

01 - Prueba técnica Python Data Analyst

Descripción La prueba tiene como objetivo desarrollar una analítica sobre un dataset volcado en una base de datos relacional e implementar un pipeline de ETL que realice cierto procesamiento y agregación. En la base de datos te encontrarás dos tablas, **orders**, con información a nivel de transacción, y **merchants**, con información a nivel de comercio.

Los datos de acceso a la base de datos son los siguientes (puedes explorar este mismo repositorio en otra herramienta como DBeaver)

Facilitamos acceso de solo lectura a la base de datos:

- Usuario: postgres
- Contraseña: password
- DB name: dbneoland
- URI completa: postgresql://postgres:password@data-challenge.co4whz3w2rtn.us-east-1.rds.amazonaws.com:5432/dbneoland

La prueba debe desarrollarse con Python 3. Puedes utilizar todas las librerías que te resulte necesario, por ejemplo pandas, plotly/bokeh o SQLAlchemy.

Antes de comenzar con la prueba, lee el enunciado al completo.

Tarea 1: Análisis del dataset

Apoyándote en uno o varios Jupyter Notebooks, realiza un análisis sobre los datos de la tabla orders contemplando los siguientes puntos:

1. Análisis sencillo del dataset proporcionado variable a variable, obteniendo métricas resumen o representaciones gráficas de cada una de ellas. ¿En base a este análisis, crees que alguna variable debería ser descartada?
 2. Realiza un scatter plot en el que se compare el volumen financiado en un crédito con el número de cuotas elegido (number_installments). ¿Qué se observa?
 3. ¿Cómo ha evolucionado el número de créditos concedidos a lo largo del año 2018? ¿Y el volumen prestado?
 4. Trata de obtener algún insight interesante del dataset y realiza una explicación detallada, apoyándote de las métricas y las representaciones gráficas que sean necesarias.
-

Tarea 2: Pipeline de ETL

Desarrollando paquetes y/o módulos de python y apoyándote de funciones y clases de python según mejor consideres, implementa un ETL siguiendo esta lógica:

1. **Extracción:** recupera los datos necesarios de la base de datos.
2. **Transformación:** realiza los procesamientos necesarios para obtener las tablas que se describen en la descripción.
3. **Carga:** como no se busca a escribir lo resultados en la DB, genera ficheros con la información que se escribiría en las tablas de la DB.

Debes generar tres ficheros con extensión .xlsx en forma de dataframes de pandas.

Los dos primeros deben contener dos columnas: un *conteo del número de créditos concedidos* y la *suma del principal financiado*. Una de ellas debe contener información a nivel de merchant cuyo índice sea el id del merchant (*merchant_uuid*). La otra debe construirse a nivel de sector y su índice debe ser el código del sector (*industry_code*).

El tercer fichero debe contener la distribución mensual del principal financiado por sector. Es decir, el índice del dataframe serán los meses de producción de crédito y las columnas los códigos de sector (*industry_code*). Te proponemos que te apoyes en un ORM como **SQLAlchemy**, pero si lo prefieres puedes escribir *vanilla/plain SQL*.

Tarea bonus

Solamente si vas bien de tiempo, te proponemos elaborar un conjunto de tests con el objetivo de validar el correcto funcionamiento de la ETL.

No se trata de una cuestión de cantidad de tests, si no de que comprueben los aspectos más fundamentales o críticos. Ten en cuenta la restricción temporal de la prueba para así priorizar los tests más relevantes

Te proponemos en utilizar **pytest** para implementación.

Objetivos

- Obtener insights que aporten valor
 - Transmitir información de forma clara y rigurosa
 - Diferenciar la información relevante del ruido
 - Demostrar claridad y eficiencia en el código y en las queries
 - Demostrar un buen manejo de la base de datos relacional con Python
 - Pasar un buen rato y aprender ;)
-

Entrega

Para la entrega de la **tarea 1** descárgate los siguientes ficheros:

1. El notebook de la tarea 1 en .ipynb
2. Este mismo notebook en .html

Para la entrega de la **tarea 2**, crea un repositorio de código donde incluyas todo el código desarrollado. Incluye en el repositorio un fichero README muy sintetizado que las instrucciones que consideres necesarias. Si has realizado la tarea bonus, indícanos cómo ejecutar la batería de tests. No olvides realizar el pull-requests de Github, para su comprobación. Adjunta los notebooks y el html (en archivos por separado o en un fichero zip) y la url del repositorio de Github (privado).