

1) Procesamiento de archivos HTML en Python

Se le ha encomendado diseñar uno o varios scripts en Python (según lo considere) que permitan:

- Recibir bien sea un listado de archivos HTML a procesar o un listado de directorios en los cuales se encuentran archivos HTML para procesar (incluyendo subdirectorios).
- Recorrer el listado completo de archivos HTML y determinar para cada archivo cuáles son las imágenes que tiene asociadas (puede asumir que todas se encuentran con el tag ``) y convertirlas a base64 (<https://en.wikipedia.org/wiki/Base64>).
- Reemplazar las imágenes originales del HTML por las codificadas en base64, sin sustituir el archivo original, es decir, creando uno nuevo.
- Debe generar un objeto que contenga la lista de imágenes procesadas de forma exitosa y las que fallaron:

```
{
    success: {},
    fail: {}
}
```

2) Preferencias de consumo

En el archivo de Excel `bd.xlsx` encontrará una base de datos que contiene información transaccional de clientes y sus categorías de consumo, cada hoja corresponde a una tabla.

A partir de dicha base y utilizando lenguaje SQL se pretende crear un conjunto de datos que contenga:

- Datos de identificación del cliente.
- Categoría de consumo preferida dentro de cada cadena (tipo o clase de almacén): Aquella en la que acumuló más cantidad de transacciones.
- Fecha de la última transacción en cada categoría preferida.

Importante:

- Las consultas deben ser flexibles en el sentido de que permitan generar uno o más niveles de preferencia, por ejemplo:
 - Las `n` primeras categorías preferidas dentro de cada cadena.
 - La segunda categoría preferida dentro de cada cadena.
- La evaluación de la preferencia se debe realizar dentro de una ventana temporal que se filtre dentro de las consultas de transacciones (puede usarse o no toda la base de datos, dependiendo del filtro y de los datos mismos).

3) Rachas

En el archivo de Excel `rachas.xlsx` se encuentra la información de saldos de clientes por corte de mes. Se le ha solicitado cargar los datos a una base de datos y generar una consulta que cumpla con los siguientes

criterios:

- Clasificar la información por niveles de deuda, así (ver hoja `historia`):
 - N0: Saldo ≥ 0 y $< 300,000$.
 - N1: Saldo $\geq 300,000$ y $< 1,000,000$.
 - N2: Saldo $\geq 1,000,000$ y $< 3,000,000$.
 - N3: Saldo $\geq 3,000,000$ y $< 5,000,000$.
 - N4: Saldo $\geq 5,000,000$.
- Si un cliente no aparece en un mes específico (después de su primera aparición), se considera que su saldo es N0, excepto si el corte de mes es superior a su fecha de retiro (ver hoja `retiros`).

Generar una consulta (o conjunto de consultas) que:

- Permita realizar todo el ejercicio con base en una fecha específica (`fecha_base`), es decir, como si estuviera "parado" en una fecha que no necesariamente es el día de hoy.
- Identificar los meses consecutivos (racha) de un cliente *dentro de un nivel de saldo*.
- Seleccionar aquellas rachas que sean mayores o iguales a un número específico `n`.
- Si un cliente tiene más de una racha que cumpla con lo anterior, se debe seleccionar la racha más larga.
- Si aún así se sigue presentando más de una racha, se selecciona aquella cuyo término sea más reciente (fecha más próxima y menor o igual a la `fecha_base`).
- El resultado debe presentar para cada cliente:
 - `identificacion`.
 - `racha`: Número de meses consecutivos seleccionados que cumplen con los criterios.
 - `fecha_fin`: Corte de mes en el cual se presenta el fin de la racha identificada.
 - `nivel`: Nivel de saldo asociado a la racha.

Aspectos a tener en cuenta

La solución debe crearse en un repositorio de GitHub (bien sea público o privado al cual nos pueda dar acceso), el cual debe tener un documento README en el cual explique el desarrollo que dio a cada ejercicio.

Para el ejercicio 1

- Debe utilizar únicamente librerías built-in (standard library).
- Se recomienda utilizar prácticas de código limpio / principios SOLID / PEP, cuando considere que aplica.
- Es deseable que utilice programación orientada a objetos.

Para los ejercicios 2 y 3

- Puede utilizar SQLite / MySQL / PostgreSQL / Impala.
- Debe proporcionar todos los scripts necesarios para replicar la construcción de la base de datos y la solución de los numerales solicitados.
- Se recomienda utilizar buenas prácticas de escritura de SQL, cuando considere que aplica.
- Es su responsabilidad controlar la calidad de los datos, dentro de lo posible con la información proporcionada.