PRUEBA TÉCNICA INGENIERIA DE DATOS

Condiciones generales de la prueba:

- El aspirante tendrá máximo 2 días calendario para realizar la prueba y entregarla.
- Podrá elegir qué puntos resolver según su experiencia y tiempo disponible. La completitud del entregable lo acercará al nivel del perfil que buscamos: senior, medium o junior.
- Se valorará un entregable bien documentado.
- Los entregables de los puntos #1 y #2 deben ser publicados en su GitHub personal y compartido el enlace del repo. Publique el código fuente de la implementación con todos sus artefactos, incluida la documentación, en su repositorio personal de Github. El repo debe quedar desplegado de forma pública, NO privada. Evite clonar repos/trabajos de otras personas, valoramos el esfuerzo por realizar un trabajo de su propia autoría.

1. Ingeniería de datos + Cloud.

El desafío de implementación consta de diferentes secciones que se centran en habilidades para la extracción, la transformación, el almacenamiento, el despliegue y el consumo de datos.

Siéntase libre de registrar los errores que cometa en el proceso de implementación; entendemos y aceptamos la incertidumbre y vemos oportunidades cuando son comprendidos y superados.

Tenga en cuenta:

- Es un plus si incorpora diagramas de arquitectura dentro de la documentación.
- Si no sabe cómo resolver alguno de los desafíos planteados, puede continuar con el siguiente.
- El lenguaje de programación a usar debe ser Python. Siéntase libre de incorporar las librerías y frameworks que considere a bien.
- Considere el uso de la capa gratuita de servicios alguna nube pública (Azure, Databricks, GCP, AWS, etc). La predilección en la escogencia de servicios de la nube pública de Azure o Databricks Community Edition, será considerado un plus.
- La documentación de la solución será un aspecto que consideraremos. Puede realizar un “readme” markdown en GitHub, publicar un PDF, desarrollar un notebook, o cualquier otra alternativa que considere a bien.
- Publique el código fuente de la implementación con todos sus artefactos, incluida la documentación, en su repositorio personal de Github. El repo debe quedar desplegado de forma pública, NO privada.
- Intente aplicar siempre las mejores prácticas y desarrollar una solución escalable.
- Piense en la deuda técnica y hazla explícita en la documentación.

Desafíos:

- ***DESAFIO #1:*** Construya un script que genere de forma automática los datos de: departamentos, puestos de trabajo, y empleados. Considere el siguiente ejemplo (no es la única solución):

```
import pandas as pd
import numpy as np
from datetime import datetime

# Course Categories Data
course_categories_data = {
   "category_id": [1, 2, 3, 4, 5],
   "category_name": ["Technology", "Business", "Art", "Science", "Health"]
}

# Course Levels Data
course_levels_data = {
   "level_id": [101, 102, 103],
   "level_name": ["Beginner", "Intermediate", "Advanced"]
}

# Courses Data
np.random.seed(0)
courses_data = {
   "course_id": np.arange(1, 101),
   "course_name": [f"Course{i}" for i in range(1, 101)],
   "category_id": np.random.choice([1, 2, 3, 4, 5], 100),
   "level_id": np.random.choice([101, 102, 103], 100),
   "start_date": [datetime(2021, np.random.randint(1, 13), np.random.randint(1, 29)) for _ in range(100)]
}

# Convert to DataFrame
course_categories_df = pd.DataFrame(course_categories_data)
course_levels_df = pd.DataFrame(course_levels_data)
courses_df = pd.DataFrame(courses_data)
```

- ***DESAFIO #2:*** Guarde los datos simulados en archivos con formato CSV/Parquet. Explique el porqué de la escogencia del formato. No descarte usar la capa gratuita de algún servicio de almacenamiento tipo cloud, será considerado un plus.

- ***DESAFIO #3:*** Implemente un proceso batch para migrar los datos a una base de datos SQL/NoSQL, o si lo desea, a un Datawarehouse o bucket analítico de un Datalake. No descarte usar la capa gratuita de algún servicio de almacenamiento tipo cloud, será considerado un plus.

- ***DESAFIO #4:*** Dependiendo si escoge una base de datos SQL/NoSQL, un Datawarehouse, o un Datalake, entonces desarrolle una view/query/report a partir del modelo de datos.

- ***DESAFIO #5:*** Desarrolle una API REST para consultar la view/query/report. Para el desarrollo de la API considere algún framework de Python, C#/.Net.

- ***DESAFIO #6:*** Mejore la implementación de la API realizando un despliegue que use contenedores (valide las distintas opciones que le brinda su nube). Considere una prueba de consumo a la API implementando o activando algún front de acceso para ejecutar la invocación a la view/query/report.

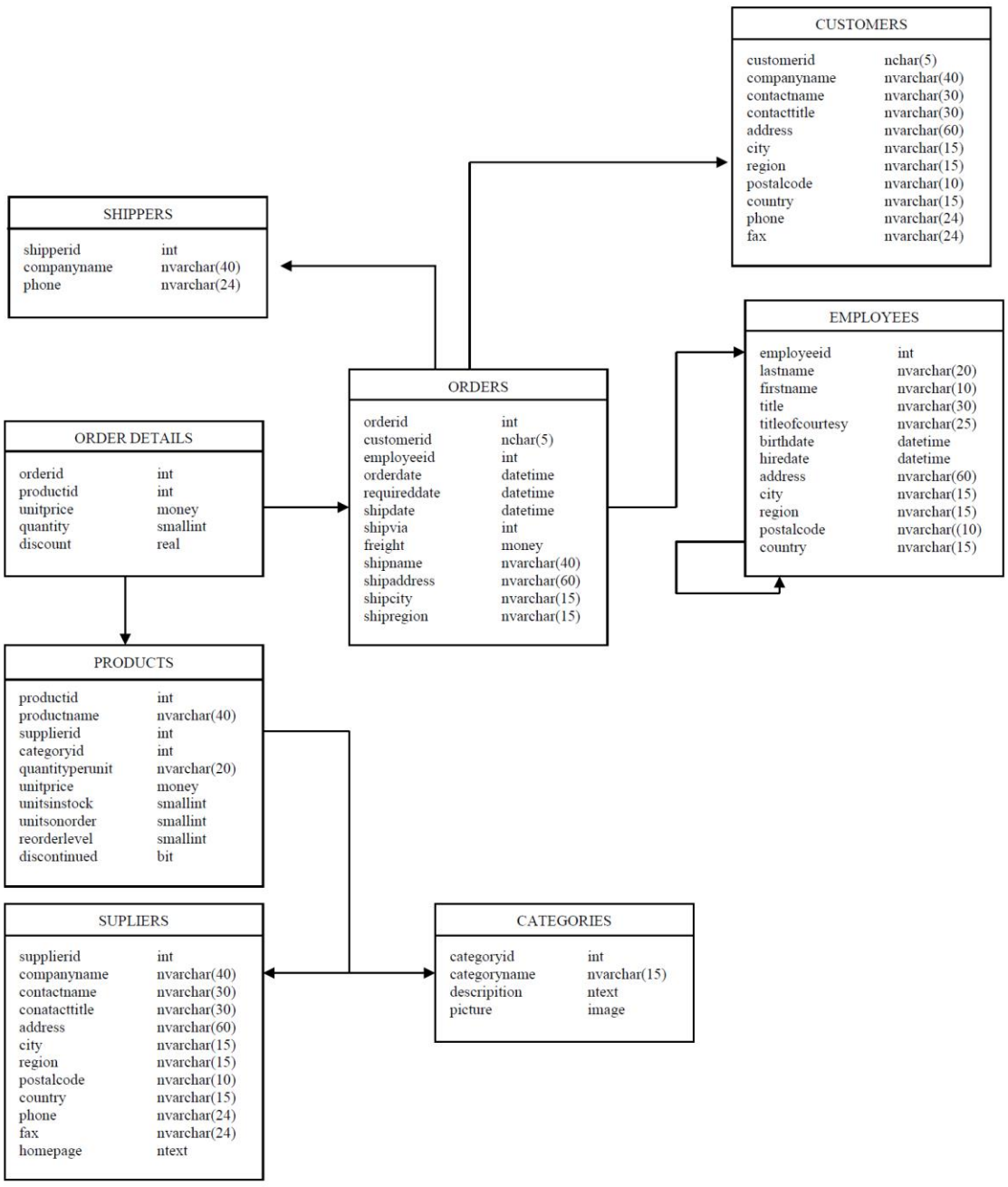
PRUEBA TÉCNICA INGENIERIA DE DATOS

2. Modelamiento de datos para BI

Para el desarrollo de este punto, considerar la siguiente fuente tipo ODATA y su respectivo diagrama de datos.

Fuente de datos: https://services.odata.org/v4/Northwind/Northwind.svc

Diagrama de datos:



Desafíos:

- ***DESAFIO #1:*** Usando la herramienta Power BI Desktop y el conector OData deberá extraer los datos y crear un modelo dimensional que contemple las buenas prácticas. El modelo deberá asegurar un adecuado performance al ser consultado y reducir la redundancia en los datos. Registre en un PDF/Word,
- ***DESAFIO #2:*** Diseñe un pequeño dashboard interactivo que visualice tres indicadores que considere los más relevantes para explicar los datos. Será un plus considerar el uso de filtros, inteligencia de tiempo, medidas DAX y una adecuada estética de colores y disposición de los elementos. Incluya una page oculta con las notas técnicas que expliquen, brevemente, porque el modelo diseñado está optimizado para un correcto rendimiento y para reducir al máximo la redundancia de los datos.