Proyecto de Inversión Gastronómica y Afines en Mercado De Estados Unidos

Integrantes

Víctor Vargas
Guillermo del Rio
Michael Martinez Chinchilla
Julian Scarpeccio
Benjamin Zelaya

Índice

1. Introducción

Podemos observar el trabajo realizado en el segundo sprint, el cual consiste en la creación y optimización del pipeline de datos (ETL), Asimismo, se pondrá en marcha la implementación de un datawarehouse automatizado, acompañado de la carga incremental. El equipo de trabajo ha demostrado una sólida capacidad de organización y colaboración para abordar las demandas de nuestra propuesta de inversión para nuestro cliente.

2. Objetivos

El objetivo principal del segundo sprint radica en la automatización completa del pipeline de datos. Esta automatización conlleva la transformación de los datos en formatos adecuados, seguida de su almacenamiento en un datawarehouse diseñado con precisión para facilitar análisis posteriores.

3. Arquitectura General

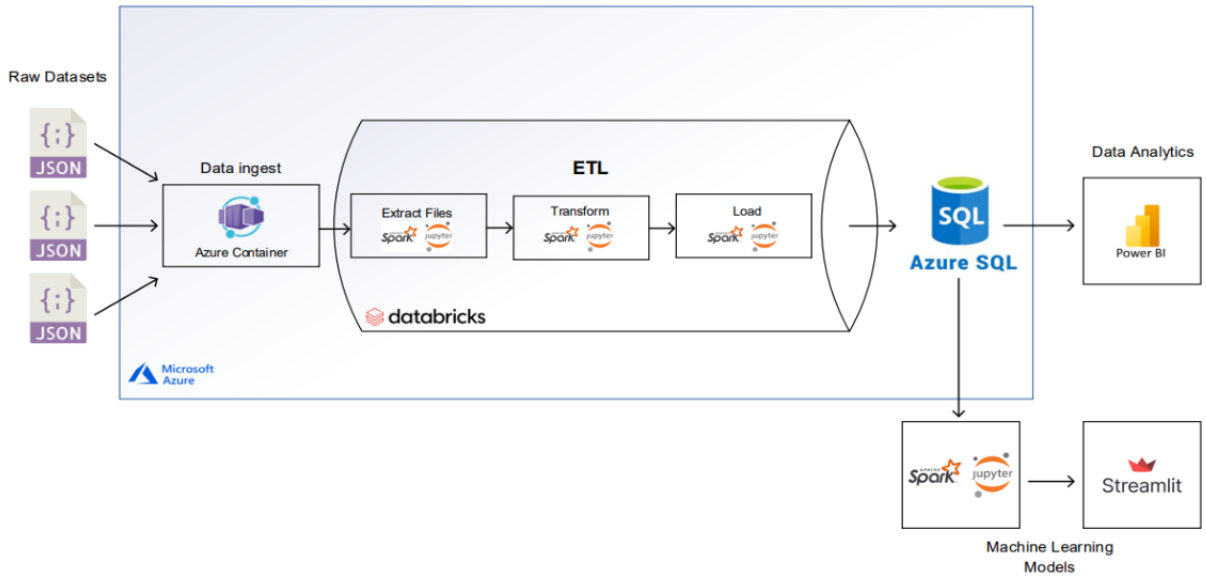

Figura 1:

3.1. Ingesta de Datos

Los datos entregados por la empresa y extraídos por nuestro equipo mediante api y web scraping se descargaron y son almacenados de manera temporal en el localhost de nuestra máquina.

Dado que trabajaremos sobre el esquema de Microsoft Azure se creará un contenedor donde se almacenarán los datasets sin procesar en la nube. Para esto, fue necesario crear una cuenta de trabajo en el portal de Azure. En dicha cuenta se crea un grupo de recursos donde incluimos una cuenta de almacenamiento con un contenedor.

3.2. Conexión con Databricks

Una vez almacenados los datasets en el contenedor de Azure se procede a realizar la conexión con Databricks, nuestro lugar de trabajo principal.

En el grupo de recursos previamente creado se añade un workspace de Databricks. Ahı se creará un cluster que permite computar nuestros datos (Single Node 10.4 LTS Apache Spark 14 GB Memory, 4 Cores), el criterio de selección es en base al alcance de nuestros recursos.

Dentro de Databricks creamos un Notebook y lo conectamos con el cluster. En dicho Notebook establecemos las variables necesarias para la conexión con el contenedor.

3.3. ETL

Se realizará todo el proceso de extracción, transformación y carga de los datos hacia el data warehouse.

3.4. Conexión con sql database

Creada la SQL Database de Azure se realizará la conexión con Databricks por medio del protocolo jdbc.

3.5. Conexión con Power BI

La conexión se realiza mediante el conector de Azure SQL Database de PowerBI. Se ingresan las credenciales del servidor de base de datos y se cargan los datos ya sea por Direct Query o Import Data.

Una vez que los datos se encuentran en la base de datos, se pueden analizar y visualizar utilizando Power BI. Esto permite identificar tendencias, patrones y obtener información valiosa para la toma de decisiones informadas.

3.6. Modelo de machine learning y streamlit

Se creará un modelo de Machine Learning en Python. Este modelo se implementará en una aplicación interactiva utilizando Streamlit, lo que permitirá a los usuarios utilizar el modelo y obtener recomendaciones en tiempo real.

3.7. Grupo de recursos en azure

Grupo de recursos en azure a utilizar.

Podemos observar los recursos creados en la cuenta de azure, en el cual tenemos.

- **datosproyecto:** cuenta de almacenamiento donde se crea el contenedor y se cargan los datos
- **keyvaultproyecto:** es nuestro almacenamiento de claves que contendrá todos nuestros secretos para que no sean visibles nuestras contraseñas en el documento.
- **databricks_proyecto:** nuestro lugar principal de trabajo en el cual realizaremos nuestras transformaciones y carga a la base de datos con pyspark.
- **servidor_reviews:** es el servidor de la base de datos en el cual es de tipo sql server.

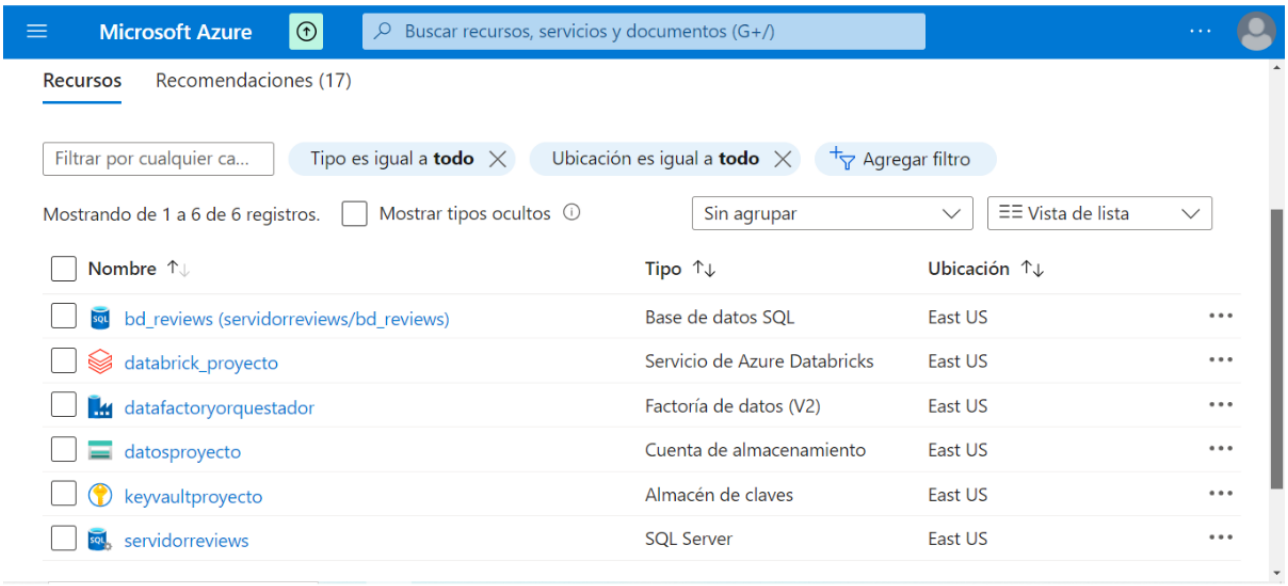

Figura 2:

- **bd_reviews:** es la base de datos de tipo sql en el cual contendrá nuestros datos limpios y organizados.
- **datafactoryorquestador:** es nuestro orquestador de datos para realizar la automatización y carga incremental.

3.8. Arquitectura En Azure

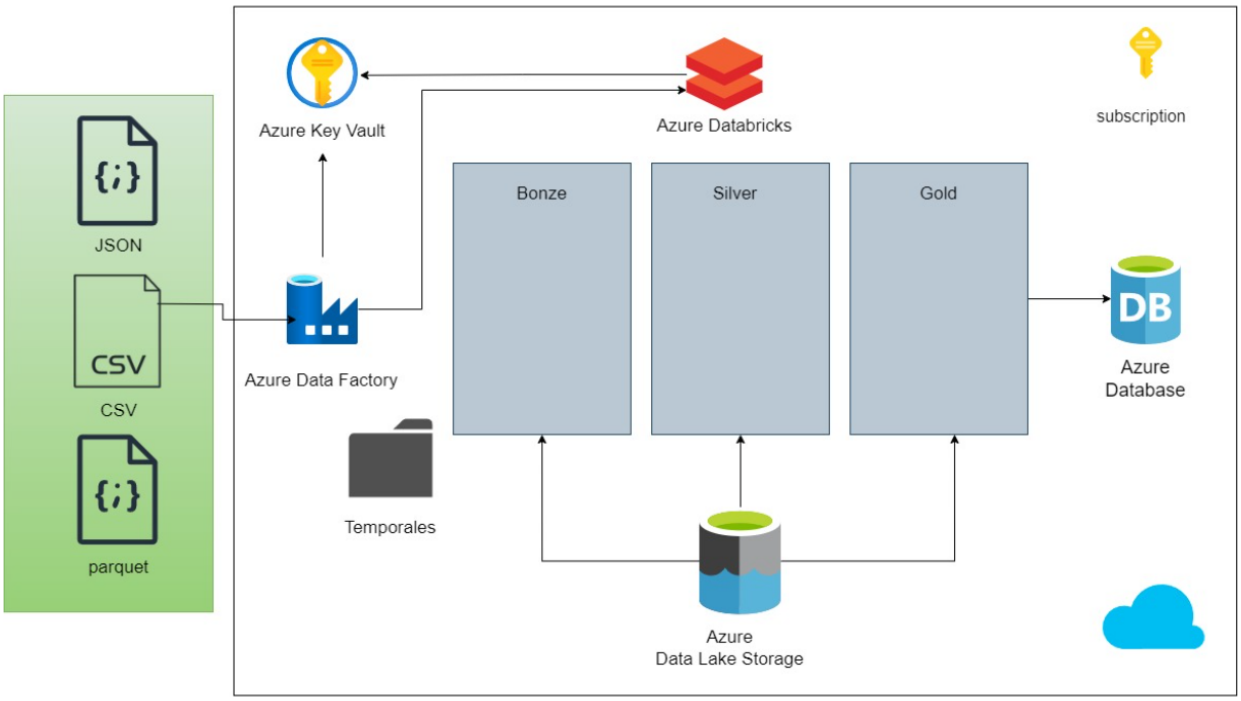

Figura 3: El flujo de datos utilizando las tecnologías de azure

Azure data lake storage: dentro del contenedor creamos cuatro folders que son.

- **Temporal:** se cargan los datos en crudo.

- **Bronze:** se hace una copia de los datos en temporal pero en formato delta para así mejorar el procesamiento de cómputo de databricks.
- **Silver:** tenemos los datos ya transformados y limpios.
- **Gold:** tenemos datos finales con agrupaciones que serán cargados a la base de datos.

3.9. Azure Data Factory y Databricks (Proceso ETL Automatizado):

El proceso de Extracción, Transformación y Carga (ETL) se ejecuta mediante el lenguaje de programación Pyspark en databrick, y se complementa con la utilización de Azure Data Factory como un orquestador de datos utilizando cada notebook en forma de tarea y secuencial. Esta combinación permite automatizar el flujo de datos y realizar una carga incremental de manera altamente eficiente.

3.10. SQL DataBase (Almacenamiento):

Los datos procesados se almacenan en una base de datos SQL, donde la estrategia de almacenamiento se divide en dos etapas. En la primera carga, se almacena el 50 % de los datos, y luego se procede con una segunda carga incremental para completar la base de datos. Esta metodología permite una gestión eficiente y escalable de la información procesada.

4. ETL

El proceso de Etl se dividió en dos partes una local y la otra en la nube utilizando Azure databricks.

ETL_01_local :

1. **extracción:** se descargaron los datos dados por la empresa en el cual son los de google maps y yelp, también se obtuvieron datos de fuentes externas que fueron los de estados que nos muestra la cantidad de población que hay en cada estado y el de inversión de franquicias que nos da a conocer la mínima y máxima inversión que se realiza a cada franquicia.
2. **transformación:**
   - **google maps:** se consolidó un dataset resultante que fue por la unión de dos dataset también consolidados que son estados y sitios_unidos que posteriormente se filtró por la categoría de restaurantes que es el tipo de negocio que se estudiará.

     carpeta reviews-estados se consolidó un dataset resultante llamado estado en el cual se filtraron dos columnas a utilizar gmad_id y estado. carpeta metadata-sitios se consolidó un dataset resultante llamado sitios_unidos en el cual se concatenaron los archivos .json en el cual nos muestra información importante de cada local comercial como las reviews, ubicación, estado, url, etc.

     Del dataset sitios unidos se filtraron cuatro columnas a utilizar çategory",."vg_rating","gmap_id" luego concatenarlo con el dataset estado por medio de su id y consolidar un dataset resultante llamado google y por último se filtró la categoría a utilizar que son los restaurantes.

- **yelp:** se consolidó un dataset resultante llamado yelp en el cual se filtró por tres columnas a utilizar que son comentarios, latitud y longitud en el cual nos permite tener los comentarios de cada review dada y la ubicación de dicho local para posteriormente relacionarlo con el dataset de google.
- **población:** se consolidó el dataset población filtrando la columna estado y cantidad de población posteriormente realizando una columna nueva llamada categoría de densidad en el cual se divide por baja, media y alta la población de cada estado.
- **Inversión de franquicias:** se observó el dataset para un posterior análisis.

ETL_02_azure:

se dividió los procesos en cuadernos de databricks:

1. **Ingesta:** se cargaron de forma manual los cuatro dataset obtenidos del etl_01_local al contenedor de azure que está dividido en cuatro folders que son temporal, bronze, silver y gold, en el que se cargaran estos dataset google, yelp, inversion_franquicias y poblacion en formato csv cada uno en el folder temporal.
2. **montaje:** se definen las variables para hacer la conexión hacia el azure data lake y su contenedor, se realiza el montaje y se listan las rutas de los folder que se trabajarán.
3. **extracción:** se definen las librerías y rutas a utilizar, se estructura cada dataset con sus nombres de columnas y tipo de datos para luego cargar y visualizar los datos de la capa temporal posteriormente se cargan estos datos a la capa bronze que son una copia de la capa temporal en formato delta para aprovechar los recursos de databricks y procese los datos con mayor rapidez.
4. **transformación:**
   - **población:** se carga el dataset de la capa bronze, se realiza una columna nueva llamada id_estado que será nuestra llave primaria, se eliminan registros no necesarios y se carga el dataset limpio a la capa silver cambiando de nombre estado.
   - **franquicias:** se carga el dataset de la capa bronze, se eliminan duplicados, se realiza la columna id_franquicia que será su llave primaria y guardó en el folder silver.
   - **google:** se carga el dataset de la capa bronze, se eliminan columnas innecesarias, se cambian registros con caracteres especiales, se crea columna id_google que sera su llave primaria y guardo el dataset ya limpio en el folder silver.
   - **yelp:** cargo del folder bronce, elimino columnas y duplicados que son innecesarios y guardó el dataset ya limpio en silver.
   - **Reviews:** se carga los dataset google y yelp de la capa silver y se realizan un join por medio de sus columnas latitud y longitud para formar un solo archivo llamado reviews que será nuestra tabla de hechos en el cual contiene toda nuestra información valiosa; posteriormente los relaciono con los dataset de estados y franquicias para añadir su llave foránea y poder relacionar los tres dataset finales, para finalizar también es guardado en silver.

     En conclusión se formaron tres dataset finales que son llamados reviews, estados y franquicias, estos dataset son particionados en un 50 % cada uno y son creados 6 dataset para posteriormente cargados en el folder gold.
5. **carga:** se cargan los 6 dataset finales de la capa gold en el cual se realizará una primera carga a la base de datos con tres dataset y la segunda carga con los otros tres dataset y

así realizar la carga incremental.

previo a esto se realizó la configuración de conexión entre databricks y la base de datos sql para cargar estos dataframe por medio del protocolo jdbc.

5. Diagrama Entidad Relación

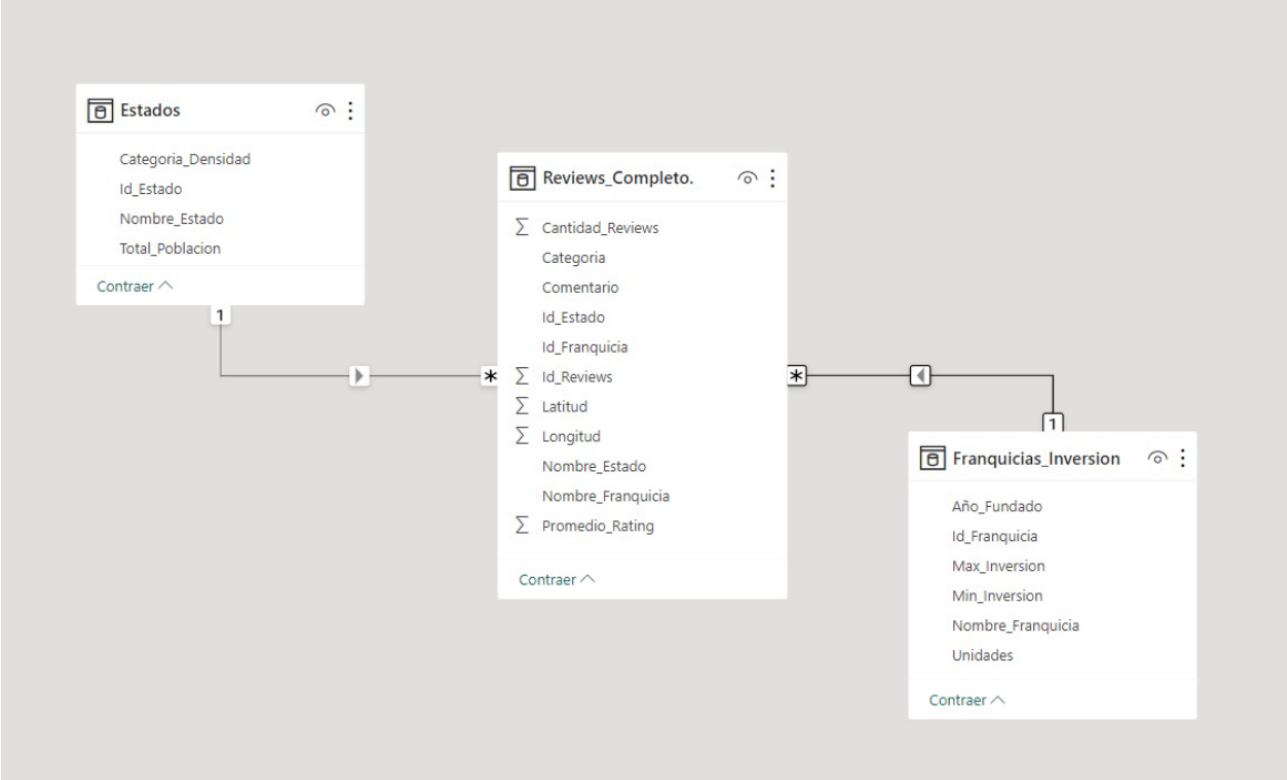

Figura 4:

Podemos examinar las relaciones entre nuestras tablas resultantes en nuestro modelo entidad-relación. En este contexto, la tabla Reviews_completo se convierte en nuestra tabla de hechos, que contiene información crítica sobre el negocio. Esta tabla de hechos tiene una clave primaria llamada id_reviews y utiliza claves foráneas, como id_franquicia e id_estado , para establecer conexiones con otras tablas de dimensiones, específicamente Estados y Franquicias_Inversion .

Estas tablas de dimensiones proporcionan información detallada sobre cada estado y franquicia, lo que enriquece nuestros datos y permite construir un sólido modelo entidad-relación que refleja de manera efectiva la estructura y relaciones en nuestros datos.

6. Diccionario de datos

Contamos con tres datasets de datos resultantes.

- **Estados**
- **Reviews_Completo**
- **Franquicias_Inversion**

Descripción

- **Dataset: Estados**
  - **Id_Estado:** dentificador único del estado -> pk tipo entero
  - **Nombre_Estado:** Nombre del estado -> tipo string.
  - **Total_Poblacion:** Total de población en el estado -> tipo entero.
  - **Categoria_Densidad:** Categoría de densidad del estado, sea este Alta, Media o Baja -> tipo string.
- **Dataset: Reviews_Completo**
  - **Id_Reviews:** Identificador único de la revisión -> pk tipo entero.
  - **Nombre_Franquicia:** Nombre de la franquicia a la que se refiere la revisión -> tipo string.
  - **Latitud:** Coordenada de latitud -> tipo float.
  - **Longitud:** Coordenada de longitud -> tipo float.
  - **Categoria:** Categoría del restaurante -> tipo string.
  - **Promedio_Rating:** Promedio de calificación de la revisión -> tipo float.
  - **Cantidad_Reviews:** Cantidad de revisiones para la franquicia. -> tipo float.
  - **Comentario:** Comentario de la revisión -> tipo string.
  - **Id_Estado:** Identificador único del estado -> fk tipo entero
  - **Nombre_Estado:** Nombre del estado al que pertenece el restaurante.
  - **Id_Franquicia:** Identificador único de la franquicia a la que se refiere la revisión -> fk tipo entero.
- **Dataset: Franquicias_Inversion**
  - **Id_Franquicia:** Identificador único de la franquicia -> pk tipo entero.
  - **Nombre_Franquicia:** Nombre de la franquicia -> tipo string
  - **Min_Inversion:** Inversión mínima requerida para la franquicia -> tipo entero.
  - **Máx_Inversion:** Inversión máxima requerida para la franquicia -> tipo entero.
  - **Año_Fundado:** Año en que se fundó la franquicia -> tipo entero.
  - **Unidades:** Número de unidades de la franquicia en operación -> tipo entero.

Este diccionario de datos ayudará a comprender la estructura de cada conjunto de datos y las columnas que contiene, lo que facilitará el análisis y la manipulación de los datos en cada conjunto.

7. KPIs

Los siguientes KPIs se utilizarán para evaluar el éxito de nuestras soluciones y la mejor opción de inversión de franquicia para nuestro cliente:

KPI 1: Tasa de Satisfacción del Cliente.

Descripción: Este KPI mide el porcentaje de clientes satisfechos en función del rating de opiniones recopiladas en plataformas como Yelp y Google Maps.

Calificación: por encima de un umbral específico de puntuación de Rating ≥ 4 .

Fórmula: Tasa de Satisfacción del Cliente = (Número de Rating/ Total de Rating) x 100

KPI 2: Cantidad de Sucursales por Conglomerado de Estados.

Descripción: Permite identificar cuantas sucursales tendremos por conglomerado de estados.

KPI 3: Porcentaje de Restaurantes con Alta Calificación.

Descripción: Este KPI muestra el porcentaje de restaurantes que tienen una calificación por encima de un umbral específico (por ejemplo, 4 estrellas).

fórmula: Porcentaje de Restaurantes con Alta Calificación = (Número de Restaurantes con Rating ¿= Umbral) / Total de Restaurantes x 100

KPI 4: Top 5 Franquicias por Conglomerado de Estados:

descripción: identificar las mejores franquicias por conglomerado de estados.

Después de haber compartido los indicadores clave de rendimiento (KPIs) que describen la situación actual, ahora introducimos dos KPIs que serán fundamentales para evaluar el éxito de la inversión una vez que haya transcurrido un tiempo desde su ejecución.

KPI 5: Comparación de Satisfacción del Cliente Promedio por Estado con el Nivel de Satisfacción de la Franquicia Elegida.

Descripción: Este indicador se establece con el propósito de evaluar el nivel de satisfacción del cliente en la franquicia elegida por el inversor y compararlo con la satisfacción promedio de los clientes dentro del estado en el que se realizó la inversión.

Si el nivel Promedio de satisfacción de nuestra Franquicia está por encima del promedio del estado significa que estamos por encima de la media y que mantenemos la calidad y servicio a nuestros clientes, lo cual nos indica un buen desempeño.

KPI 6: Comparación del Nivel de Satisfacción del Cliente en Franquicias Seleccionadas frente al Promedio Nacional.

Descripción: Este indicador de desempeño tiene como objetivo evaluar la satisfacción de los clientes en la franquicia que se ha seleccionado por el inversor, y contrastar con la satisfacción promedio de los usuarios en dicha franquicia a nivel nacional.

8. Análisis Exploratorio

En el proceso de realizar el Análisis Exploratorio de Datos (EDA) realizado en los conjuntos de datos df_estado, Franquicias_inversion.csv y Reviews_Completos nos proporciono una base solida para desarrollar un modelo de machine learning de sistema de recomendacion de inversion por franquicias, estado, categorias y monto a invertir. A traves del EDA, hemos examinado la poblacion y densidad demografica en los estados, explorado las caracteristicas clave de las franquicias, como inversiones y unidades, y profundizado en las revisiones de franquicias para entender la satisfaccion del cliente.

En este análisis exploratorio de datos, hemos llevado a cabo un estudio detallado de la densidad demográfica en los diferentes estados.

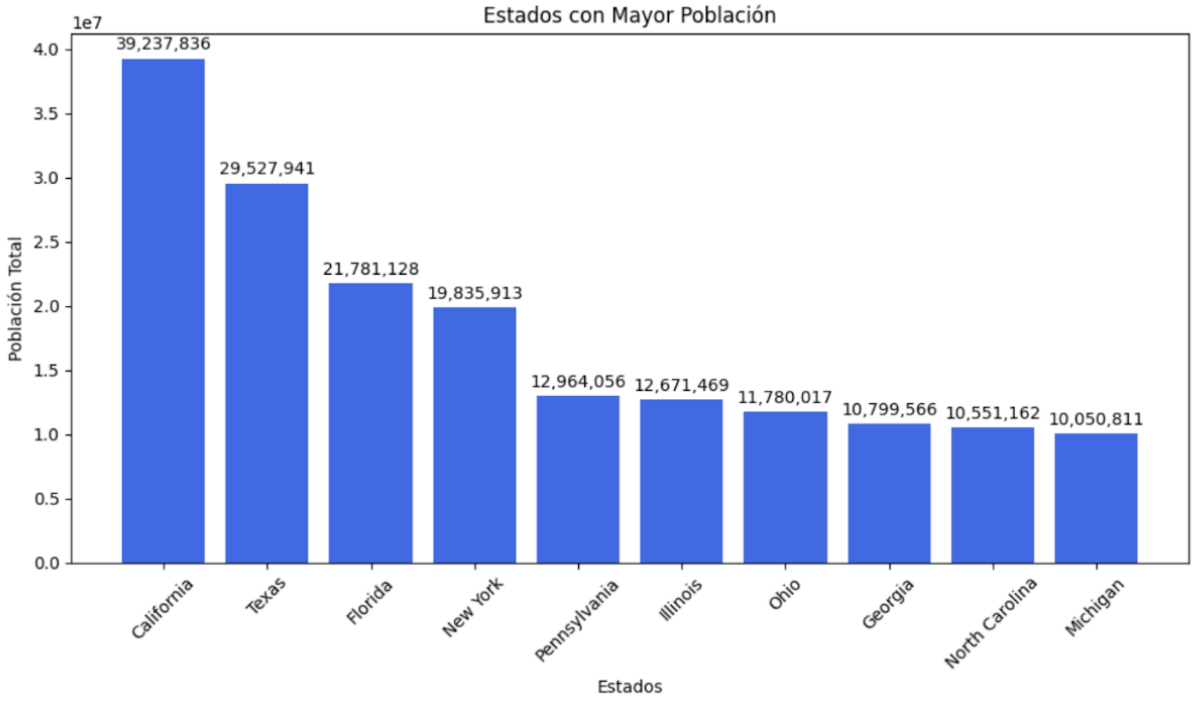

Figura 5:

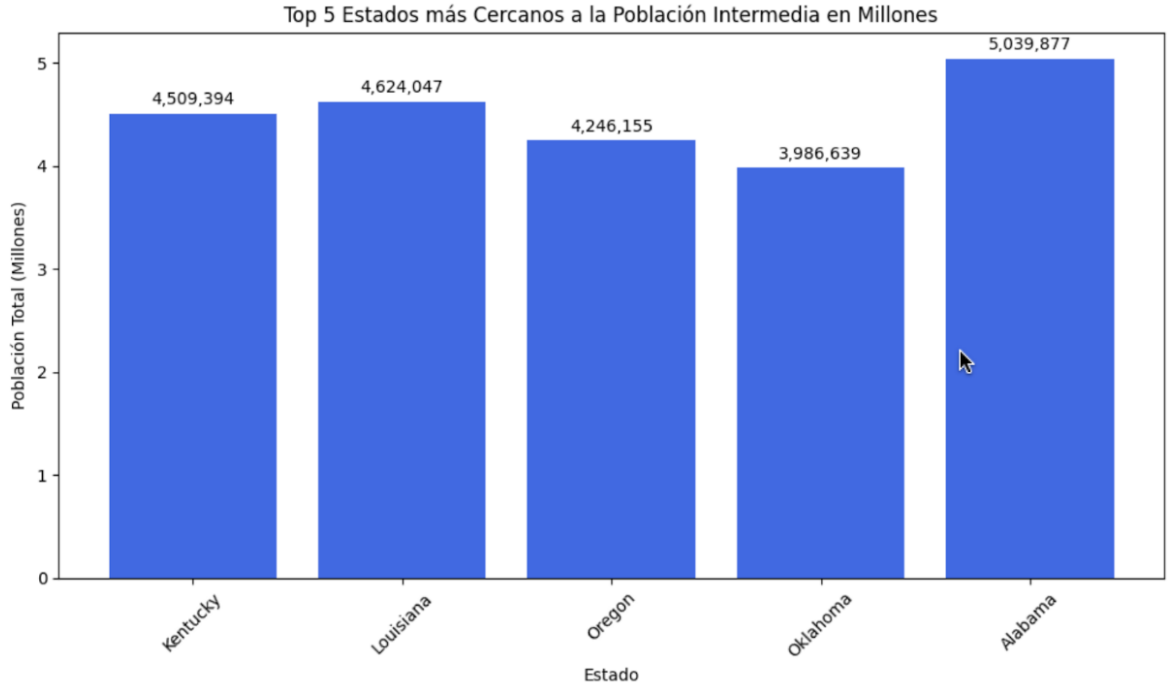

Figura 6:

El objetivo principal de esta investigación es comprender y visualizar cómo se distribuye la población en los estados y destacar aquellos con una densidad demográfica particularmente baja. Para lograr esto, comenzamos por recopilar datos de población por estado y luego calculamos la densidad demográfica, que es la relación entre la población y el área geográfica de cada estado. Luego, identificamos los estados con la densidad demográfica más baja y presentamos estos resultados visualmente mediante un gráfico de barras.

En dicho gráfico, cada barra representa un estado, y la altura de la barra muestra la población,

lo que nos permite destacar de manera efectiva los estados con menos habitantes. Además, para facilitar la interpretación de la información, hemos añadido etiquetas con la cantidad exacta de habitantes encima de cada barra en el gráfico. Esto proporciona una comprensión instantánea de las diferencias en la densidad demográfica entre los estados.

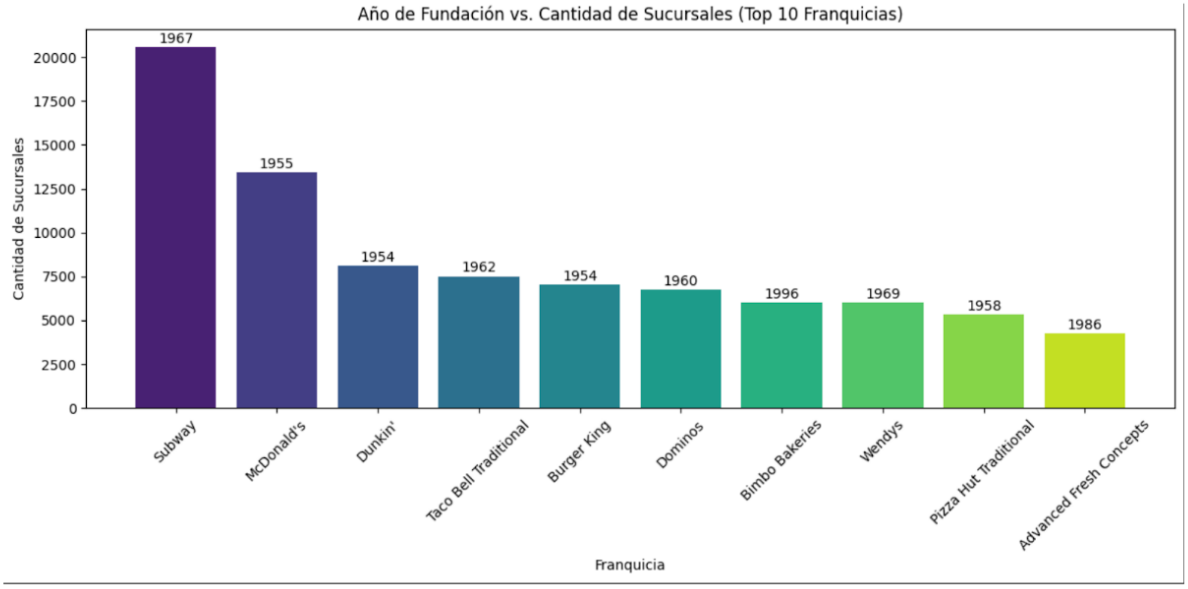

Figura 7:

Al analizar el gráfico que compara el año de fundación con la cantidad de sucursales de las 10 franquicias más grandes, se pueden extraer varias conclusiones significativas.

En primer lugar, se destaca que no existe una correlación lineal clara entre la antigüedad de una franquicia y su expansión. Aunque franquicias más antiguas muestran una presencia sólida en términos de sucursales, franquicias relativamente nuevas también ocupan posiciones destacadas. Este hecho sugiere que el éxito en la expansión de una franquicia no está determinado únicamente por su antigüedad, sino que factores como la estrategia de negocio, la demanda del mercado y la gestión eficiente pueden desempeñar roles fundamentales.

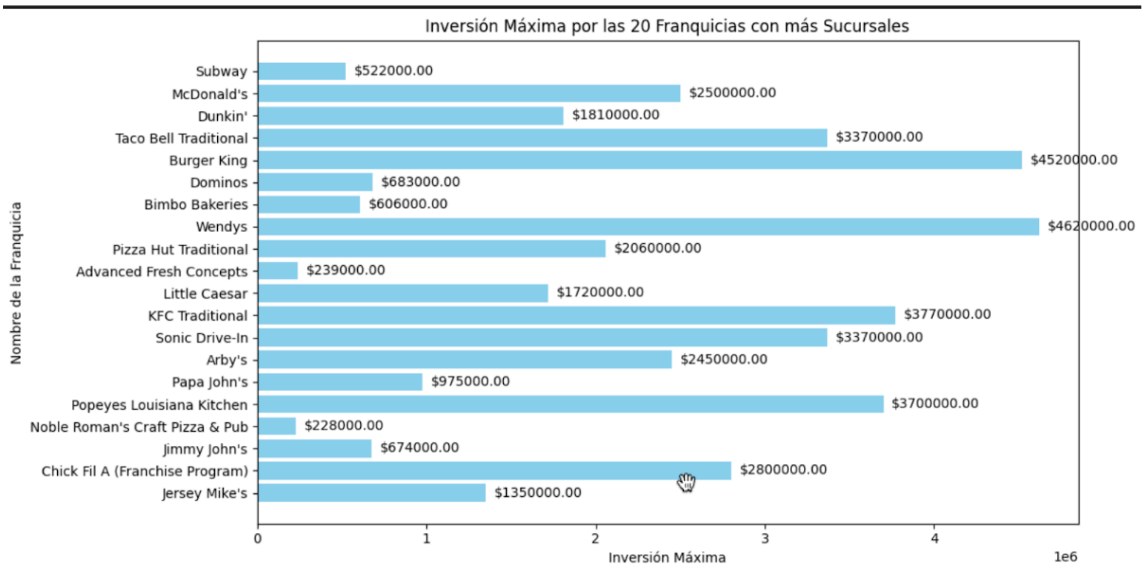

Figura 8:

En este gráfico de barras que representa las inversiones máximas requeridas por las 20 franquicias con mayor cantidad de unidades ofrece una visión reveladora de la diversidad en el panorama de inversión de franquicias en Estados Unidos. Observamos que estas franquicias líderes reflejan la amplitud de oportunidades disponibles para los inversores. Una tendencia interesante es que muchas de estas franquicias ofrecen inversiones relativamente asequibles en comparación con otras opciones en el mercado de franquicias.

Esta accesibilidad en términos de inversión ha contribuido a su crecimiento en términos de unidades y, en última instancia, a su éxito continuo. Por otro lado, también resalta la presencia de franquicias más grandes y establecidas que requieren inversiones más significativas. Esto proporciona una visión valiosa para nuestros clientes - inversores que buscan oportunidades dentro de su presupuesto como para aquellos que desean considerar opciones de inversión de mayor envergadura en el competitivo mundo de las franquicia.

Utilizando esta informacion que se encuentra con mas detalle en la carpeta EDA del sprint_2, hemos definido indicadores clave de rendimiento (KPIs) que incluyen la Tasa de Satisfaccion del Cliente y la Cantidad de Sucursales por Conglomerado de Estados. Ademas, hemos establecido KPIs adicionales como la Comparacion de Satisfaccion del Cliente Promedio por Estado con el Nivel de Satisfaccion de la Franquicia Elegida y la Comparacion del Nivel de Satisfaccion del Cliente en Franquicias Seleccionadas frente al Promedio Nacional. Estos KPIs seran fundamentales para evaluar el exito de la inversion una vez que haya transcurrido un tiempo desde su ejecucion, brindándonos informacion valiosa para tomar decisiones informadas en el ambito de las franquicias y las inversiones.

9. Tareas por realizar

9.1. Automatizar Data Warehouse

Para lograr la automatización del data warehouse, implementaremos el uso de azure data factory. En el cual se relacionarán los cuadernos realizados en databricks, configuraremos un flujo de trabajo que facilitará la ejecución de nuestro proceso ETL, asegurando así la transformación y carga eficiente de los datos depurados en sql database.

9.2. Carga incremental al Data Warehouse

data factory realiza una carga incremental al emplear un enfoque inteligente para identificar y procesar solo los nuevos datos o los que han cambiado desde la última ejecución. Utilizando marcas de tiempo y metadatos, data factory compara los registros existentes con los nuevos datos, permitiendo una actualización eficiente de la base de datos sin tener que procesar nuevamente todo el conjunto de datos. Esto agiliza el proceso y optimiza el uso de recursos al reducir la carga de trabajo y el tiempo requerido para mantener la integridad y actualidad de la información en el sistema.