Proyecto de Data Engineering

Este proyecto de Data Engineering se llevó a cabo con el objetivo de manejar, transformar y visualizar datos de manera eficiente utilizando diversas herramientas y técnicas. A continuación, se detalla cada una de las fases principales del proyecto

Acerca del proyecto:

Herramientas usadas:

- Python
- Jupyter
- SQLAlchemy
- PostgreSQL
- PowerBI

About the data

- **FirstName**: El primer nombre del candidato
- **LastName**: El apellido del candidato
- **Email**: La dirección de correo electrónico del candidato
- **ApplicationDate**: La fecha en que el candidato aplicó para el trabajo
- **YOE**: Años de experiencia
- **Seniority**: La posición a la que el candidato está aplicando
- **Technology**: El área o campo al que el candidato está aplicando
- **Code Challenge Score**: Una puntuación de 0 a 10 para la prueba práctica
- **Technical Interview Score**: Una puntuación de 0 a 10 para la entrevista teórica o técnica

ejemplo del csv:

candidates.csv

First Name	Last Name	Email	Application Date	Country	YOE	Seniority	Technology	Code Challenge Score	Technical Interview Score
Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway	2	Intern	Data Engineer	3	3
Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama	10	Intern	Data Engineer	2	10
Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus	4	Mid-Level	Client Success	10	9
Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea	25	Trainee	QA Manual	7	1
Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar	13	Mid-Level	Social Media Community Management	9	7
Alec	Abbott	juanita_hansen@gmail.com	2019-08-17	Zimbabwe	8	Junior	Adobe Experience Manager	2	9
Allison	Jacobs	alba_rolfson27@yahoo.com	2018-05-18	Wallis and Futuna	19	Trainee	Sales	2	9
Nya	Skiles	madisen.zulauf@gmail.com	2021-12-09	Myanmar	1	Lead	Mulesoft	2	5
Mose	Lakin	dale_murazik@hotmail.com	2018-03-13	Italy	18	Lead	Social Media Community Management	7	10
Terrance	Zieme	dustin31@hotmail.com	2022-04-08	Timor-Leste	25	Lead	DevOps	2	0

1. Almacenamiento de Datos en PostgreSQL con SQLAlchemy

El proceso comenzó con la importación de un archivo CSV que contenía un conjunto de datos clave para el análisis. Este archivo se seleccionó cuidadosamente por su relevancia en el contexto del proyecto, y su contenido incluía varias columnas con información estructurada que debía ser almacenada de forma eficiente en una base

de datos. Para lograr esto, se utilizó SQLAlchemy, una potente biblioteca en Python que facilita la interacción con bases de datos relacionales.

Primero, se configuró una conexión a una base de datos PostgreSQL local, aprovechando la flexibilidad de SQLAlchemy para manejar la comunicación entre Python y la base de datos. Esta conexión se estableció utilizando un URI de conexión, que incluye detalles específicos como el tipo de base de datos, la ubicación del servidor, y las credenciales necesarias para acceder a la misma.

Con la conexión establecida, se procedió a definir las tablas en las que se almacenarían los datos del archivo CSV. SQLAlchemy permite definir estas tablas utilizando un enfoque de modelado orientado a objetos, donde cada clase de Python representa una tabla en la base de datos, y cada atributo de la clase corresponde a una columna en la tabla. Este método facilita la creación de esquemas de base de datos bien estructurados y asegura que las relaciones entre los diferentes conjuntos de datos sean claras y mantenidas correctamente.

Luego, se realizó la inserción de los datos. El archivo CSV se leyó utilizando pandas, una biblioteca ampliamente usada para la manipulación de datos en Python, y los datos se transformaron en objetos que SQLAlchemy puede manejar. Estos objetos fueron luego insertados en la base de datos de manera eficiente, aprovechando las capacidades de SQLAlchemy para manejar grandes volúmenes de datos. Este proceso no solo garantizó que los datos estuvieran organizados y accesibles, sino que también preparó el terreno para las siguientes fases del proyecto, donde la estructura de la base de datos sería crucial para el análisis y la visualización de la información.

2. Análisis Exploratorio de Datos(EDA)

Análisis Exploratorio de Datos (EDA)

Después de almacenar los datos en PostgreSQL, se procedió con un Análisis Exploratorio de Datos (EDA), una etapa crucial en el flujo de trabajo de ciencia de datos. El objetivo principal de esta fase fue comprender profundamente los datos y preparar el terreno para las etapas posteriores de modelado y análisis.

El EDA comenzó con una inspección minuciosa de las características estadísticas de los datos. Esto incluyó cálculos de métricas como la media, la mediana, la desviación estándar y la distribución de frecuencias. Se evaluaron las relaciones entre diferentes variables, buscando correlaciones que pudieran ser significativas para el análisis futuro.

Además, durante esta fase se identificaron y trataron valores atípicos, datos faltantes y posibles errores en el conjunto de datos. Esto fue esencial para asegurar

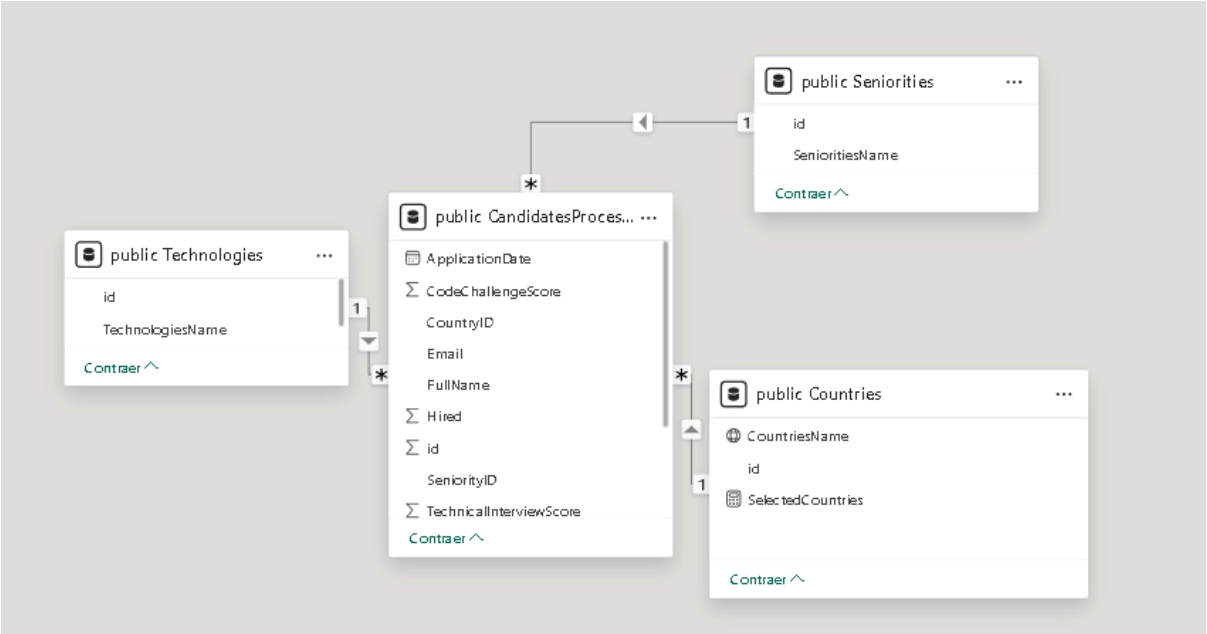
la calidad y la integridad de los datos antes de proceder con cualquier transformación o modelado.

Para complementar el análisis estadístico, se emplearon herramientas como pandas y matplotlib para visualizar los datos. Las visualizaciones, como histogramas, gráficos de dispersión y diagramas de caja, proporcionaron una visión clara de la distribución de los datos y ayudaron a identificar patrones, tendencias y anomalías de manera más intuitiva. Estas visualizaciones no solo facilitaron la comprensión de los datos, sino que también permitieron comunicar de manera efectiva los hallazgos a otras partes interesadas.

Finalmente, el EDA sirvió como base para planificar las transformaciones necesarias en los datos. Con una comprensión clara de la estructura y las características de los datos, se definieron las estrategias de normalización, agrupación y agregación que se aplicarían en la siguiente fase. Este enfoque permitió optimizar el uso de los datos en el modelo de estrella que se construiría posteriormente, asegurando que la estructura final de los datos fuera la más adecuada para el análisis y visualización en el dashboard de Power BI.

3. Transformación de Datos y Creación de Esquema Estrella

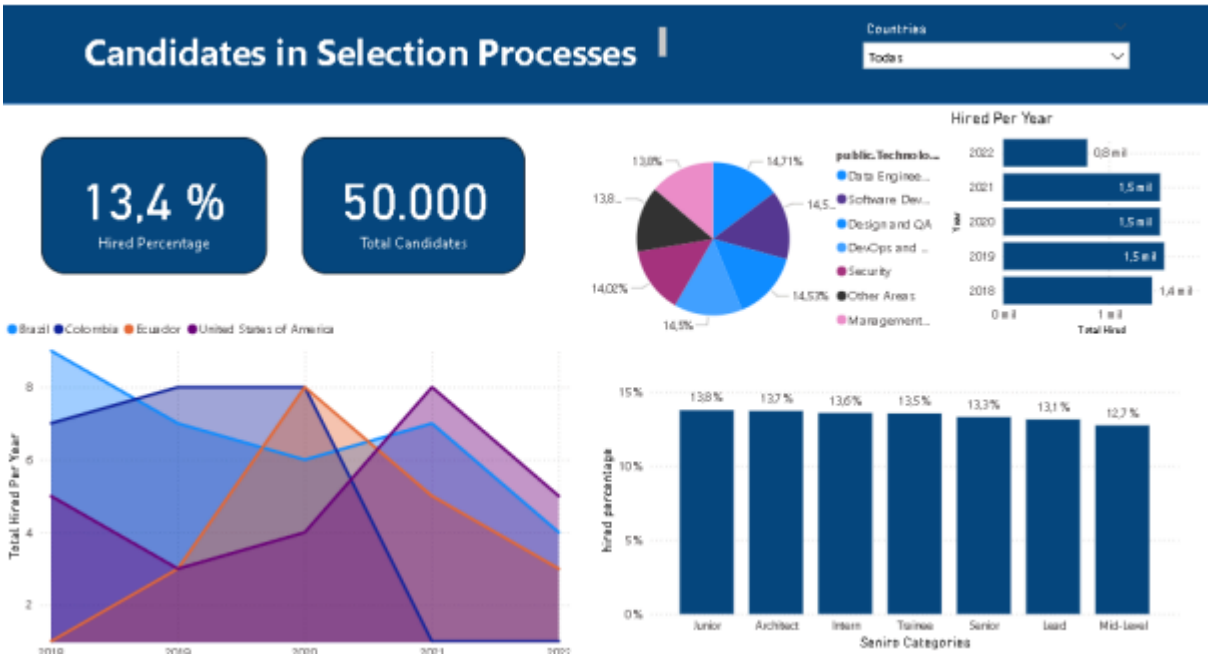
Con base en los resultados del EDA, se decidió transformar los datos a un modelo más eficiente y fácil de consultar, optando por un esquema de estrellas. Este tipo de modelo organizacional es muy utilizado en data warehouses para optimizar consultas analíticas. La transformación implicó la creación de tablas de hechos y dimensiones, donde las tablas de hechos almacenan eventos o transacciones, y las tablas de dimensiones contienen atributos descriptivos. Esta reestructuración mejoró significativamente el rendimiento en las consultas y facilitó la generación de reportes en etapas posteriores.



Por ejemplo, en el dashboard (como se muestra en la imagen), se puede observar un gráfico de líneas que compara el número total de candidatos contratados por país a lo largo de varios años. Esta visualización ofrece una perspectiva clara sobre la tendencia de contratación en diferentes países, lo que podría ser útil para identificar mercados emergentes o áreas que requieren atención. Además, el gráfico de pastel desglosa las contrataciones por áreas tecnológicas, permitiendo a los usuarios ver rápidamente cuáles son las áreas más demandadas.

4. Visualización de Datos con Power BI

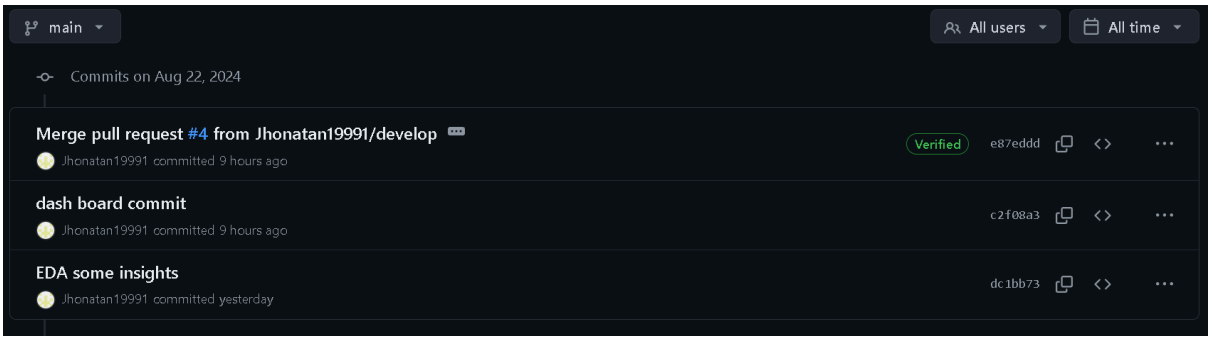
Finalmente, los datos transformados fueron importados a Power BI, donde se creó un dashboard interactivo. Este dashboard proporciona una visualización intuitiva de los datos, permitiendo a los usuarios finales explorar la información de manera dinámica. Power BI ofreció diversas opciones de visualización, como gráficos de barras, líneas de tiempo, tablas dinámicas, entre otros, que facilitaron la toma de decisiones basadas en los datos.



Por ejemplo, en el dashboard (como se muestra en la imagen), se puede observar un gráfico de líneas que compara el número total de candidatos contratados por país a lo largo de varios años. Esta visualización ofrece una perspectiva clara sobre la tendencia de contratación en diferentes países, lo que podría ser útil para identificar mercados emergentes o áreas que requieren atención. Además, el gráfico de pastel desglosa las contrataciones por áreas tecnológicas, permitiendo a los usuarios ver rápidamente cuáles son las áreas más demandadas.

5. Documentación y Repositorio en Git

Todo el proceso fue cuidadosamente documentado y subido a un repositorio en Git. Esta documentación incluyó detalles técnicos sobre la configuración de la base de datos, el código utilizado para la transformación de datos, y las configuraciones del dashboard en Power BI. Al mantener un registro en Git, se garantizó que el proyecto se guardará en los servidores de github y no solamente de forma local para no perder los procesos hechos de forma local.



para poder correr el proyecto en tu ambiente se debe de seguir el procedimiento hecho en el readme, el repositorio estaba distribuido de la siguiente forma:

- data: esta carpeta contiene el archivo candidates.csv utilizado en este proyecto.
- python_code: esta carpeta contiene la configuración de la base de datos y el archivo transform.py, que se utiliza para normalizar los datos.
- notebooks: esta carpeta contiene los cuadernos (notebooks) en este proyecto, en este caso, el análisis exploratorio de datos y los datos procesados.
- src: en esta carpeta hacemos los códigos de Python que utilizamos para conectarnos a la base de datos y crear los modelos sobre las tablas.

Conclusión

Este proyecto demostró un flujo de trabajo completo y eficaz en la ingeniería de datos, desde la importación de un archivo CSV hasta la creación de un dashboard interactivo en Power BI. Al utilizar SQLAlchemy y PostgreSQL para almacenar y gestionar los datos, se aseguró una estructura sólida para el Análisis Exploratorio de Datos (EDA), que permitió identificar patrones y planificar transformaciones. Finalmente, la visualización en Power BI brindó una forma intuitiva de explorar la información, facilitando la toma de decisiones. Todo el proceso fue documentado y subido a un repositorio de Git, mostrando una integración eficiente de herramientas y técnicas para convertir datos en información valiosa.