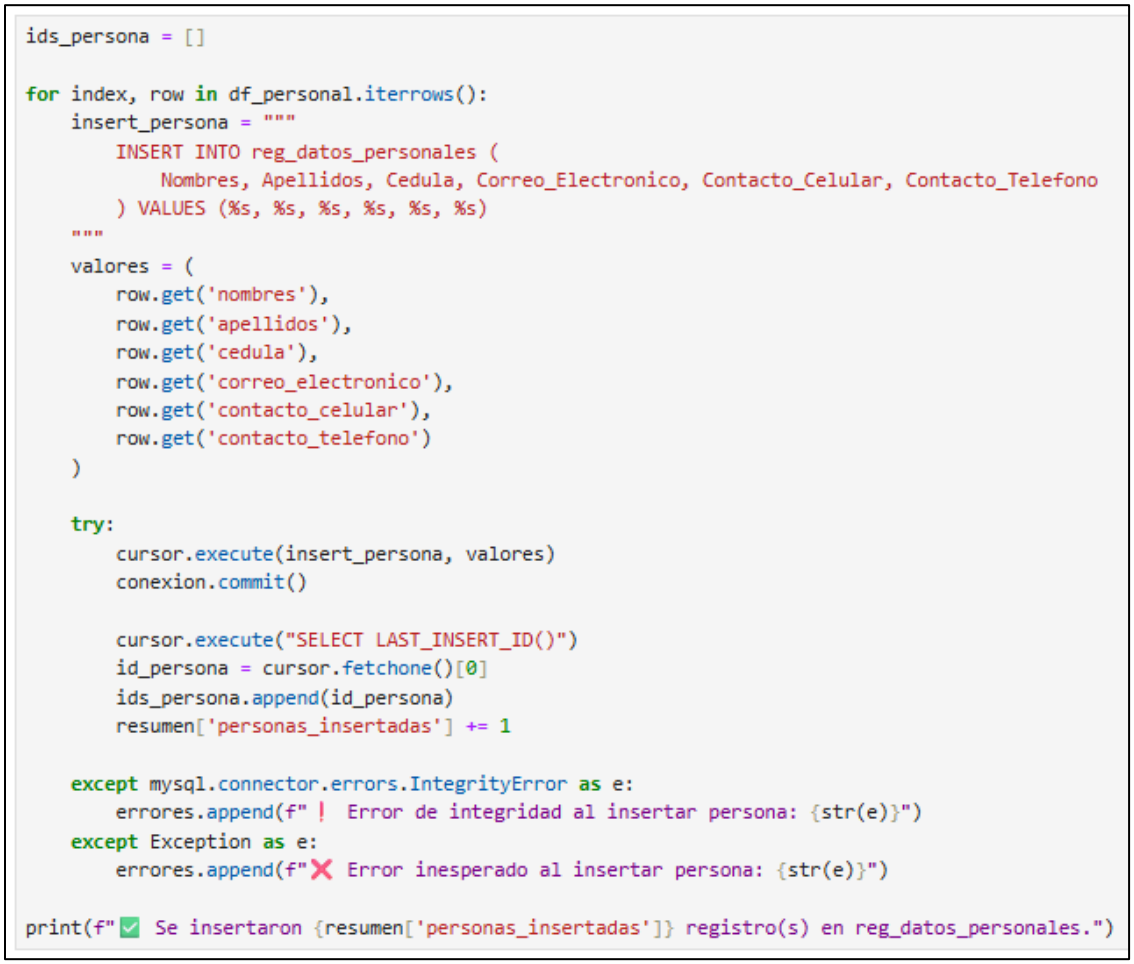


```
columnas_personales = [
    'nombres',
    'apellidos',
    'cedula',
    'correo electronico',
    'contacto celular',
    'contacto telefono'
]

columnas_existentes = [col for col in columnas_personales if col in df.columns]

if len(columnas_existentes) < 4:
    print("❌ Columnas personales incompletas.")
    print("📋 Columnas disponibles:", df.columns.tolist())
    df_personal = pd.DataFrame()
else:
    df_personal = df[columnas_existentes].copy()

    df_personal.rename(columns={
        'correo electronico': 'correo_electronico',
        'contacto celular': 'contacto_celular',
        'contacto telefono': 'contacto_telefono'
    }, inplace=True)

    print(f"✅ df_personal creado con {len(df_personal)} registro(s)")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para seleccionar únicamente los datos personales para guardarlos en un dataframe generando un ID único para cada persona.

Figura 58

Insertar Datos Personales en la Base de Dato en Mysql



```
ids_persona = []

for index, row in df_personal.iterrows():
    insert_persona = """
        INSERT INTO reg_datos_personales (
            Nombres, Apellidos, Cedula, Correo_Electronico, Contacto_Celular, Contacto_Telefono
        ) VALUES (%s, %s, %s, %s, %s, %s)
    """
    valores = (
        row.get('nombres'),
        row.get('apellidos'),
        row.get('cedula'),
        row.get('correo_electronico'),
        row.get('contacto_celular'),
        row.get('contacto_telefono')
    )

    try:
        cursor.execute(insert_persona, valores)
        conexion.commit()

        cursor.execute("SELECT LAST_INSERT_ID()")
        id_persona = cursor.fetchone()[0]
        ids_persona.append(id_persona)
        resumen['personas_insertadas'] += 1

    except mysql.connector.errors.IntegrityError as e:
        errores.append(f" ! Error de integridad al insertar persona: {str(e)}")
    except Exception as e:
        errores.append(f"❌ Error inesperado al insertar persona: {str(e)}")

print(f"✅ Se insertaron {resumen['personas_insertadas']} registro(s) en reg_datos_personales.")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para realizar la conexión y la posterior carga de los datos personales en la tabla Reg_Datos_Comerciales

Figura 59

Insertar Datos Comerciales en la Base de Dato En Mysql

```
ids_comercial = []

for i in range(len(df_comercial)):
    row = df_comercial.iloc[i]
    id_persona_fk = ids_persona[i]

    id_barrio = obtener_id(cursor, 'Zn_Ubicacion_Barrio', 'Id_Barrio_PK', 'Nombre_Barrio', row.get('nombre_barrio'))
    id_categoria = obtener_id(cursor, 'Cat_Categoria_L1', 'Id_Categoria_L1', 'Nombre_Categoria_L1', row.get('nombre_categoria'))

    insert_comercial = """
        INSERT INTO Reg_Datos_Comerciales (
            Id_Persona_FK, Nombre_Comercial, Estado, Direccion, Correo_Electronico,
            Contacto_Celular_1, Contacto_Celular_2, Contacto_Telefono,
            Id_Barrio_FK, Id_Categoria_L1_FK, Descripcion, Observacion
        ) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    """
    valores_comercial = (
        id_persona_fk,
        row.get('nombre_comercial'),
        1,
        row.get('direccion'),
        row.get('correo_electronico'),
        row.get('contacto_celular_1'),
        row.get('contacto_celular_2'),
        row.get('contacto_telefono'),
        id_barrio,
        id_categoria,
        row.get('descripcion'),
        None
    )

    try:
        cursor.execute(insert_comercial, valores_comercial)
        conexion.commit()

        cursor.execute("SELECT LAST_INSERT_ID()")
        id_comercial = cursor.fetchone()[0]
        ids_comercial.append(id_comercial)
        resumen['comercios_insertados'] += 1

    except Exception as e:
        errores.append(f"❌ Error al insertar comercio (fila {i+1}): {str(e)}")

print(f"✅ Se insertaron {len(ids_comercial)} registro(s) en Reg_Datos_Comerciales.")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, guarda la información del comercio en la tabla Reg_Datos_Comerciales, relacionándola con la persona y otros datos como barrio y categoría

Figura 60

Asociar Comercio con Subcategoría en la Base de Datos en MySQL.

```python
if 'subcategoría' in df.columns:
    df['subcategoría'] = df['subcategoría'].astype(str).str.strip()

df['subcategoría'] = df[[col for col in df.columns if 'subcategorias' in col]].fillna('').agg(','.join, axis=1)
df['subcategoría'] = df['subcategoría'].str.replace(r',+', ',', regex=True).str.strip(', ')


for i in range(len(df)):
    id_comercial = ids_comercial[i]
    subcategorias = df.loc[i, 'subcategoría']

    if pd.notna(subcategorias):
        lista_subcats = [sub.strip() for sub in subcategorias.split(',')]

        for subcat in lista_subcats:
            try:
                id_sub = obtener_id(cursor, 'Cat_Subcategoria_L2', 'Id_Subcategoria_L2', 'Nombre_Subcategoria_L2', subcat)
                if id_sub:
                    cursor.execute("""
                        SELECT 1 FROM inter_datos_comerciales_subcategoria_l2
                        WHERE Id_Comercial_FK = %s AND Id_Subcategoria_L2_FK = %s
                    """, (id_comercial, id_sub))
                    if cursor.fetchone() is None:
                        cursor.execute("""
                            INSERT INTO inter_datos_comerciales_subcategoria_l2
                            (Id_Comercial_FK, Id_Subcategoria_L2_FK, Estado)
                            VALUES (%s, %s, %s)
                        """, (id_comercial, id_sub, 1))
                        conexion.commit()
                        resumen['subcategorias_insertadas'] += 1
                else:
                    errores.append(f"❌ Subcategoría no encontrada: {subcat}")
            except Exception as e:
                errores.append(f"❌ Error en subcategoría '{subcat}' para comercial {id_comercial}: {str(e)}")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para asociar cada comercio con una o varias subcategorías, registrándolas en la tabla intermedia inter_datos_comerciales_subcategoria_l2

Figura 61

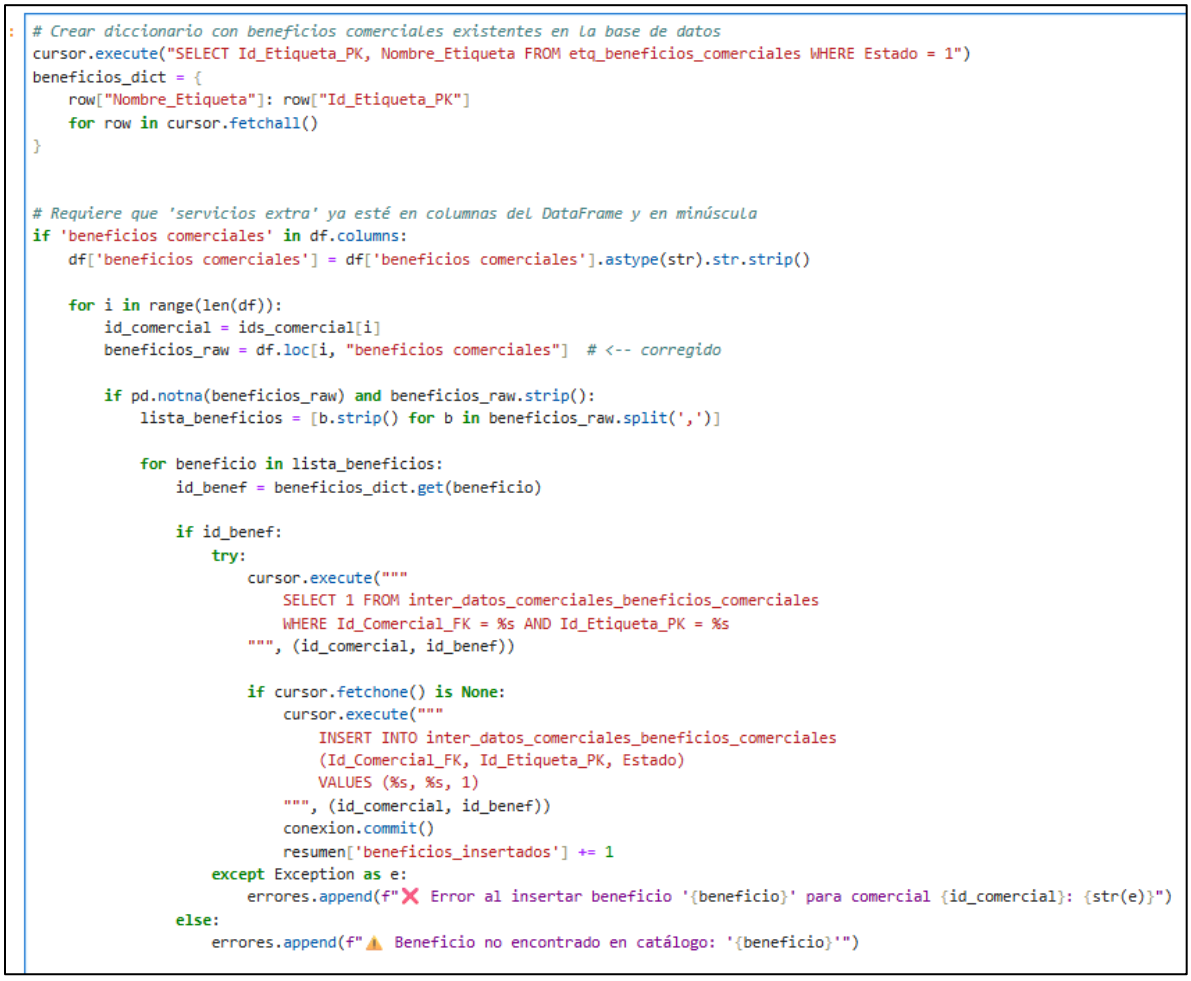
Asociar Comercio con Servicios Extra en la Base de Dato en Mysql

```
cursor = conexion.cursor(dictionary=True)
# Cargar servicios extra existentes desde la tabla
cursor.execute("SELECT Id_Etiqueta_PK, Nombre_Etiqueta FROM etq_servicios_extra WHERE Estado = 1")
servicios_extra_dict = {
    unidecode.unidecode(row["Nombre_Etiqueta"]).strip().title(): row["Id_Etiqueta_PK"]
    for row in cursor.fetchall()
}
# Procesar los servicios extra para cada comercio
for i in range(len(df)):
    id_comercial = ids_comercial[i]
    servicios_raw = df.loc[i, "servicios extra"]

    if pd.notna(servicios_raw) and servicios_raw.strip():
        servicios_lista = [s.strip() for s in servicios_raw.split(",")]

        for servicio in servicios_lista:
            servicio_normalizado = servicio.strip().title()
            id_servicio = servicios_extra_dict.get(servicio_normalizado)

            if id_servicio:
                try:
                    cursor.execute("""
                        SELECT 1 FROM inter_datos_comerciales_servicios_extra
                        WHERE Id_Comercial_FK = %s AND Id_Etiqueta_PK = %s
                    """, (id_comercial, id_servicio))

                    if cursor.fetchone() is None:
                        cursor.execute("""
                            INSERT INTO inter_datos_comerciales_servicios_extra
                            (Id_Comercial_FK, Id_Etiqueta_PK, Estado)
                            VALUES (%s, %s, 1)
                        """, (id_comercial, id_servicio))
                        resumen["servicios_extra_insertados"] += 1
                except Exception as e:
                    errores.append(f"❌ Error al insertar servicio extra '{servicio}' para comercial {id_comercial}: {str(e)}")
            else:
                errores.append(f"⚠️ Servicio extra no encontrado en catálogo: '{servicio}'")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para relacionar los comercios con los servicios adicionales que ofrecen, usando la tabla Inter_Datos_Comerciales_Servicios_Extra

Figura 62

Asociar Comercio con Beneficios Comerciales en la Base de Dato en Mysql.



```
# Crear diccionario con beneficios comerciales existentes en la base de datos
cursor.execute("SELECT Id_Etiqueta_PK, Nombre_Etiqueta FROM etq_beneficios_comerciales WHERE Estado = 1")
beneficios_dict = {
    row["Nombre_Etiqueta"]: row["Id_Etiqueta_PK"]
    for row in cursor.fetchall()
}


# Requiere que 'servicios extra' ya esté en columnas del DataFrame y en minúscula
if 'beneficios comerciales' in df.columns:
    df['beneficios comerciales'] = df['beneficios comerciales'].astype(str).str.strip()

    for i in range(len(df)):
        id_comercial = ids_comercial[i]
        beneficios_raw = df.loc[i, "beneficios comerciales"]  # <-- corregido

        if pd.notna(beneficios_raw) and beneficios_raw.strip():
            lista_beneficios = [b.strip() for b in beneficios_raw.split(',')]

            for beneficio in lista_beneficios:
                id_benef = beneficios_dict.get(beneficio)

                if id_benef:
                    try:
                        cursor.execute("""
                            SELECT 1 FROM inter_datos_comerciales_beneficios_comerciales
                            WHERE Id_Comercial_FK = %s AND Id_Etiqueta_PK = %s
                        """, (id_comercial, id_benef))

                        if cursor.fetchone() is None:
                            cursor.execute("""
                                INSERT INTO inter_datos_comerciales_beneficios_comerciales
                                (Id_Comercial_FK, Id_Etiqueta_PK, Estado)
                                VALUES (%s, %s, 1)
                            """, (id_comercial, id_benef))
                            conexion.commit()
                            resumen['beneficios_insertados'] += 1
                    except Exception as e:
                        errores.append(f"❌ Error al insertar beneficio '{beneficio}' para comercial {id_comercial}: {str(e)}")
                else:
                    errores.append(f"⚠️ Beneficio no encontrado en catálogo: '{beneficio}'")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para registrar los beneficios que ofrece cada comercio en la tabla Inter_Datos_Comerciales_Beneficios_Comerciales

Figura 63

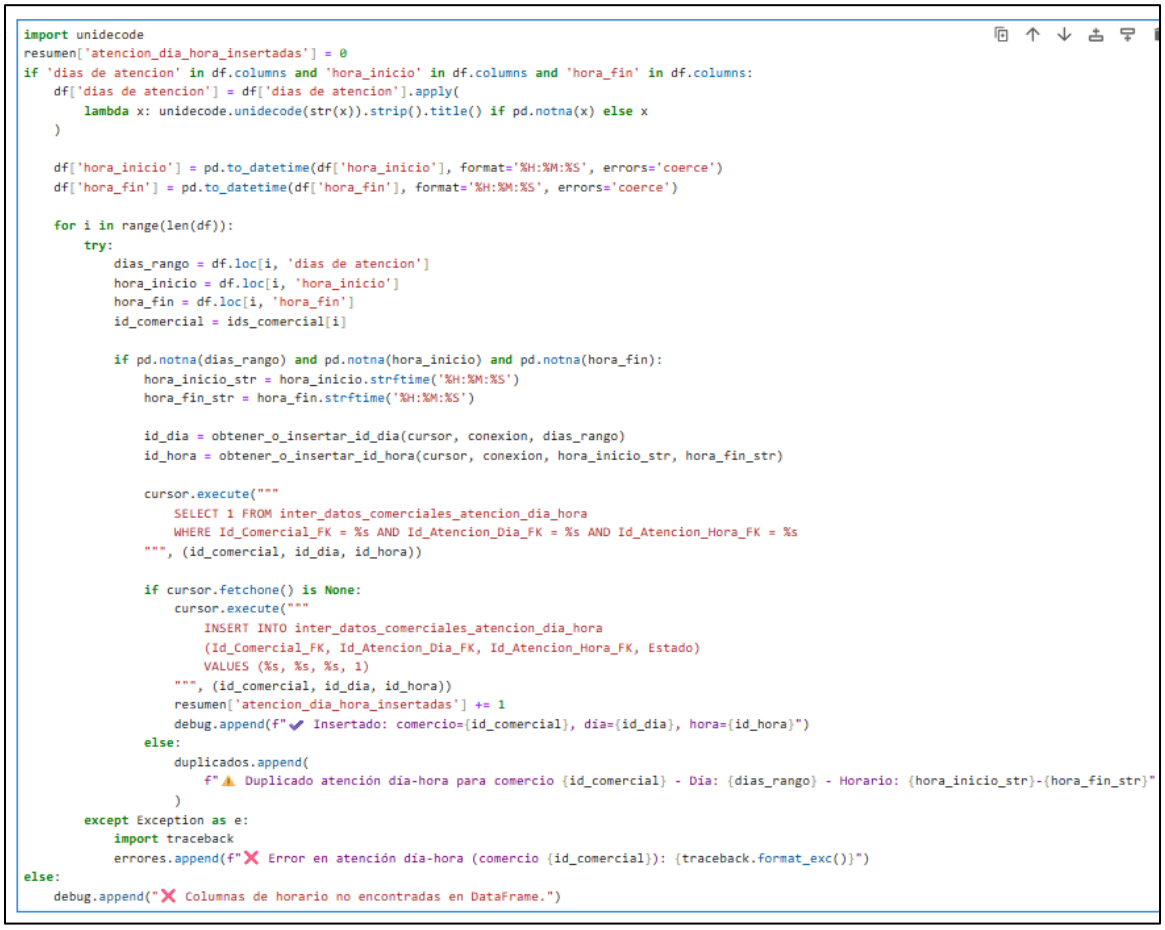
Asociar Comercio con Métodos de Pago en la Base de Dato en Mysql

```
# Normalizar los nombres de columnas
df.columns = df.columns.str.strip().str.lower()

# Cargar métodos de pago válidos desde la tabla etq_metodos_pago
cursor.execute("SELECT Id_Etiqueta_PK, Nombre_Etiqueta FROM etq_metodos_pago WHERE Estado = 1")
metodos_pago_dict = {
    row["Nombre_Etiqueta"]: row["Id_Etiqueta_PK"]
    for row in cursor.fetchall()
}

if 'metodos de pago' in df.columns:
    df['metodos de pago'] = df['metodos de pago'].astype(str).str.strip()

    for i in range(len(df)):
        id_comercial = ids_comercial[i]
        metodos_raw = df.loc[i, "metodos de pago"]

        if pd.notna(metodos_raw) and metodos_raw.strip():
            metodos_lista = [m.strip() for m in metodos_raw.split(",")]

            for metodo in metodos_lista:
                id_metodo = metodos_pago_dict.get(metodo)

                if id_metodo:
                    try:
                        cursor.execute("""
                            INSERT INTO inter_datos_comerciales_metodos_pago
                            (Id_Comercial_FK, Id_Etiqueta_PK, Estado)
                            VALUES (%s, %s, %s)
                        """, (id_comercial, id_metodo, 1))
                        resumen["metodos_pago_insertados"] += 1
                    except Exception as e:
                        errores.append(
                            f"❌ Error al insertar método de pago '{metodo}' para comercial {id_comercial}: {str(e)}"
                        )
                else:
                    errores.append(
                        f"⚠️ Método de pago no encontrado en catálogo: '{metodo}'"
                    )
else:
    errores.append("❌ La columna 'métodos de pago' no está disponible en el DataFrame.")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para registrar los beneficios que ofrece cada comercio en la tabla Inter_Datos_Comerciales_Beneficios_Comerciales

Figura 64

Registrar Horario de Atención de cada Comercio en la Base de Dato en Mysql



```
import unidecode
resumen['atencion_dia_hora_insertadas'] = 0
if 'dias de atencion' in df.columns and 'hora_inicio' in df.columns and 'hora_fin' in df.columns:
    df['dias de atencion'] = df['dias de atencion'].apply(
        lambda x: unidecode.unidecode(str(x)).strip().title() if pd.notna(x) else x
    )

    df['hora_inicio'] = pd.to_datetime(df['hora_inicio'], format='%H:%M:%S', errors='coerce')
    df['hora_fin'] = pd.to_datetime(df['hora_fin'], format='%H:%M:%S', errors='coerce')

    for i in range(len(df)):
        try:
            dias_rango = df.loc[i, 'dias de atencion']
            hora_inicio = df.loc[i, 'hora_inicio']
            hora_fin = df.loc[i, 'hora_fin']
            id_comercial = ids_comercial[i]

            if pd.notna(dias_rango) and pd.notna(hora_inicio) and pd.notna(hora_fin):
                hora_inicio_str = hora_inicio.strftime('%H:%M:%S')
                hora_fin_str = hora_fin.strftime('%H:%M:%S')

                id_dia = obtener_o_insertar_id_dia(cursor, conexion, dias_rango)
                id_hora = obtener_o_insertar_id_hora(cursor, conexion, hora_inicio_str, hora_fin_str)

                cursor.execute("""
                    SELECT 1 FROM inter_datos_comerciales_atencion_dia_hora
                    WHERE Id_Comercial_FK = %s AND Id_Atencion_Dia_FK = %s AND Id_Atencion_Hora_FK = %s
                """, (id_comercial, id_dia, id_hora))

                if cursor.fetchone() is None:
                    cursor.execute("""
                        INSERT INTO inter_datos_comerciales_atencion_dia_hora
                        (Id_Comercial_FK, Id_Atencion_Dia_FK, Id_Atencion_Hora_FK, Estado)
                        VALUES (%s, %s, %s, 1)
                    """, (id_comercial, id_dia, id_hora))
                    resumen['atencion_dia_hora_insertadas'] += 1
                    debug.append(f"✔ Insertado: comercio={id_comercial}, dia={id_dia}, hora={id_hora}")
                else:
                    duplicados.append(
                        f"⚠ Duplicado atención dia-hora para comercio {id_comercial} - Dia: {dias_rango} - Horario: {hora_inicio_str}-{hora_fin_str}"
                    )
        except Exception as e:
            import traceback
            errores.append(f"❌ Error en atención dia-hora (comercio {id_comercial}): {traceback.format_exc()}")
else:
    debug.append("❌ Columnas de horario no encontradas en DataFrame.")
```

Nota. Código utilizado en la herramienta Jupyter haciendo uso del lenguaje de Python, para registrar los días y horarios en los que cada comercio atiende, usando las tablas Time_Atencion_Dia, Time_Atencion_Hora y la tabla intermedia Inter_Datos_Comerciales_Atencion_Dia_Hora

Selección de Datos de Prueba

Con el objetivo de validar la estructura, funcionamiento e integración del modelo relacional con el flujo de carga y visualización de datos, se realizó una prueba controlada con registros simulados. Esta prueba permitió simular el comportamiento completo del sistema,

desde la recolección inicial de datos hasta su análisis visual en Power BI, pasando por el proceso ETL y la inserción en MySQL.

Paso 1 Diligenciar el Formulario

Figura 65

Parte 1 del Formulario Diligenciado con Registros de Prueba

Form_Responses1

Marca temporal	Nombres	Apellidos	Cédula	Correo electrónico	Contacto Celular	Contactc	Nombre Comercial	Dirección	Correo electronico c
28/6/2025 1:02:21	Laura	Rodríguez	1023567890	laura.rodriguez@email.c	3102345678		Tienda Natural Verde	Calle 10 #5-33	contacto@naturalverde
28/6/2025 1:02:21	Juan	Martínez	1032456789	juan.martinez@email.co	3123456789		Panadería San Juan	Carrera 3 #12-45	ventas@pansanjuan.co
28/6/2025 1:02:21	Catalina	Gómez	1041234567	catalina.gomez@email.(	3119876543		Café Aroma de Casa	Calle 15 Sur #4-88	cafe@aromacasa.com
28/6/2025 1:02:21	Sebastián	Ríos	1012345678	sebastian.rios@email.co	3134567890		Taller Ríos Motor	Transversal 7 #23-10	taller@riosmotor.com
28/6/2025 1:02:21	Diana	Hernández	1029876543	diana.hernandez@email	3142223344		Estética Divine Look	Avenida Caracas #89-20	info@divinelook.com
28/6/2025 1:02:21	Andrés	López	1001122334	andres.lopez@email.cor	3151234567		Droguería Vida Plena	Calle 30B Sur #2-01	contacto@vidaplena.co
28/6/2025 1:02:21	Valentina	Torres	1054321987	valentina.torres@email.(	3161112233		Boutique Luna Rosa	Carrera 12 Este #15-44	ventas@lunarosa.com
28/6/2025 1:02:21	Diego	Ramírez	1076543210	diego.ramirez@email.co	3179871234		Papelería Escolar Pro	Calle 45A #9-76	info@escolarpro.com
28/6/2025 1:02:21	Carolina	Ruiz	1067894321	carolina.ruiz@email.con	3186543210		Pastelería Dulce Arte	Carrera 18 #50-30	dulcearte@pasteles.co
28/6/2025 1:02:21	Nicolás	Vargas	1087654321	nicolas.vargas@email.c	3191122334		Barbería El Corte	Av. Villavicencio #25-55	corte@barberiaelcorte.

Nota. Captura de pantalla de la hoja de cálculo de Google Sheets que contiene datos personales que debe diligenciar una persona interesada al registrar un negocio o servicio

Figura 66

Parte 2 del Formulario Diligenciado con Registros de Prueba

Barrio	Elige la categoria de	Descripción del ser	Servicios extra	Métodos de pago	Beneficios comercia	Comidas_Rapidas -	Regalos_Desayunos	Belleza_Bienestar -	Días de atención	Hora_inicio	Hora_fin
Alfonso Lopez	Comidas_Rapidas	Bienvenidos somos los	Recarga Minutos, Recar	Pago con Tarjeta, Pago	Cita previa, Contraentrej	Hamburguesas_Perros_			Lunes a Sábado	10:00:00	23:00:00
Alfonso Lopez	Regalos_Desayunos_Flo	Bienvenidos somos los	Recarga Tu Llave, Movi	Pago con Tarjeta, Pago	Cita previa, Contraentrej			Estilistas_Independiente	Lunes - Viernes	9:30:00	16:30:00
Usminia	Comidas_Rapidas	Bienvenidos somos los	Recarga Minutos, Recar	Pago con Tarjeta, Pago	Cita previa, Contraentrej		Productos_Personalizad		Lunes a Sábado	10:00:00	23:00:00
Usminia	Regalos_Desayunos_Flo	Bienvenidos somos los	Recarga Tu Llave, Movi	Pago con Tarjeta, Pago	Cita previa, Contraentrej	Pizzeria			Lunes - Viernes	9:30:00	16:30:00
Alfonso Lopez	Comidas_Rapidas	Bienvenidos somos los	Recarga Minutos, Recar	Pago con Tarjeta, Pago	Cita previa, Contraentrej			Centros_Estetica_Spa	Lunes a Sábado	10:00:00	23:00:00
Alfonso Lopez	Regalos_Desayunos_Flo	Bienvenidos somos los	Recarga Tu Llave, Movi	Pago con Tarjeta, Pago	Cita previa, Contraentrej		Floristería_Plantas_Hierl		Lunes - Viernes	9:30:00	16:30:00
Alfonso Lopez	Comidas_Rapidas	Bienvenidos somos los	Recarga Minutos, Recar	Pago con Tarjeta, Pago	Cita previa, Contraentrej	Mazorcada, Pizzeria			Lunes a Sábado	10:00:00	23:00:00
Alfonso Lopez	Regalos_Desayunos_Flo	Bienvenidos somos los	Recarga Tu Llave, Movi	Pago con Tarjeta, Pago	Cita previa, Contraentrej			Tiendas_Naturistas	Lunes - Viernes	9:30:00	16:30:00
Alfonso Lopez	Comidas_Rapidas	Bienvenidos somos los	Recarga Minutos, Recar	Pago con Tarjeta, Pago	Cita previa, Contraentrej		Tiendas_Detalles_Pelucl		Lunes a Sábado	10:00:00	23:00:00
Alfonso Lopez	Regalos_Desayunos_Flo	Bienvenidos somos los	Recarga Tu Llave, Movi	Pago con Tarjeta, Pago	Cita previa, Contraentrej	Mazorcada, Arepas_Emp			Lunes - Viernes	9:30:00	16:30:00

Nota. Captura de pantalla de la hoja de cálculo de Google Sheets que contiene datos comerciales que debe compartir una persona interesada al registrar un negocio o servicio

Paso2 - Extracción de Datos desde Google Forms**:** Se establece conexión con el archivo de Google Sheets generado automáticamente por el formulario, el cual es cargado en un DataFrame de Pandas desde el entorno Jupyter Notebook.

Transformación de los datos: Durante esta etapa, se realiza una limpieza inicial de los textos (conversión a minúsculas, eliminación de espacios innecesarios) y se renombran las columnas para que coincidan con la estructura definida en el modelo relacional de MySQL. Posteriormente, la información es segmentada en diferentes partes: datos personales, datos comerciales y tablas intermedias (como beneficios, métodos de pago, subcategorías y horarios). **Carga en MySQL:** Se realizó la inserción de los registros en la tabla Reg_Datos_Personales, confirmando su éxito mediante la salida impresa del script (véase Figura 66). De igual forma, se efectuó la carga de los datos en la tabla Reg_Datos_Comerciales (véase Figura 67) y en las correspondientes tablas intermedias.

Para asegurar la persistencia de los datos, se utilizó el método conexion.commit(), confirmando así la ejecución exitosa en cada entidad.

A su vez, se validó el envío del correo electrónico con el resumen de novedades sobre la ejecución del script (Figura 68).

Finalmente, se cerró la conexión con la base de datos y se verificó la recepción del correo en la bandeja correspondiente (Figuras 69 y 70), confirmando la correcta finalización del proceso.

Figura 67

10 Registros Insertados en Reg_Datos_Personales

```
ids_persona = []

for index, row in df_personal.iterrows():
    insert_persona = """
        INSERT INTO reg_datos_personales (
            Nombres, Apellidos, Cedula, Correo_Electronico, Contacto_Celular, Contacto_Telefono
        ) VALUES (%s, %s, %s, %s, %s, %s)
    """
    valores = (
        row.get('nombres'),
        row.get('apellidos'),
        row.get('cedula'),
        row.get('correo_electronico'),
        row.get('contacto_celular'),
        row.get('contacto_telefono')
    )

    try:
        cursor.execute(insert_persona, valores)
        conexion.commit()

        cursor.execute("SELECT LAST_INSERT_ID()")
        id_persona = cursor.fetchone()[0]
        ids_persona.append(id_persona)
        resumen['personas_insertadas'] += 1

    except mysql.connector.errors.IntegrityError as e:
        errores.append(f" ❗ Error de integridad al insertar persona: {str(e)}")
    except Exception as e:
        errores.append(f"❌ Error inesperado al insertar persona: {str(e)}")

print(f"✅ Se insertaron {resumen['personas_insertadas']} registro(s) en reg_datos_personales.")
```

```
✅ Se insertaron 10 registro(s) en reg_datos_personales.
```

Nota. Captura de pantalla con la salida de confirmación que muestra la inserción exitosa de los registros encontrados en la hoja de Google Sheets, los cuales correspondían efectivamente a un total de 10 registros

Figura 68

10 Registros Insertados en Reg_Datos_Comerciales

```
ids_comercial = []

for i in range(len(df_comercial)):
    row = df_comercial.iloc[i]
    id_persona_fk = ids_persona[i]

    id_barrio = obtener_id(cursor, 'Zn_Ubicacion_Barrio', 'Id_Barrio_PK', 'Nombre_Barrio', row.get('nombre_barrio'))
    id_categoria = obtener_id(cursor, 'Cat_Categoria_L1', 'Id_Categoria_L1', 'Nombre_Categoria_L1', row.get('nombre_categoria'))

    insert_comercial = """
        INSERT INTO Reg_Datos_Comerciales (
            Id_Persona_FK, Nombre_Comercial, Estado, Direccion, Correo_Electronico,
            Contacto_Celular_1, Contacto_Celular_2, Contacto_Telefono,
            Id_Barrio_FK, Id_Categoria_L1_FK, Descripcion, Observacion
        ) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    """
    valores_comercial = (
        id_persona_fk,
        row.get('nombre_comercial'),
        1,
        row.get('direccion'),
        row.get('correo_electronico'),
        row.get('contacto_celular_1'),
        row.get('contacto_celular_2'),
        row.get('contacto_telefono'),
        id_barrio,
        id_categoria,
        row.get('descripcion'),
        None
    )

    try:
        cursor.execute(insert_comercial, valores_comercial)
        conexion.commit()

        cursor.execute("SELECT LAST_INSERT_ID()")
        id_comercial = cursor.fetchone()[0]
        ids_comercial.append(id_comercial)
        resumen['comercios_insertados'] += 1

    except Exception as e:
        errores.append(f"❌ Error al insertar comercio (fila {i+1}): {str(e)}")

print(f"✅ Se insertaron {len(ids_comercial)} registro(s) en Reg_Datos_Comerciales.")
```

```
✅ Se insertaron 10 registro(s) en Reg_Datos_Comerciales.
```

Nota. Captura de pantalla con la salida de confirmación con la inserción exitosa de datos comerciales encontrados en la hoja de Google Sheets, los cuales correspondían efectivamente a un total de 10 registros ingresados

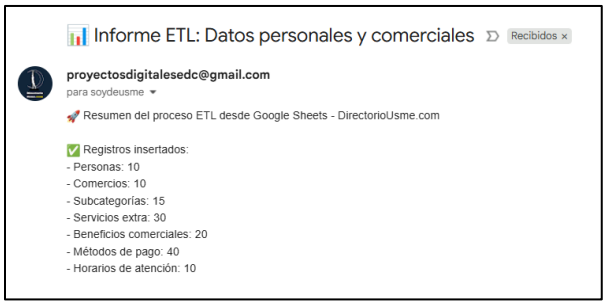
Figura 69

Confirmación de Cierre del Proceso

Nota. Captura de pantalla con la confirmación que el proceso se ejecutó y se envía notificación al correo, luego se cierra la conexión completando el proceso de ETL

Figura 70

Correo Recibido con el Resumen del Proceso ETL



Nota. Captura de pantalla del correo recibido al finalizar exitosamente el proceso ETL. En dicho mensaje se notifican posibles fallas, registros duplicados o, en su defecto, la confirmación de una ejecución sin novedades

Consulta de prueba: Como etapa final del proceso de modelado e integración de datos, se realizó una consulta de validación en el entorno de MySQL con el fin de verificar la correcta implementación del modelo relacional. Esta validación resulta fundamental para comprobar que las relaciones entre las entidades, las claves foráneas y la estructura general del sistema permiten recuperar información de manera coherente, completa y útil para su análisis posterior.

En esta prueba, se diseñó una consulta que integra múltiples tablas del sistema —incluyendo datos personales, datos comerciales, ubicación geográfica, categorías y horarios— con el objetivo de construir una vista consolidada por comercio. Esta vista busca representar los elementos más relevantes de cada registro en una sola fila, facilitando su interpretación y análisis.

Cada resultado muestra información del comercio, tales como el nombre del negocio, nombres y apellidos del representante, barrio asociado, categoría principal (Categoría_L1), subcategoría (Subcategoría_L2) y los días y horarios de atención consolidados en una única línea. Para lograr esta integración se emplearon múltiples sentencias LEFT JOIN, permitiendo conservar registros, aunque no cuenten con toda la información.

Este tipo de consulta no solo valida la estructura del modelo, sino que demuestra su capacidad para responder a necesidades analíticas y operativas, consolidando datos distribuidos de forma lógica y útil.

A continuación, se presentan la consulta realizada (véase Figura 70) y su tabla de respuesta (véase Figura 71), donde es posible observar que algunos registros se repiten debido a la relación de uno a muchos entre comercios y subcategorías.

Figura 71

Consulta de Prueba en Mysql para la Integración de Varias Tablas

```sql
SELECT
  ROW_NUMBER() OVER (ORDER BY c.Id_Comercial_PK) AS number,
  p.Nombres,
  p.Apellidos,
  c.Id_Comercial_PK AS id_comercial,
  c.Nombre_Comercial,
  b.Nombre_Barrio AS barrio,
  catL1.Nombre_Categoria_L1 AS nombre_categoria_l1,
  catL2.Nombre_Subcategoria_L2 AS nombre_subcategoria,
  GROUP_CONCAT(DISTINCT d.Dias_Rango ORDER BY d.Dias_Rango SEPARATOR ', ') AS dias_atencion,
  GROUP_CONCAT(DISTINCT CONCAT(h.Hora_Inicio, ' - ', h.Hora_Fin) ORDER BY h.Hora_Inicio SEPARATOR ', ') AS horario_atencion
FROM reg_datos_comerciales c
LEFT JOIN reg_datos_personales p ON p.Id_Persona_PK = c.Id_Persona_FK
LEFT JOIN zn_ubicacion_barrio b ON b.Id_Barrio_PK = c.Id_Barrio_FK
LEFT JOIN cat_categoria_l1 catL1 ON catL1.Id_Categoria_L1 = c.Id_Categoria_L1_FK
LEFT JOIN inter_datos_comerciales_subcategoria_l2 interSub ON interSub.Id_Comercial_FK = c.Id_Comercial_PK
LEFT JOIN cat_subcategoria_l2 catL2 ON catL2.Id_Subcategoria_L2 = interSub.Id_Subcategoria_L2_FK
LEFT JOIN inter_datos_comerciales_atencion_dia_hora interHorario ON interHorario.Id_Comercial_FK = c.Id_Comercial_PK
LEFT JOIN time_atencion_dia d ON d.Id_Atencion_Dia_PK = interHorario.Id_Atencion_Dia_FK
LEFT JOIN time_atencion_hora h ON h.Id_Atencion_Hora_PK = interHorario.Id_Atencion_Hora_FK
WHERE c.Estado = 1
GROUP BY
  p.Id_Persona_PK,
  p.Nombres,
  p.Apellidos,
  c.Id_Comercial_PK,
  c.Nombre_Comercial,
  b.Nombre_Barrio,
  catL1.Nombre_Categoria_L1,
  catL2.Nombre_Subcategoria_L2;
```

Nota. Captura de pantalla. Consulta de prueba con las principales tablas relacionadas a datos personales y datos comerciales

Figura 72

Tabla de Respuesta de la Consulta de Prueba Aplicada

number	Nombres	Apellidos	id_comercial	Nombre_Comercial	barrio	nombre_categoria_l1	nombre_subcategoria	dias_atencion	horario_atencion
1	Camila	Rodríguez	357	Panadería El Sol	El Virrey	Comidas_Rapidas	Arepas_Empanadas_Chuzos	Lunes A Sabado	10:00:00 - 23:00:00
2	Camila	Rodríguez	357	Panadería El Sol	El Virrey	Comidas_Rapidas	Hamburguesas_Perros_Mas	Lunes A Sabado	10:00:00 - 23:00:00
3	Jorge	Martínez	358	Autos JM	La Andrea	Regalos_Desayunos_Floristeria	Estilistas_Independientes	Lunes - Viernes	09:30:00 - 16:30:00
4	Laura	Gómez	359	Ropa Urbana	Usminia	Comidas_Rapidas	Productos_Personalizados	Lunes A Sabado	10:00:00 - 23:00:00
5	Andrés	Díaz	360	Café Monte Verde	Usminia	Regalos_Desayunos_Floristeria	Pizzeria	Lunes - Viernes	09:30:00 - 16:30:00
6	Mariana	Torres	361	Librería Omega	La Aurora 1	Comidas_Rapidas	Centros_Estetica_Spa	Lunes A Sabado	10:00:00 - 23:00:00
7	Sebastián	Pardo	362	TecniCel	El Virrey	Regalos_Desayunos_Floristeria	Floristería_Plantas_Hierbas	Lunes - Viernes	09:30:00 - 16:30:00
8	Sebastián	Pardo	362	TecniCel	El Virrey	Regalos_Desayunos_Floristeria	Productos_Personalizados	Lunes - Viernes	09:30:00 - 16:30:00
9	Valentina	Niño	363	Boutique BellaFlor	La Andrea	Comidas_Rapidas	Mazorcada	Lunes A Sabado	10:00:00 - 23:00:00
10	Valentina	Niño	363	Boutique BellaFlor	La Andrea	Comidas_Rapidas	Pizzeria	Lunes A Sabado	10:00:00 - 23:00:00
11	Daniel	Ruiz	364	Ferretería El Clavo	La Aurora 1	Regalos_Desayunos_Floristeria	Tiendas_Naturistas	Lunes - Viernes	09:30:00 - 16:30:00
12	Carolina	Méndez	365	Peluquería Glamour	Alfonso Lo…	Comidas_Rapidas	Floristería_Plantas_Hierbas	Lunes A Sabado	10:00:00 - 23:00:00
13	Carolina	Méndez	365	Peluquería Glamour	Alfonso Lo…	Comidas_Rapidas	Tiendas_Detalles_Peluches	Lunes A Sabado	10:00:00 - 23:00:00
14	Felipe	Castaño	366	Papelería PapelArte	Alfonso Lo…	Regalos_Desayunos_Floristeria	Arepas_Empanadas_Chuzos	Lunes - Viernes	09:30:00 - 16:30:00
15	Felipe	Castaño	366	Papelería PapelArte	Alfonso Lo…	Regalos_Desayunos_Floristeria	Mazorcada	Lunes - Viernes	09:30:00 - 16:30:00

Nota. Captura de pantalla en MySQL. Esta tabla muestra registros de comercios con su categoría, asociando el barrio, datos personales, horario de atención. Se ven repetidos algunos registros se debe a la relación de uno a muchos entre comercios y subcategorías: cada comercio puede estar asociado a múltiples subcategorías, lo que produce múltiples filas con la misma información comercial pero distinta subcategoría

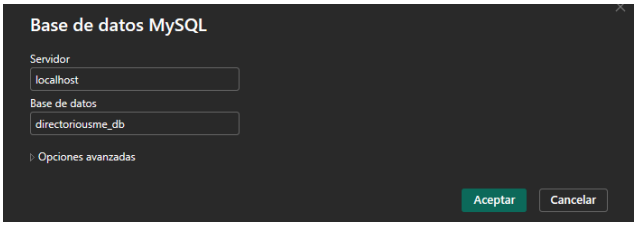
Para continuar con la validación funcional del sistema, a cada uno de los 10 registros se le asignó una URL única, lo que permitió su vinculación con métricas alojadas en las tablas de comportamiento digital. A partir de esta integración, se llevaron a cabo los siguientes pasos clave: Asignación de URLs. A cada negocio se le generó un identificador tipo slug (ejemplo: panaderia-el-sol, autos-jm) y se insertó un registro correspondiente en la tabla dim_pagina, asociando así el comercio con su respectiva página web. Esta relación se estableció mediante la clave foránea Id_URL_Pagina_FK. Inserción de métricas simulados. En Google Analytics (Data_GA_Paginas) se agregaron registros simulados que incluyen datos como usuarios nuevos, sesiones, porcentaje de rebote y duración media, todos vinculados a las URLs creadas previamente. Inserción de métricas orgánicas. En Search Console (Data_GSC_Pagina) se registraron también datos simulados como clics, impresiones, CTR y posición promedio, igualmente conectados por medio del campo Pagina_Destino. Estas inserciones permitieron

evaluar la consistencia relacional del modelo, así como la posibilidad de realizar consultas integradas para alimentar visualizaciones en herramientas con información estructurada.

Visualización en Power BI**:** Una vez comprobada la relación entre tablas, se conectó Power BI directamente a MySQL. Esta integración permitió construir visualizaciones interactivas a partir de los datos simulados, generando los siguientes gráficos de validación:

Figura 73

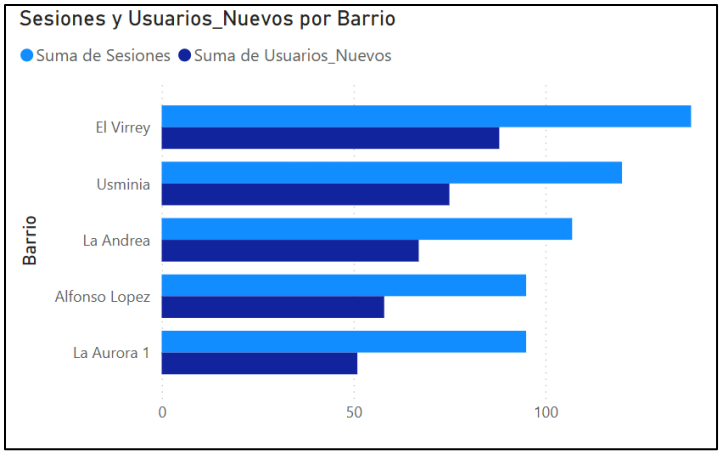
Conexión a Power BI



Nota. Captura de pantalla de la conexión entre MySQL y Power BI

Figura 74

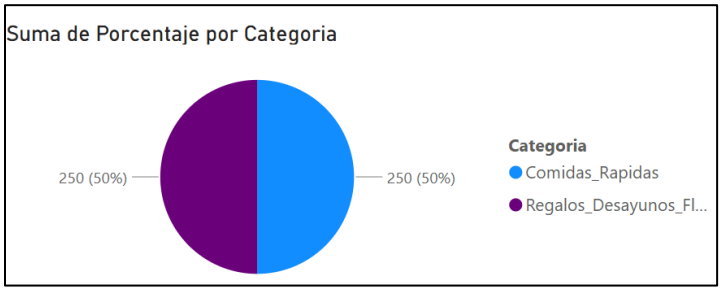
Gráfica de Barras. con Suma de Sesiones y Usuarios Nuevos por Barrio (Junio)



Nota. Captura de pantalla en Power BI. Grafica de la relación de Sesiones y usuarios nuevos por barrio, de acuerdo con los datos simulados ingresados

Figura 75

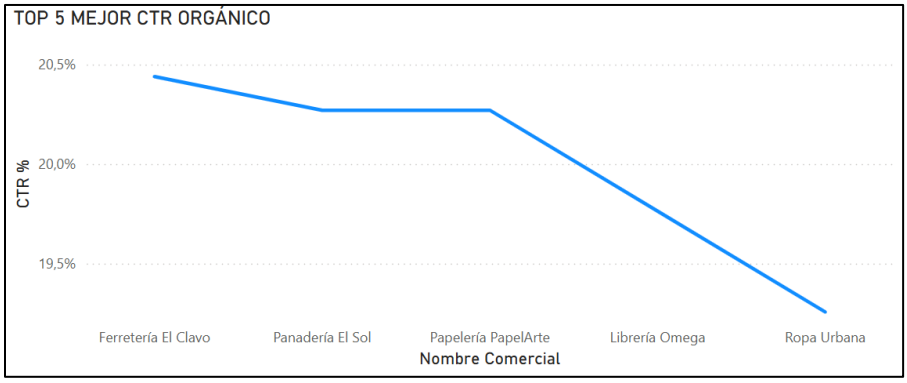
Gráfica de Pastel con el Porcentaje de Participación por Categoría Registrada



Nota. Captura de pantalla en Power BI. Grafica de la relevancia que tienen las diferentes categoria, de acuerdo con los datos simulados ingresados

Figura 76

Gráfico de Líneas Mostrando el Top 5 Negocios con Mejor CTR Orgánico



Nota. Captura de pantalla en Power BI. Grafica para identificar por nombre comercial, los registros que más generan CTR orgánico, de acuerdo con los datos simulados ingresados

Las consultas de prueba permitieron verificar de forma integral el correcto funcionamiento del sistema propuesto, desde la recolección estructurada de datos hasta su transformación y carga en el modelo relacional.

El uso de registros simulados controlados garantizó que todas las rutas de inserción, las relaciones entre tablas y los procesos de análisis en Power BI pudieran ser evaluados sin riesgos, asegurando la calidad y escalabilidad del sistema en un entorno real. Por lo tanto, la selección de datos de prueba fue fundamental para validar la operatividad del sistema antes de ponerlo en producción.

Obtención del Modelo

Una vez definida la estructura lógica del modelo relacional mediante el Diagrama Entidad-Relación (DER), se procedió a su implementación física en el sistema gestor de base de datos MySQL, utilizando la herramienta MySQL Workbench.

Este paso implicó crear las tablas, claves primarias, claves foráneas y restricciones necesarias para garantizar la integridad de los datos y permitir las consultas relacionales entre entidades.

La creación del modelo se realizó teniendo como referencia directa el DER diseñado previamente (ver Figura 30), el cual identifica claramente las entidades principales (Reg_Datos_Personales, Reg_Datos_Comerciales), las entidades categóricas (Cat_Categoria_L1, Cat_Subcategoria_L2, Zn_Ubicacion_Barrio), y las entidades intermedias que gestionan relaciones muchos a muchos. Una vez creada la base de datos, se procedió con la creación de cada tabla siguiendo las estructuras normalizadas diseñadas en fases anteriores. Se respetaron las siguientes características:

- Asignación de claves primarias auto incrementales (PK).
- Establecimiento de claves foráneas (FK) para mantener la integridad referencial entre tablas.
- Normalización hasta la tercera forma normal (3FN).

- Tipos de datos adecuados para cada campo (ej. VARCHAR, TEXT, DATETIME, TINYINT, etc.).

Con esta implementación, se logró trasladar el diseño lógico a un entorno operativo en **MySQL**, habilitando el sistema para recibir datos reales mediante el proceso ETL y preparar la base para la integración con la herramienta analítica Power BI.

Complemento fuentes externas, cabe destacar que en etapas anteriores se abordó el proceso de extracción, transformación y carga (ETL) de las fuentes de datos principales, las cuales contenían tanto información personal como comercial. Se realizó la división de esta información y su posterior inserción en las tablas intermedias diseñadas para dicho fin. A continuación, se detallan las fuentes de datos externas y el proceso ETL desarrollado a la medida para su integración al sistema gestor de base de datos de DirectorioUsme.com

Las fuentes externas de sitio web DirectorioUsme.com integra tres herramientas clave del ecosistema Google como fuentes externas de datos: Google Analytics (GA), Google Search Console (GSC) y Google AdSense (GADSE). Estas plataformas proporcionan métricas esenciales que respaldan la toma de decisiones estratégicas, al ofrecer información cuantitativa sobre el comportamiento de los usuarios, el rendimiento de las páginas y la monetización del sitio.

GA - comportamiento de las páginas en este reporte extrae información sobre todas las URLs vinculadas al sitio, entre ellas los perfiles de cada negocio o servicio registrado en el directorio. Las métricas incluyen:

- Página de destino
- Usuarios nuevos y recurrentes
- Sesiones

- Tiempo promedio de sesión
- Porcentaje de rebote, entre otros.

GA - comportamiento de los usuarios por dispositivo en este reporte proporciona información detallada sobre cómo acceden los usuarios al sitio web DirectorioUsme.com, clasificando los datos según el tipo de dispositivo utilizado, **(desktop, mobile, tablet)**. Este análisis permite entender las preferencias de navegación y optimizar la experiencia del usuario en función del canal de acceso. Las métricas incluidas son:

- Categoría del dispositivo (desktop, mobile, tablet)
- Usuarios nuevos y recurrentes
- Sesiones
- Tiempo promedio de sesión
- Porcentaje de rebote, entre otros.

Google search console contiene la información extraída de esta fuente se estructura y transforma en cinco entidades clave: consulta, página, dispositivo, país y fecha, las cuales se cargan de forma automatizada en tablas específicas dentro de la base de datos.

Las entidades extraídas son:

- Data_GSC_Consulta: términos de búsqueda orgánica.
- Data_GSC_Pagina: comportamiento por URL indexada.
- Data_GSC_Dispositivo: interacción por tipo de dispositivo.
- Data_GSC_Pais: distribución geográfica.
- Data_GSC_Fecha: resumen diario de métricas.

Google search console contiene la información extraída de esta fuente se relaciona a las métricas generadas por los banners publicados en el sitio web, entre ellos se encuentra impresiones, clics, CTR, ingresos RPM.

Se desarrollaron flujos de extracción, transformación y carga (ETL) para incorporar datos de fuentes externas clave: Google Analytics (GA4), Google Search Console (GSC) y Google AdSense. Estos procesos, implementados en Python 3.11 y ejecutados desde Jupyter Notebook. Los scripts utilizados requieren de bibliotecas especializadas (como google-analytics-data, google-api-python-client y oauth2client) para interactuar con las APIs de Google. Cada flujo se autentica mediante un proyecto de Google Cloud configurado con OAuth 2.0, utilizando claves de servicio que garantizan un acceso seguro y autorizado.

Estructura General del Flujo ETL

Todos los flujos implementados siguen la misma lógica estructural de cinco pasos aplicados a las fuentes externas.:

1. Autenticación. En esta etapa inicial, se establece una conexión segura con los servicios de Google Analytics 4 (GA4), Google Search Console (GSC) y Google AdSense, utilizando las credenciales generadas previamente en Google Cloud Platform. Este paso es fundamental para habilitar el acceso autorizado a las APIs correspondientes.
2. Extracción de datos. Una vez autenticado el acceso, se procede a consultar las métricas del día anterior, según los siguientes criterios por herramienta:
    - Google Analytics (GA4): Se extraen métricas por página como número de sesiones, usuarios únicos, páginas vistas y porcentaje de rebote.

- Google Search Console (GSC): Se obtienen datos relacionados con clics, impresiones, CTR (Click Through Rate) y posición promedio, filtrados por dimensiones como página, dispositivo, país y término de búsqueda.
- Google AdSense: Se recuperan métricas vinculadas con la visibilidad de los anuncios por página. Para este proyecto, no se incluyen datos financieros.

3. Transformación de Datos. Durante esta fase, los datos extraídos son limpiados, eliminando registros duplicados y normalizando los valores de las URLs. Se verifica la existencia de cada URL en la tabla dim_pagina; si una URL no está registrada, se crea automáticamente un nuevo registro para garantizar la trazabilidad y consistencia relacional.
4. Carga de Datos. Una vez procesados, los datos son insertados en las tablas analíticas correspondientes del modelo relacional, entre las cuales se destacan:
    - Data_GA_Paginas
    - Data_GSC_Dispositivo
    - Data_GSC_Consulta
    - Data_GSC_Pagina
    - Data_Adsense_Paginas

Estas tablas permiten consolidar las métricas por fuente y dimensión de análisis.

5. Notificación y Control del Proceso. El flujo ETL se ejecuta automáticamente todos los días a las 00:01 horas, extrayendo la información del día anterior. En caso de error —como fallos de conexión o detección de datos duplicados— el sistema genera una notificación automática al correo institucional soydeusme@directoriousme.com, permitiendo un monitoreo proactivo y una rápida resolución de incidencias.

6. Scripts Desarrollados. Para llevar a cabo estos procesos, se diseñaron y validaron un total de ocho scripts distribuidos de la siguiente manera:

- Google Analytics (2 scripts): Uno para métricas generales del tráfico web y otro para comportamiento por página.
- Google Search Console (5 scripts): Scripts separados para extraer consultas por país, dispositivo, página, fecha y término de búsqueda.
- Google AdSense (1 script): Dedicado a la extracción de métricas relacionadas con la visualización de anuncios en cada página.

Estos scripts están listos para su integración con datos reales y forman la base técnica que permitirá medir el impacto digital del sitio web DirectorioUsme.com desde múltiples dimensiones.

Figura 77

Inicialización o Conexión con una API De Google

```
try:
    os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'credentials.json'
    property_id = '267724401'
    client = BetaAnalyticsDataClient()
    print("✅ Cliente GA inicializado.")
except Exception as e:
    hubo_error = True
    errores.append(f"Error autenticando con GA: {e}")
```

```
✅ Cliente GA inicializado.
```

*Nota***.** La integración con las herramientas de Google se realiza mediante el uso de sus APIs. Para establecer esta conexión, es indispensable crear un proyecto en Google Cloud y generar las credenciales necesarias, lo que permitirá la autenticación y el acceso a los servicios deseados

Figura 78

Script de filtro de fecha para obtener métricas

```
from datetime import datetime, timedelta

# Día anterior
ayer = datetime.today() - timedelta(days=1)
start_date = ayer.strftime('%Y-%m-%d')
end_date = ayer.strftime('%Y-%m-%d')

request = RunReportRequest(
    property=f"properties/{property_id}",
    dimensions=[Dimension(name="date"), Dimension(name="pagePath")],
    metrics=[
        Metric(name="newUsers"), Metric(name="activeUsers"),
        Metric(name="sessions"), Metric(name="averageSessionDuration"),
        Metric(name="bounceRate"), Metric(name="sessionsPerUser")
    ],
    date_ranges=[DateRange(start_date=start_date, end_date=end_date)]
)

try:
    response = client.run_report(request)
    print(f"✅ Reporte GA del {start_date} obtenido.")
except Exception as e:
    hubo_error = True
    errores.append(f"Error ejecutando reporte GA: {e}")
```

```
✅ Reporte GA del 2025-06-30 obtenido.
```

*Nota***.** Para las herramientas de Google conectadas al proyecto, el proceso ETL se ha configurado para extraer automáticamente las métricas correspondientes solo al día anterior a la ejecución

Figura 79

Confirmación de la Transformación de los Datos de Herramientas de Google

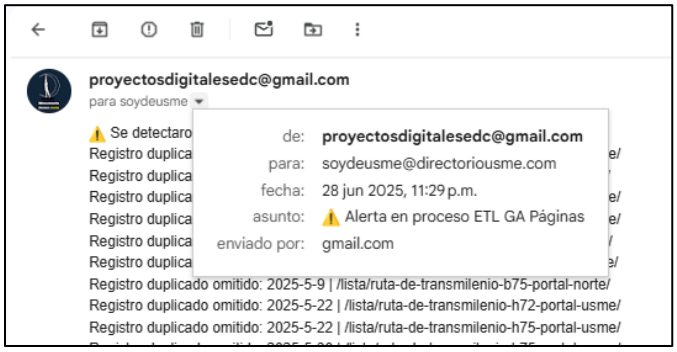
```
# BLOQUE 5 – Transformación de datos
from urllib.parse import urlparse
rows = []
for row in response.rows:
    fecha = row.dimension_values[0].value
    pagina_url = row.dimension_values[1].value  # Ej: https://directoriousme.com/ferreteria-el-clavo
    ruta = urlparse(pagina_url).path            # Ej: /ferreteria-el-clavo
    pagina = ruta.strip('/').lower()            # Ej: ferreteria-el-clavo
    anio = int(fecha[:4])
    mes = int(fecha[4:6])
    dia = int(fecha[6:8])
    nuevos = int(row.metric_values[0].value)
    activos = int(row.metric_values[1].value)
    sesiones = int(row.metric_values[2].value)
    duracion = round(float(row.metric_values[3].value))
    rebote = round(float(row.metric_values[4].value), 2)
    sesiones_usuario = round(float(row.metric_values[5].value), 2)
    recurrentes = max(activos - nuevos, 0)
    rows.append({
        "Anio": anio,
        "Mes": mes,
        "Dia": dia,
        "Pagina_Destino": pagina,
        "Url_Completa": pagina_url,
        "Usuarios_Nuevos": nuevos,
        "Usuarios_Activos": activos,
        "Usuarios_Recurrentes": recurrentes,
        "Sesiones": sesiones,
        "Duracion_Media_Segundos": duracion,
        "Sesiones_Usuario_Activo": sesiones_usuario,
        "Porcentaje_Rebote": rebote
    })

df_paginas = pd.DataFrame(rows)
print("✅ Transformación completada.")
```

```
✅ Transformación completada.
```

*Nota***.** A pesar de que los datos proporcionados por las herramientas de Google vienen optimizados, es necesario aplicar procesos de transformación para separarlos y consolidarlos adecuadamente con las tablas existentes en la base de datos de DirecotioUsme.com

Figura 80

Notificación del Proceso ETL por Correo



*Nota***.** Con cada ejecución del proceso ETL, se enviará una notificación por correo electrónico. Esta puede indicar un éxito sin incidencias o, en caso contrario, alertar sobre fallos o situaciones como la detección de registros duplicados (según se muestra en la captura de pantalla)

Una vez validadas las estructuras relacionales en MySQL y probada la inserción de datos mediante la automatización del proceso ETL (Extract, Transform, Load), se procede con la evaluación del modelo.

Fase 5 Evaluación del Modelo

Evaluación de la Calidad del Modelo

Una vez implementado el modelo relacional en el sistema gestor de base de datos MySQL y validadas sus estructuras mediante datos de prueba, se procedió a evaluar su funcionamiento desde dos enfoques: técnico y funcional. Esta evaluación permite determinar si el sistema cumple con los requisitos establecidos y si es viable su puesta en marcha en un entorno real.

Evaluación técnica desde el punto de vista técnico, se verificó que las claves primarias y foráneas están correctamente definidas. Las relaciones entre tablas funcionan conforme a lo

establecido en el DER. Las inserciones de datos respetan la integridad referencial, sin duplicidades ni errores de clave. El sistema es escalable y permite la inserción continua de registros mediante el proceso ETL.

La Evaluación funcional desde el punto de vista funcional, se confirmó que el modelo responde correctamente a las necesidades del directoriousme.com. Permite almacenar información organizada y detallada sobre cada negocio (nombre, categoría, ubicación, horario, beneficios, etc.). Soporta consultas eficientes que integran múltiples tablas. Está preparado para alimentar herramientas de análisis externo como *Power BI.*

Comparación con la Situación Inicial

Con el propósito de evaluar el impacto del sistema implementado, se realizó una comparación entre el estado inicial del proyecto y la situación actual, posterior a la implementación del modelo relacional, el desarrollo del proceso ETL, la integración con *Power BI* y la validación de consultas analíticas.

Durante la etapa de comprensión del negocio (Fase 1), se identificaron diversas problemáticas que afectaban el manejo de la información, tales como la dispersión de los datos en múltiples archivos no relacionados, la falta de estructura para su almacenamiento y la imposibilidad de realizar análisis estratégicos en tiempos razonables. Esta situación comprometía la calidad de los datos, la toma de decisiones y la experiencia de los usuarios.

A partir de dichos hallazgos, se definieron objetivos orientados a transformar esa realidad, mediante la consolidación de un sistema de gestión de base de datos relacional, procesos automatizados de extracción y carga, así como la implementación de herramientas de visualización. La siguiente tabla presenta los principales aspectos evaluados antes y después del desarrollo del sistema:

Tabla 41

Comparación entre Situación Inicial y Situación Actual del Sistema

Indicador clave	Situación al iniciar	Resultado alcanzado
Estructura organizada de datos	Información dispersa y sin normalizar	Modelo relacional normalizado con claves primarias y foráneas
Ingreso de información	Manual, propenso a errores	Automatizado mediante ETL desde formulario a MySQL
Tiempo de generación de un reporte de tráfico	5–7 días	2 h promedio (Power BI + consultas SQL)
Porcentaje de registros duplicados	10–15 %	< 2 % (notificación automática por correo en caso de duplicados)
Indicador clave	Situación al iniciar	Resultado alcanzado
Tiempo medio de respuesta de una consulta	> 60 s (en hojas) las búsquedas son manuales	1–2 s (consultas JOIN en MySQL)
Número de vistas/consultas automatizadas	No existe	4 vistas creadas en MySQL
Dashboards creados de referencia	No existe	≥ 3 dashboards en desarrollo con Power BI
Existencia de control de accesos por rol	No	Sí (rol administrador y rol de consulta definidos en MySQL)

Nota. La tabla presenta una comparación estructurada entre el estado inicial del sistema de información del sitio web DirectorioUsme.com y la situación actual, luego de la implementación del modelo relacional. Esta comparación evidencia las mejoras logradas en aspectos clave como la organización de los datos, la automatización del ingreso de información, la capacidad de análisis y visualización, así como la escalabilidad del sistema

Toma de Decisiones

A partir de los resultados obtenidos durante la evaluación técnica y funcional del modelo relacional, se concluye que la solución desarrollada cumple con los requisitos definidos en las fases iniciales del proyecto. La estructura de base de datos implementada en **MySQL** demostró ser sólida, al cumplir con los principios de normalización, mantener la integridad referencial y permitir relaciones complejas entre entidades mediante el uso de claves primarias, foráneas y tablas intermedias.

La validación con datos de prueba evidenció el correcto funcionamiento de las consultas relacionales, así como la eficacia del proceso ETL desarrollado en Python para automatizar el ingreso de información desde Google Sheets. Del mismo modo, la integración con Power BI confirmó la capacidad del sistema para generar reportes dinámicos y facilitar la toma de decisiones estratégicas.

Con base en estos resultados, se aprueba la puesta en marcha del sistema en un entorno operativo real. Se considera que el modelo está preparado para escalar con el crecimiento de la base de datos, sin comprometer la calidad, integridad ni disponibilidad de la información. En consecuencia, se da paso a la siguiente fase del proyecto, correspondiente a la implementación final del sistema, donde se documentará los procesos a tener en cuenta en cada etapa.

Fase 6 Implementación

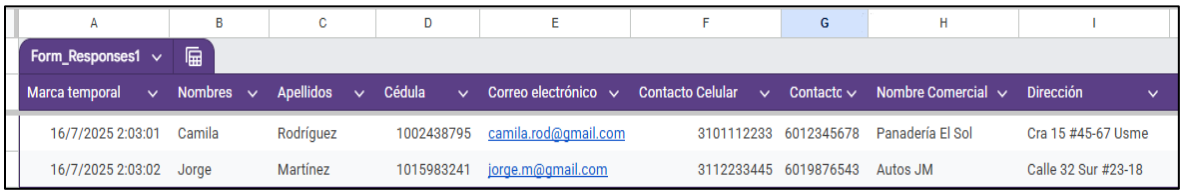
Planificación del Despliegue

El Ingreso de Datos. Tal como se ha mencionado previamente, el proceso inicia con la recolección de información mediante un formulario digital, distribuido a través de canales como el sitio web, WhatsApp y contacto directo. Una vez diligenciado por los usuarios, los datos son

almacenados automáticamente en una hoja de cálculo de Google Sheets, la cual constituye el punto de partida para el flujo de procesamiento posterior.

Figura 81

Datos Almacenados en Google Sheets del Formulario Creado



	A	B	C	D	E	F	G	H	I
	Form_Responses1								
	Marca temporal	Nombres	Apellidos	Cédula	Correo electrónico	Contacto Celular	Contactc	Nombre Comercial	Dirección
	16/7/2025 2:03:01	Camila	Rodríguez	1002438795	camila.rod@gmail.com	3101112233	6012345678	Panadería El Sol	Cra 15 #45-67 Usme
	16/7/2025 2:03:02	Jorge	Martínez	1015983241	jorge.m@gmail.com	3112233445	6019876543	Autos JM	Calle 32 Sur #23-18

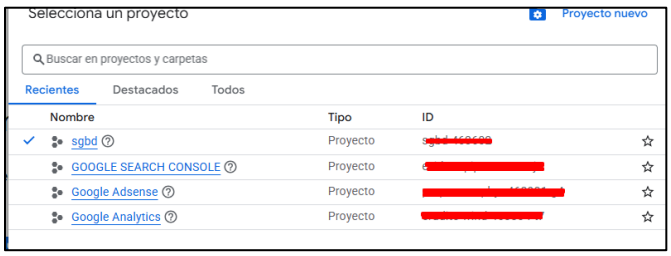
*Nota***.** Captura de pantalla del formulario de registro comercial en Google Sheets

Google cloud Platform garantiza un despliegue eficiente y seguro del sistema, se habilitó una cuenta en Google Cloud Platform (GCP), esta cuenta funciona como intermediaria para el establecimiento de conexiones mediante las APIs necesarias con diversas plataformas.

En esta cuenta se configuraron cuatro proyectos activos: uno vinculado al formulario digital alojado en Google Sheets y tres adicionales destinados a la recolección de métricas clave desde Google Analytics 4, Google Search Console y Google AdSense. En la siguiente página se visualiza en la figura 82 los 4 proyectos utilizados en la consola de *Google Cloud Platform.*

Figura 82

Proyectos Creados en Google Cloud Platform para el Despliegue del Sistema



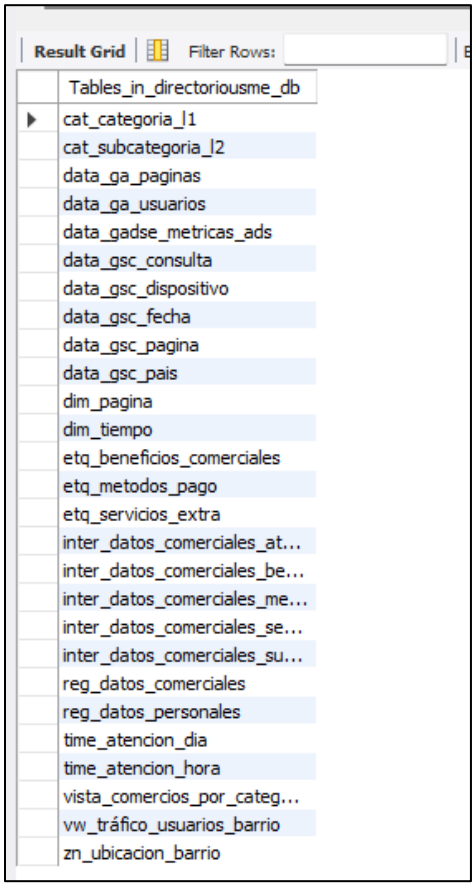
*Nota***.** Captura de la consola de Google Cloud con los cuatro proyectos configurados

Ejecución del proceso ETL una vez capturada la información desde el formulario digital, los datos son procesados mediante scripts desarrollados en Python, los cuales ejecutan de forma secuencial las etapas de extracción, transformación y carga (ETL). Este proceso convierte los registros en estructuras normalizadas, listas para ser insertadas automáticamente en la base de datos relacional directoriousme_db.

Las tablas principales del sistema, como Reg_Datos_Personales y Reg_Datos_Comerciales, se relacionan con tablas auxiliares e intermedias (por ejemplo: Cat_Categoria_L1, Cat_Subcategoria_L2, Etq_Servicios_Extra, Time_Atencion_Hora), garantizando la integridad referencial mediante claves primarias y foráneas. En la siguiente página se observa en la Figura 83 la sentencia *show tables* ejecutada en MySQL donde muestra las tablas que actualmente se encuentran en base de datos.

Figura 83

Listado de Tablas Creadas en MySQL Mediante Sentencia Show Tables



*Nota***.** Captura de pantalla en MySQL Workbench. Este listado refleja la implementación física del modelo relacional diseñado para el sistema

Diccionario de Datos se presenta a continuación, el cual describe las entidades que conforman la base de datos relacional del sistema de gestión para el DirectorioUsme.com. Cada tabla está acompañada de una breve descripción que especifica su propósito y función dentro del modelo, facilitando su interpretación, mantenimiento y posterior análisis por parte del equipo técnico o analítico.

Tabla 42

Diccionario de Datos del Sistema de Gestión para el DirectorioUsme.com

Nombre de la tabla	Descripción
Reg_Datos_Personales	Almacena información básica del representante del negocio (nombre, cédula, contacto).
Reg_Datos_Comerciales	Registra datos del negocio como nombre comercial, dirección, categoría y barrio.
Cat_Categoria_l1	Tabla de referencia para las categorías principales de los negocios.
Cat_Subcategoria_l2	Contiene las subcategorías asociadas a cada categoría principal.
Zn_Ubicacion_Barrio	Lista de barrios geográficos utilizados para clasificar los negocios.
Etq_Metodos_Pago	Catálogo de métodos de pago ofrecidos (efectivo, tarjeta, Nequi, etc.).
Etq_Beneficios_Comerciales	Etiquetas que describen beneficios comerciales (envío gratis, promociones, etc.).
Etq_Servicios_Extra	Servicios adicionales ofrecidos por los negocios (domicilio, asesoría, etc.).

Nombre de la tabla	Descripción
Dim_Pagina	Tabla que registra las URLs de los perfiles creados en el sitio web.
Dim_Tiempo	Dimensión temporal utilizada para análisis por fecha (día, mes, año).
Inter_Datos_Comerciales_Subcategoria_L2	Tabla intermedia para gestionar relación N:M entre negocios y subcategorías.
Inter_Datos_Comerciales_Metodos_Pago	Tabla intermedia para métodos de pago asociados a un negocio.
Inter_Datos_Comerciales_Beneficios_Comerciales	Relaciona negocios con beneficios comerciales disponibles.
Inter_Datos_Comerciales_Servicios_Extra	Vincula cada comercio con los servicios adicionales que ofrece.
Inter_Datos_Comerciales_Atencion_Dia_Hora	Gestión de horarios y días de atención por comercio.
Data_GA_Paginas	Métricas de tráfico web por URL, obtenidas desde Google Analytics.
Nombre de la tabla	Descripción
Data_GA_Dispositivo	Información sobre acceso por tipo de dispositivo (mobile, desktop, etc.).
Data_GSC_Consulta	Términos de búsqueda orgánica desde Google Search Console.

Nombre de la tabla	Descripción
Data_GSC_Pagina	Rendimiento de cada URL en buscadores.
Data_GSC_Dispositivo	Métricas por tipo de dispositivo (GSC).
Data_GSC_Pais	Distribución geográfica de las búsquedas.
Data_GSC_Fecha	Métricas diarias de rendimiento desde GSC.
Data_GADSE_Metricas_Ads	Información sobre impresiones, clics y CTR de anuncios de Google AdSense.

*Nota***.** Estas entidades fueron diseñadas aplicando los principios de normalización, asignando nombres según la nomenclatura técnica establecida, tal como se evidencia en la Tabla 41 y en el Diagrama Entidad-Relación (DER) descrito en la Fase 4: Modelado.

Programador de Tareas

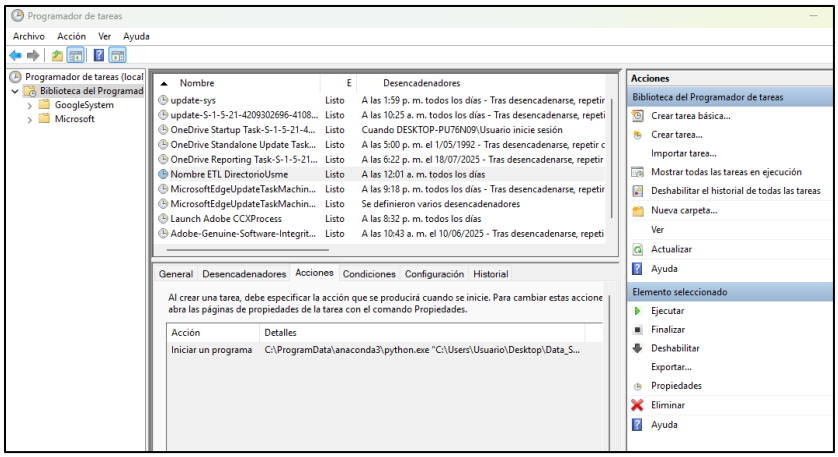
Cabe destacar que el proceso ETL ha sido completamente automatizado mediante el Programador de tareas de Windows (*Task Schedule*r), herramienta que permite ejecutar scripts de manera programada y sin supervisión manual.

En este caso, se ha configurado para que el flujo principal de carga y transformación de datos se ejecute todos los días a las **00:01 horas**, asegurando así la recolección y procesamiento oportuno de los registros del día anterior.

Cada fuente de datos ya sea interna (formularios de Google Sheets) o externa (Google Analytics, Google Search Console y Google AdSense)— dispone de su propio script independiente, lo que permite modularidad, trazabilidad y un mayor control de errores.

Figura 84

Tarea Programada en Windows para Ejecución Diaria del Proceso ETL



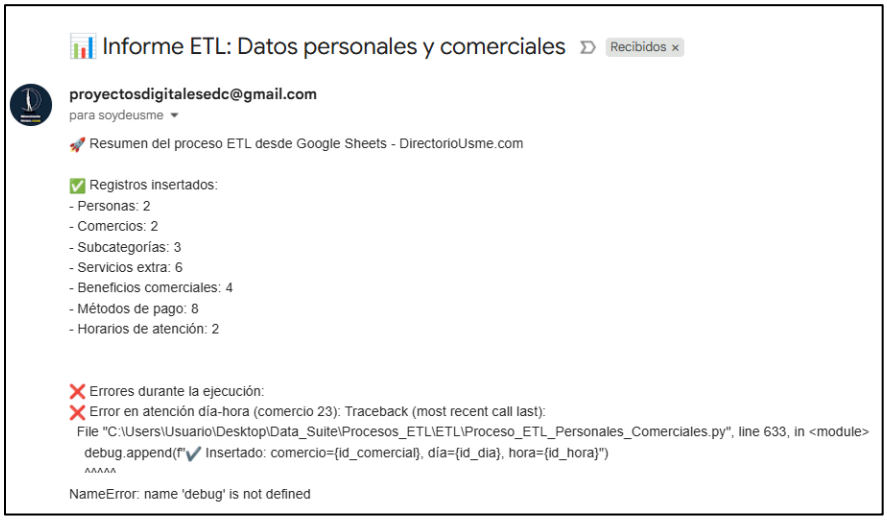
*Nota***.** Captura de pantalla del programador de tareas mostrando el script configurado para ejecutarse todos los días a las 00:01 a.m

Una vez finalizada cada ejecución del flujo ETL, el sistema genera automáticamente un informe de seguimiento que es enviado al correo institucional (**soydeusme@directoriousme.com**).

Este informe incluye un resumen detallado con la cantidad de registros procesados por tabla y notifica cualquier error, inconsistencia o novedad detectada durante la ejecución.

Figura 85

Correo de Notificación Automática con el Resumen del Proceso ETL



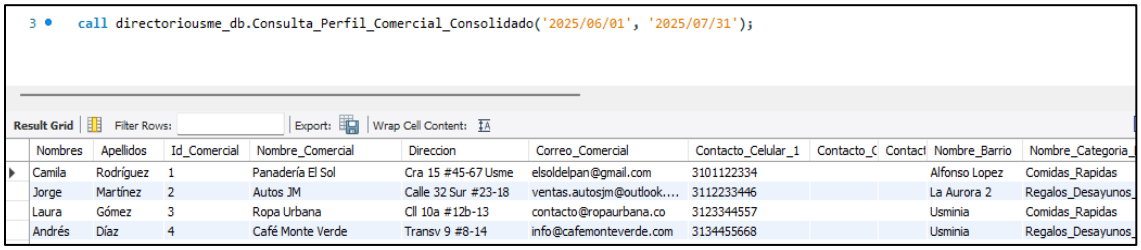
*Nota***.** Captura de pantalla del correo institucional mostrando el detalle de registros insertados y alertas del sistema

Procedimiento Almacenado para Activación de Registro

Una vez que los datos han sido integrados correctamente en la base de datos, se ejecuta el procedimiento almacenado denominado Consulta_Perfil_Comercial_Consolidado, el cual permite obtener un resumen estructurado de los registros nuevos ingresados durante un período específico. Este procedimiento consolida información clave como el nombre comercial, ubicación, barrio, categoría principal, subcategorías, horarios de atención, etiquetas asociadas y datos del representante legal, cruzando múltiples tablas del sistema. La información obtenida se descarga en un archivo CSV y es enviada al *tracfficker*" quien se encarga de crear manualmente el perfil y la página del negocio o servicio, la cual será visible dentro del sitio web **DirectorioUsme.com**. Esta información es pública y se basa en los datos estructurados proporcionados.

Figura 86

Resultado del Procedimiento Consulta_Perfil_Comercial_Consolidado



```
call directoriousme_db.Consulta_Perfil_Comercial_Consolidado('2025/06/01', '2025/07/31');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Nombres	Apellidos	Id_Comercial	Nombre_Comercial	Direccion	Correo_Comercial	Contacto_Celular_1	Contacto_C	Contact	Nombre_Barrio	Nombre_Categoria_
Camila	Rodríguez	1	Panadería El Sol	Cra 15 #45-67 Usme	elsoldelpan@gmail.com	3101122334			Alfonso Lopez	Comidas_Rapidas
Jorge	Martínez	2	Autos JM	Calle 32 Sur #23-18	ventas.autosjm@outlook....	3112233446			La Aurora 2	Regalos_Desayunos_
Laura	Gómez	3	Ropa Urbana	Cll 10a #12b-13	contacto@ropaurbana.co	3123344557			Usminia	Comidas_Rapidas
Andrés	Díaz	4	Café Monte Verde	Transv 9 #8-14	info@cafemonteverde.com	3134455668			Usminia	Regalos_Desayunos_

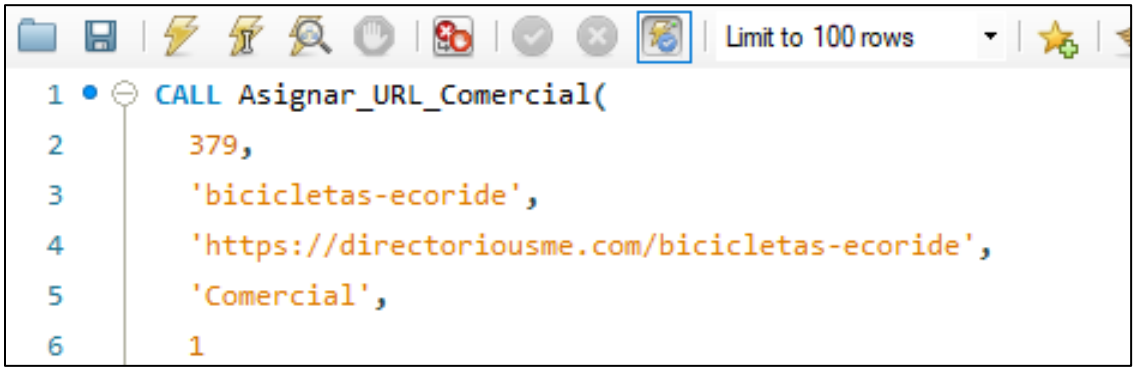
*Nota***.** Captura de pantalla en MySQL. Consolidado que reúne columnas de las tablas personales, comerciales, categorías, subcategorías y ubicación

Procedimiento Almacenado para Asignar URL Comercial

Una vez publicada la página del negocio, se genera una URL única que identifica dicho perfil comercial dentro del sitio. Para registrar esta URL en la base de datos, se utiliza un segundo procedimiento almacenado denominado Asignar_URL_Comercial, el cual permite asociar la URL generada al identificador del comercio (Id_Comercial) dentro de la tabla dim_pagina.

Figura 87

Ejecución del Procedimiento Asignar_URL_Comercial



*Nota***.** Captura de pantalla del procedimiento con ejemplo de asignación para el comercio bicicleta-ecoride

Este proceso cierra el ciclo de activación del registro, garantizando que la información publicada en el sitio web esté correctamente enlazada y documentada dentro del modelo relacional del sistema.

Exploración de Datos y Toma de Decisiones

Para la visualización gráfica y el análisis dinámico de la información almacenada, se utiliza Power BI, conectado directamente con la base de datos MySQL. Esta integración permite ejecutar consultas SQL que alimentan paneles interactivos, facilitando la exploración de los datos por parte del equipo interno del proyecto.

Gracias al modelo relacional estructurado, es posible generar vistas analíticas y alimentar dashboards en tiempo real.

Esto fortalece la capacidad de monitoreo de métricas clave y el análisis del rendimiento tanto del sitio web como del comportamiento de los usuarios.

Este entorno visual permite:

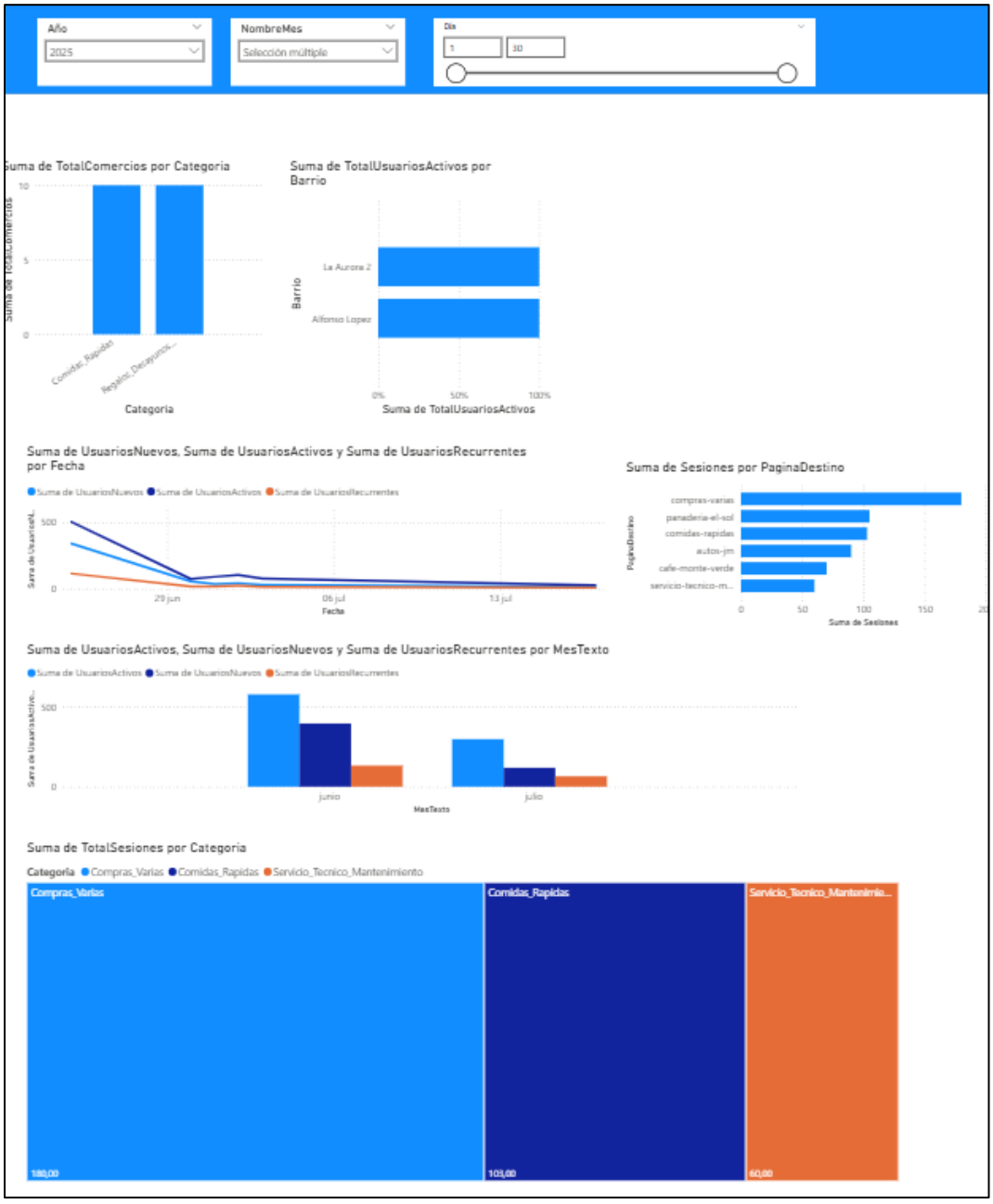
- Presentar informes de gestión periódicos de manera automatizada.
- Redefinir dinámicamente las categorías destacadas en el sitio web, según las tendencias de búsqueda o el tráfico por barrio.
- Identificar oportunidades de mejora en la experiencia del usuario y la arquitectura del sitio.
- Diseñar campañas digitales estacionales más efectivas.
- Tomar decisiones estratégicas basadas en datos, orientadas a la sostenibilidad y crecimiento del DirectorioUsme.com como ecosistema digital local.

En la siguiente página, se puede visualizar en la Figura 88 diferentes gráficos realizados en Power BI. Estos gráficos pueden consultarse por año, mes o día y presentan información variada, como: usuarios por barrio, tráfico de usuarios nuevos y recurrentes por barrio, top de páginas, principales categorías con mayor tráfico, entre otros.

Todos estos gráficos se generan a partir de consultas a la base de datos, que pueden ser tanto consultas directas a una tabla como a una vista predefinida con los parámetros necesarios para el negocio.

Figura 88

Dashboard en Power BI Conectado a MySQL.



*Nota***.** Captura de pantalla del dashboard. Se visualizan distintos gráficos manipulables dinámicamente: Gráfico de barras que muestra la cantidad de negocios por categoría. Distribución de usuarios por barrio.Serie temporal de usuarios activos, nuevos y recurrentes. Filtros dinámicos por año, mes y día.

Cierre de la Arquitectura General

La implementación del sistema de gestión de datos para el DirectorioUsme.com se realizó en un entorno local controlado. Se utilizó un equipo con sistema operativo Windows 8, 8 GB de memoria RAM y el gestor de base de datos MySQL 8.0, operado a través de MySQL Workbench.

Esta configuración fue seleccionada por su estabilidad, escalabilidad, y por ofrecer integración directa con herramientas de análisis como Power BI, así como compatibilidad total con Python 3.11 para el desarrollo de procesos automatizados.

La arquitectura construida permite capturar, transformar, validar y explotar datos provenientes de diversas fuentes, incluyendo formularios en línea y plataformas de analítica digital como Google Analytics, Search Console y AdSense.

Todo el flujo desde el ingreso de datos, la ejecución del proceso ETL, el análisis en SQL, hasta la presentación visual en Power BI ha sido diseñado para garantizar la integridad, trazabilidad y utilidad de la información.

La Figura 89 sintetiza este proceso, mostrando de manera gráfica cómo los datos recorren cada una de las etapas, desde su origen hasta su conversión en conocimiento útil para la toma de decisiones estratégicas.

Este diseño arquitectónico no solo cumple con el objetivo general del proyecto “consolidar y analizar la información del DirectorioUsme.com mediante herramientas tecnológicas”, sino que también responde a cada uno de los objetivos específicos:

- Diseño estructurado de la base de datos: Se estableció una arquitectura relacional basada en el análisis de fuentes internas (formulario web) y externas (herramientas de Google), permitiendo organizar los datos de manera coherente y normalizada.

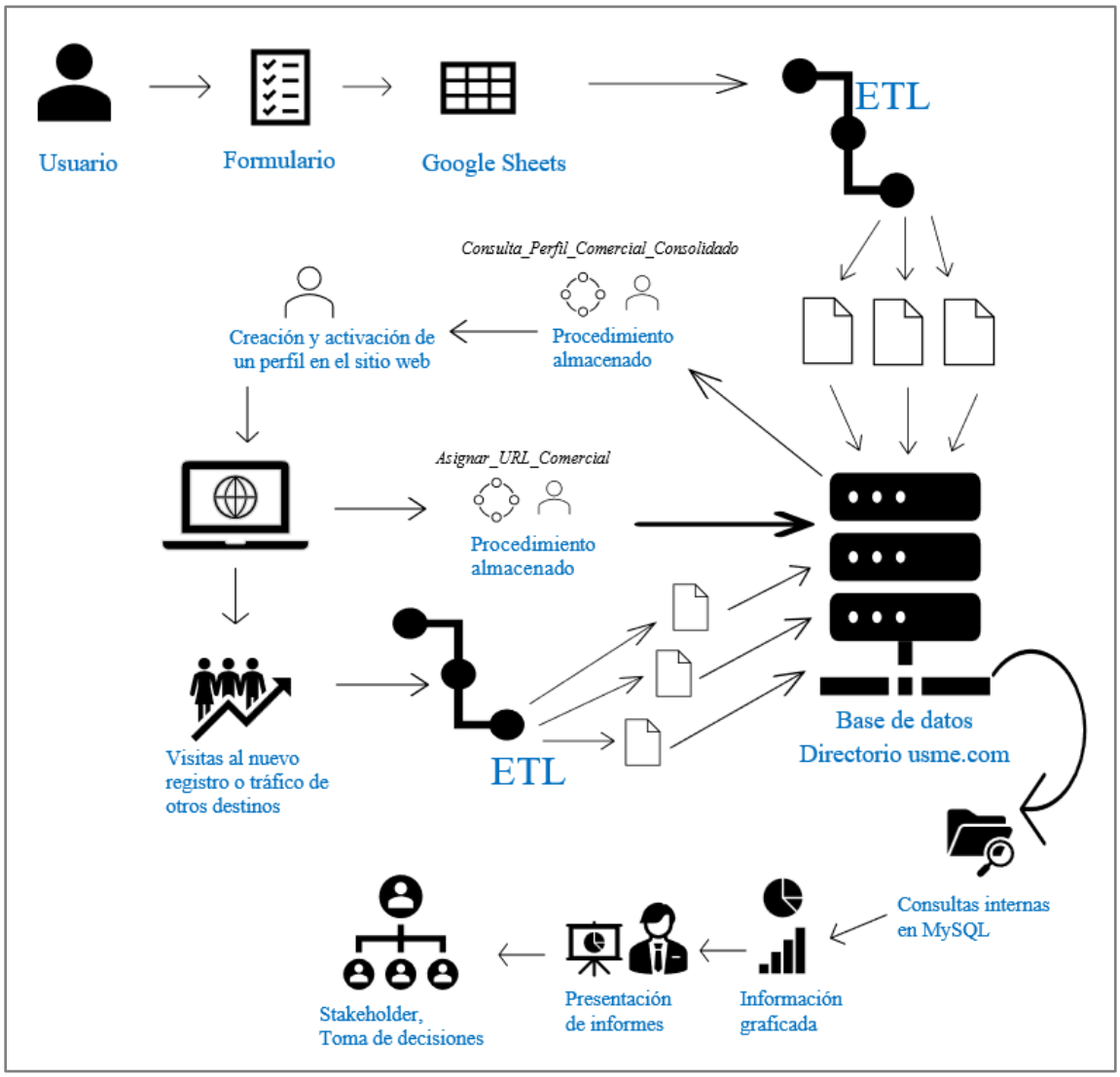
- Implementación del proceso ETL: Se desarrolló un flujo automatizado utilizando Python y MySQL, que permite extraer, transformar y cargar eficientemente la información hacia la base de datos centralizada.
- Funcionalidades de consulta y análisis: Se diseñaron dashboards interactivos en Power BI, alimentados por consultas en MySQL, lo que permite explorar la información consolidada de forma dinámica y facilita la toma de decisiones basada en datos reales.

En conjunto, esta arquitectura se consolida como el eje central del sistema, aportando una solución sostenible, escalable y alineada con los fines estratégicos del sitio web y con la misión del DirectorioUsme.com como plataforma digital de alcance comunitario.

Asimismo, responde a los requerimientos del proyecto académico, demostrando la viabilidad de integrar tecnología, automatización y análisis de datos para mejorar la gestión de plataformas digitales con impacto local.

Figura 89

Arquitectura General del Sistema de Gestión de Datos del Directoriousme.Com



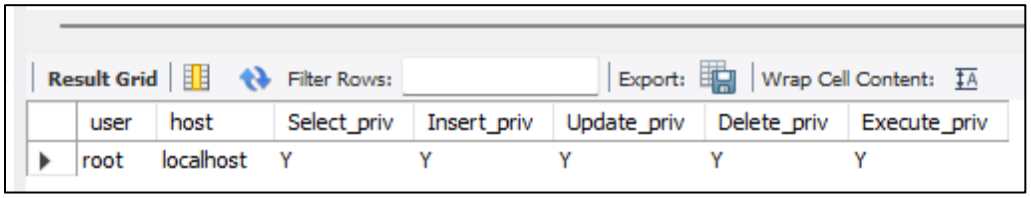
*Nota***.** Elaboración propia.

Documentación Seguridad y Manuales del Sistema

Como medida de protección para la base de datos, se establecieron dos tipos de usuarios con roles diferenciados: un perfil administrativo (root) y un perfil limitado (consultor).

Figura 90

Permisos Asignados al Usuario Root, con Privilegios Totales en Base de Datos

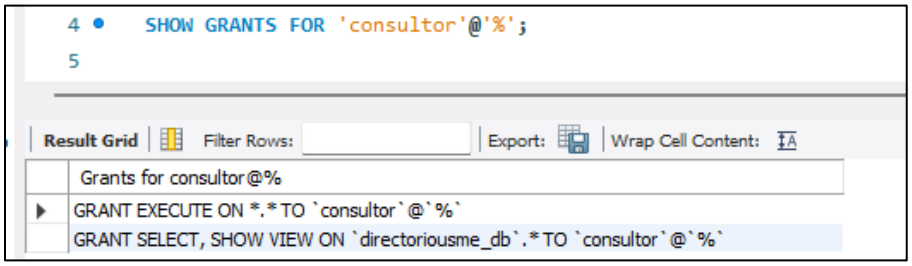


*Nota***.** Captura de pantalla del entorno MySQL Workbench

El usuario root cuenta con acceso completo al sistema de gestión, lo que incluye la capacidad de crear, modificar y eliminar estructuras y registros, así como la administración integral de los esquemas y procedimientos almacenados. En contraste, se definió el perfil consultor, creado específicamente para realizar consultas personalizadas y ejecutar procedimientos almacenados previamente autorizados. Este usuario no posee permisos de edición, eliminación ni acceso a bases del sistema como mysql, sys o performance_schema.

Figura 91

Permisos Asignados al Usuario "Consultor"



*Nota***.** Captura de pantalla en MySQL Workbench

Este esquema de privilegios diferenciados aporta varias ventajas:

Mayor seguridad, los usuarios limitados no pueden manipular directamente los datos, lo que mitiga riesgos como la inyección de SQL o modificaciones accidentales.

Eficiencia operativa, los procedimientos almacenados están precompilados en el servidor, lo que optimiza el rendimiento y reduce el tráfico entre cliente y servidor.

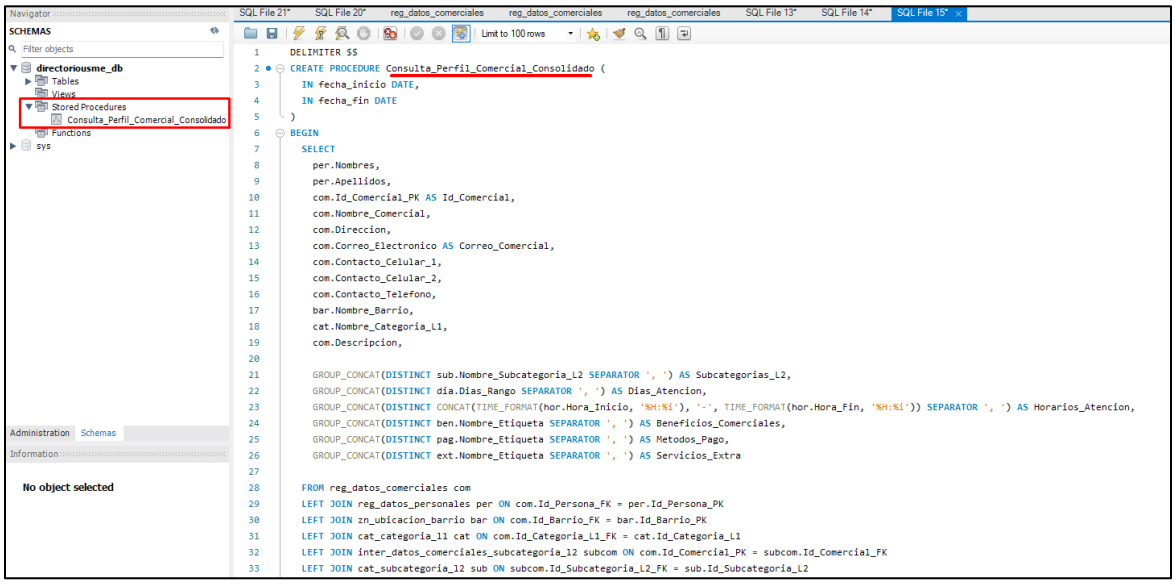
Reutilización y consistencia, encapsular la lógica de negocio en procedimientos garantiza que los procesos se ejecuten de forma uniforme y facilita el mantenimiento futuro del sistema.

Procedimientos Almacenados Consulta Perfil Comercial

Para consolidar la información de cada nuevo registro y facilitar su revisión por parte del tracfficker, se desarrolló el procedimiento almacenado Consulta_Perfil_Comercial_Consolidado.

Figura 92

Código del Procedimiento Almacenado Consulta_Perfil_Comercial_Consolidado

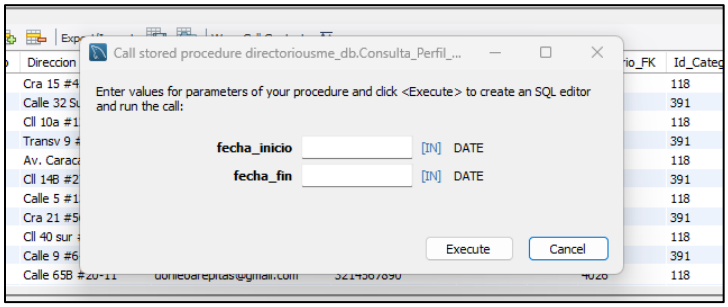


*Nota***.** Captura de pantalla desde MySQL Workbench.

Este procedimiento permite unificar la información relevante de cada comercio en una sola salida estructurada, integrando datos personales, comerciales, de ubicación, categorías, subcategorías y etiquetas. Se diseñó con parámetros de tipo DATE, lo que posibilita ejecutar la consulta para un día específico o para un rango de fechas.

Figura 93

Interfaz del Procedimiento Almacenado, Ingreso de Parámetros Tipo Fecha



*Nota***.** Captura de pantalla de ejecución en MySQL Workbench. La fecha debe ser 2025/07/31 se puede el mismo día o en un periodo de tiempo determinado

El resultado se puede visualizar directamente en el entorno de gestión y exportar en formato .CSV. Este archivo consolidado se remite al trafficker, encargado de verificar manualmente la información y construir el perfil del negocio en el DirectorioUsme.com.

Figura 94

Procedimiento Consulta_Perfil_Comercial_Consolidado

*Nota***.** Captura de pantalla en MySQL

Una vez creado el perfil web del comercio, se genera una URL única que se registra en la tabla dim_pagina, vinculada al identificador del negocio mediante un segundo procedimiento almacenado: Asignar_URL_Comercial.

Procedimientos Almacenados Consulta Asignación URL Comercial

Una vez el perfil del negocio ha sido creado y publicado en el sitio web DirectorioUsme.com, el equipo operativo cuenta con un procedimiento almacenado diseñado específicamente para registrar y vincular la URL generada con el registro correspondiente en la base de datos. El procedimiento denominado Asignar_URL_Comercial permite Insertar una nueva URL en la tabla dim_pagina.

Capturar automáticamente el identificador único (Id_Pagina_PK) generado.

Asignarlo al campo Id_URL_Pagina_FK dentro de la tabla Reg_Datos_Comerciales.

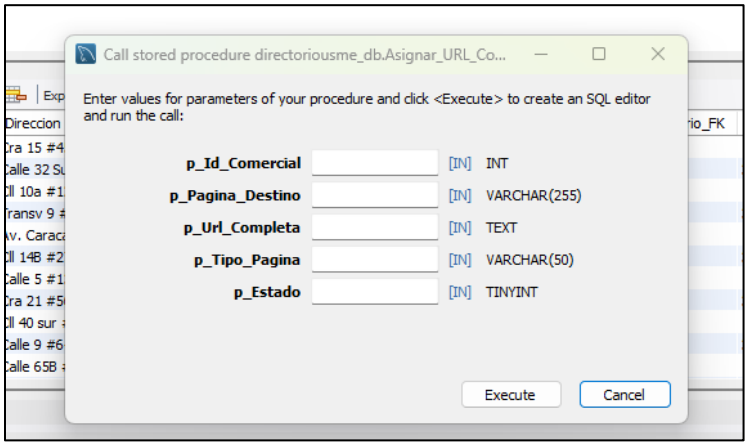
Esta implementación encapsula la lógica de negocio, garantizando la integridad referencial del modelo relacional, la trazabilidad de cada perfil dentro del sistema y la minimización de errores por manipulación directa.

La siguiente Figura 95 muestra el cuadro de ejecución automática que aparece al ejecutar el procedimiento almacenado desde MySQL Workbench. En él, se deben ingresar los siguientes parámetros:

- Id_Comercial: Identificador único del negocio.
- Nombre_Destino: Nombre semántico o simplificado de la URL.
- URL_Completa: Enlace completo del perfil publicado.
- Tipo_Pagina: Clasificación como “comercial” o “institucional”.
- Estado: Indicador binario (1 = activo, 0 = inactivo).

Figura 95

Ventana de Ejecución del Procedimiento Almacenado Asignar_URL_Comercial



Nota.: Elaboración propia. Captura en MySQL

Gracias a esta automatización, cada registro comercial queda vinculado correctamente con su respectiva URL pública, permitiendo, su seguimiento posterior desde herramientas analíticas como Google Search Console, la integración con visualizaciones en Power BI, facilitando el análisis de tráfico, visibilidad y rendimiento del portal web.

La estrategia de respaldo se definió que el primer día de cada mes se descargará una copia del archivo de Google Sheets con los datos del formulario, la cual será almacenada en el directorio local data_suite/backup/forms. Esta carpeta estará sincronizada con una cuenta de Google Drive con cuenta del DirectorioUsme.com para tener un respaldo en la nube. Adicionalmente, ese mismo día se almacenará una copia externa en un disco duro físico en custodia del fundador del proyecto.

El monitoreo y mantenimiento se realizará una revisión al inicio del mes consultando las tablas de Reg_Datos_Personales y Reg_Datos_Comerciales si hay registros duplicados y se validará la integridad referencial de la base de datos. Adicional se prevé realizar una revisión de

los scripts cada 3 meses donde se revisarán y si alguna de las fuentes de datos (APIs de Google) cambia o antes si se detecta un mal procedimiento o se integra nuevas fuentes.

Resultados

Resultado 1: Se diseñó una arquitectura robusta de base de datos relacional para el sitio DirectorioUsme.com, partiendo de un análisis exhaustivo de las fuentes de información disponibles y su estructura inicial. El modelo se construyó aplicando principios de normalización hasta la Tercera Forma Normal (3FN), modelando cuidadosamente las entidades principales, sus atributos y relaciones, con especial atención a aquellas de tipo muchos a muchos, gestionadas mediante tablas intermedias.

Esta estructura permitió transformar información previamente dispersa en un modelo centralizado, coherente y escalable, que garantiza la integridad de los datos y mejora significativamente su organización. El modelo fue implementado físicamente en MySQL Workbench como se evidencia en la Figura 83, incorporando claves primarias, foráneas y restricciones referenciales para asegurar la consistencia entre las entidades. (ver Tabla 42).

Con esta implementación se cumplió uno de los objetivos principales del proyecto: disponer de una estructura clara y organizada que permita realizar consultas cruzadas entre tablas, responder con mayor agilidad a las necesidades del negocio y proteger la calidad de la información. Además, se establecieron convenciones de nombres para las tablas, reglas de funcionamiento y esquemas que facilitarán futuras integraciones con otras fuentes de datos.

Durante el proceso se identificaron oportunidades de mejora no previstas inicialmente, como la necesidad de campos adicionales para controlar la fecha de modificación o actualización de registros, así como la implementación de respaldos periódicos para garantizar la trazabilidad y la seguridad de la información. Estos hallazgos contribuyeron a mejorar la limpieza, orden y control del sistema, sentando una base sólida para el crecimiento futuro del sitio web DirectorioUsme.com.

Resultado 2: Durante el desarrollo del proyecto se logró una implementación del proceso ETL de fuentes internas y externas**.** Con un total de ocho scripts de ETL utilizando Python 3.11, los cuales permiten extraer, transformar y cargar datos desde distintas fuentes, incluyendo formularios de Google, Google Analytics, Google Search Console y Google AdSense, hacia una base de datos relacional en MySQL.

Es un proceso automatizado mediante el Programador de Tareas de Windows (ver Figura 84), con ejecución diaria a las 00:01 horas, y se configuró el envío automático de notificaciones por correo electrónico en caso de errores, duplicados o fallos en la ejecución (ver Figura 85)

El flujo de trabajo automatizado de tipo ETL (Extract, Transform, Load) permite la conexión con APIs externas y hojas de cálculo dinámicas, facilita la limpieza y estandarización de los datos y garantiza su carga estructurada en la base de datos.

El proceso no solo minimiza errores humanos, sino que también reduce considerablemente los tiempos de análisis, asegurando que la información se mantenga actualizada y disponible para su uso en tiempo real. Se cumplió el segundo objetivo específico del proyecto: desarrollar un proceso ETL funcional, que inicialmente no se había proyectado como automático, pero cuya automatización representó un valor agregado clave. Este avance impulsó la creación de una fuente exclusiva de datos a través del formulario web, lo cual no estaba contemplado al inicio del proyecto.

Este descubrimiento permitió establecer un puente directo con herramientas tecnológicas modernas para la limpieza, transformación y carga de información, integrando dichos datos eficientemente al sistema.

Adicionalmente, la incorporación de notificaciones automáticas por correo fortaleció el monitoreo del sistema, permitiendo hacer seguimiento oportuno a posibles incidentes que

comprometan la integridad de la información. En conjunto, estos resultados mejoran la confiabilidad del sistema, fortalecen la toma de decisiones y dejan preparada la infraestructura para futuras ampliaciones del proyecto.

Resultado 3: Desarrollo de interfaces de consulta y visualización en Power BI, como parte del tercer objetivo del proyecto, se implementaron mecanismos de consulta y visualización interactiva de datos que permiten interpretar de forma estratégica la información almacenada en la base de datos del DirectorioUsme.com. Para ello, se desarrollaron consultas estructuradas en MySQL, así como vistas especializadas que consolidan datos clave sobre registros comerciales, categorías, subcategorías, ubicación geográfica, etiquetas y horarios.

Estas consultas fueron integradas con la herramienta de visualización Power BI, estableciendo una conexión directa con la base de datos en MySQL Workbench. Esta integración permite traducir las consultas SQL en gráficos dinámicos, generando paneles que reflejan métricas operativas y de rendimiento casi en tiempo real (ver Figura 88). Gracias a esta visualización, se pueden analizar variables como registros recientes, distribución por barrios, métodos de pago más comunes, visitas a páginas y comportamiento de búsqueda orgánica en el sitio, entre muchas más métricas. El sistema permite filtrar y consultar información por año, mes, día; ofreciendo una interacción robusta para monitorear la evolución de la plataforma, evaluar campañas de posicionamiento y ajustar decisiones comerciales en función del comportamiento de los usuarios. Este desarrollo representa un avance significativo respecto al estado inicial del sistema, donde no existían mecanismos automáticos de consulta ni representación gráfica de los datos. Con los dashboards diseñados y la capacidad de análisis por múltiples dimensiones, se fortalece el proceso de toma de decisiones basado en evidencia, se mejora el control interno y se incrementa la capacidad de respuesta frente a cambios en su entorno digital.

Conclusiones

El desarrollo de este proyecto fue un reto técnico y estratégico, así como una experiencia integral que permitió consolidar habilidades técnicas en todo el ciclo de vida de los datos, desde su captura hasta su análisis. Enfrentar un entorno real con información dispersa y no estructurada exigió aplicar con rigor los principios de limpieza, normalización y modelado relacional.

Uno de los principales aprendizajes fue la comprensión del valor de una arquitectura modular. Separar el diseño del modelo de datos, los flujos ETL y las visualizaciones no solo facilitó el desarrollo, sino que también garantizó la escalabilidad y el mantenimiento del sistema. Esta planificación fue clave para implementar una solución funcional en un entorno local, utilizando tecnologías de código abierto como Python, MySQL y Power BI.

La automatización mediante el programador de tareas de Windows y el uso de procedimientos almacenados permitió garantizar la continuidad operativa diaria sin necesidad de intervención manual, lo que representa un gran paso hacia la eficiencia y profesionalización de los procesos internos del sitio DirectorioUsme.com. Además, la integración con APIs de Google fue una valiosa curva de aprendizaje en aspectos como autenticación segura, transformación de datos en formatos JSON y extracción de métricas clave para el análisis digital.

Uno de los logros más valiosos fue implementar una solución automatizada y escalable, capaz de funcionar diariamente sin intervención humana, demostrando que un entorno de bajo presupuesto puede alcanzar niveles de eficiencia comparables con soluciones profesionales. Además, fortaleció de manera profunda competencias en análisis de datos, programación, diseño de bases de datos, visualización y documentación técnica.

Asimismo, dejó como resultado una solución concreta y funcional que no solo resuelve una necesidad operativa actual, sino que también sienta las bases para futuros desarrollos, como

la implementación de roles de usuario, monitoreo de comportamiento en tiempo real y análisis predictivo.

Finalmente, este trabajo representa una contribución tangible al ecosistema digital comunitario, demostrando que la Ciencia de Datos puede ser aplicada eficazmente en contextos locales para generar impacto, eficiencia y conocimiento estratégico.

Recomendaciones

Automatizar las copias de seguridad y pruebas de restauración realizando backups incrementales diarios o completos semanales, tanto de la base de datos como de los scripts ETL y de los informes de Power BI.

Implementar un monitoreo y alertas proactivas al usar herramientas como CloudWatch (AWS), Stackdriver (GCP) o Azure Monitor para supervisar métricas de CPU, memoria, IOPS y conexiones de la instancia MySQL. Configurar alertas (por ejemplo, Slack o correo) para picos de latencia, fallos de ETL, espacio en disco bajo o errores de consulta.

Optimizar el rendimiento de la base de datos considerando particionar tablas muy grandes (datos históricos) por rangos de fecha para mejorar tiempos de respuesta.

Mantener los scripts en un repositorio Git (GitHub, GitLab), con tags para cada release del sistema.

Cumplimiento normativo validando que la configuración cumpla con la Ley 1581 de 2012 (Colombia) y GDPR: manejo de datos personales, políticas de retención y borrado.

Capacitar y preparar un manual de usuario (técnico) para el equipo, con diagramas, flujos ETL, procedimientos y pasos de solución de incidentes. Dejar videos como sesión de entrenamiento o workshop para que los responsables entiendan cómo monitorear, depurar y evolucionar el sistema.

Referencias Bibliográficas

Brzozowska, J., Pizoń, J., Baytikenova, G., Gola, A., Zakimova, A., & Piotrowska, K. (2023). Data engineering in CRISP-DM process production data–case study. *Applied Computer Science, 19*(3).

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (1999). The CRISP-DM user guide. *4th CRISP-DM SIG Workshop in Brussels in March, 1999*.

Conde Ramírez, D. (2022). *Diseño de un framework de análisis de datos abiertos mediante un proceso ETL*.

Cortina, V. G. (2015). Aplicación de la metodología crisp-dm a un proyecto de minería de datos en el entorno universitario. *Universidad Carlos III de Madrid*.

De Mendıvil, J. R. G., & Las Encinas, E. (s. f.). *Lectura [2]: Tipos de datos*.

Encalada Garcia, D. A. (2025). *Diseño de un marco de trabajo para la implementación de procesos ETL* [B.S. thesis].

Espinosa-Zúñiga, J. J. (2020). Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública. *Ingeniería, investigación y tecnología*, *21*(1).

GALINDO, A. E. C. (s. f.). *ORIGEN DEL HABEAS DATA*.

Giménez, J. A. (2019). Buenas prácticas en el diseño de bases de datos. *ARANDU UTIC*, *6*(1), 193-210.

López Espinoza, A. I. (2024). *Nuevo modelo de datos e implementación de subflujos en proceso ETL para aumentar la eficiencia operativa en NeoSoft*.

Lopez-Nunez, J. (s. f.). *Fundamentos bases de datos relacionales: Conceptos básicos para su comprensión.*

Machuca Vivar, S. A., Vinueza Ochoa, N. V., Sampedro Guamán, C. R., & Santillán Molina, A. L. (2022). Habeas data y protección de datos personales en la gestión de las bases de datos. *Revista Universidad y Sociedad*, *14*(2), 244-251.

Marrero, L., Olsowy, V., Thomas, P. J., Delía, L. N., Tesone, F., Fernández Sosa, J., & Pesado, P. M. (2019). Un estudio comparativo de bases de datos relacionales y bases de datos nosql. *XXV Congreso Argentino de Ciencias de la Computación (CACIC)(Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019)*.

Orozco, F., Guaygua, S., López Villacis, D. H., Muñoz, F., & Urquía, M. L. (2021). Vinculación de datos administrativos y su utilidad en salud pública: El caso de Ecuador. *Revista Panamericana de Salud Pública*, *45*, e9.

Pacheco Castañeda, M. (2019). *Lenguaje SQL Avanzado*.

Palacios Martel, M. J. (2019). *Sistema ETL para la mejora en el movimiento de información entre servidores no comunicados en el Data Warehouse Mis de Financiera Confianza SAA-2019*.

Ramírez González, C. C. (s. f.). *Modelo de Implementación, estrategia de Analítica de Datos como soporte de las funciones de IVC de la SDEGC*.

Rawat, B., Purnama, S., & others. (2021). MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung. *International Journal of Cyber and IT Service Management*, *1*(2), 173-179.

Rivera Resina, F. J. & others. (2018). *Aplicación de Busines Intelligence en una pequeña empresa mediante el uso de Power Bi*.

Sánchez, A. A. C. (2020). Implementación de la arquitectura de datos usando el modelo relacional y MySQL Community Edition para el diseño de la base de datos del sistema de capellanía UM. *Anuario de Investigación UM*, *1*(1), 1-10.

Soberón, L., & Jesús, J. (2020). *Análisis comparativo de sistemas gestores de bases de datos postgresql y mysql en procesos crud*.

Torres, S. L. (2021). Componente de revisión de estándar de arquitectura de datos para el gestor de bases de datos SQLite. *Innovación y Software*, *2*(1), 20-32.

Treviño, R., Rivera, F., & Garza, J. (2020). *La analítica de datos como ventaja competitiva en las organizaciones. VinculaTégica, 6 (2), 1063–1074*.

Urtiaga, G. G. (2020). *Administrar MySQL y MariaDB: Aprende a administrar MySQL y MariaDB fácilmente*. AprendeIT.

Valverde, V., Portalanza, N., Mora, P., & others. (2019). Análisis descriptivo de base de datos relacional y no relacional. *Revista Atlante: Cuadernos de Educación y Desarrollo*, *3*.