**Documentación del Código: Sistema de Cálculo de Comisiones y Envío de Resultados**

**Objetivo**

El propósito de este código es automatizar el cálculo de comisiones de una API para diferentes empresas clientes de BATSEJ OPEN FINANCE S.A. Este sistema tiene en cuenta las condiciones de los contratos, el cálculo de IVA, posibles descuentos y el envío de los resultados a través de correo electrónico.

**Estructura General del Proyecto**

El proyecto está dividido en varias clases, cada una encargada de una responsabilidad específica. Estas clases incluyen:

1. **Comisiones**: Calcula las comisiones de las empresas, aplica descuentos si es necesario, guarda los resultados en un archivo Excel y envía los resultados por correo electrónico.
2. **DataLoader**: Carga los datos necesarios desde la base de datos SQLite y maneja la conexión con la base de datos.
3. **EmailSendOutlook**: Se encarga de autenticar y enviar correos electrónicos a través de Outlook utilizando la autenticación OAuth2 se realizó de esta forma ya que Microsoft tomo la decisión de desactivar la forma Básica que consistía en autenticarte solo con usuario y contraseña y hay librería que ni siquiera requerían autenticación para enviar un correo.
4. **Archivos de SQL**: Consultas SQL para interactuar con la base de datos y obtener la información, se trató de manejar casi todo desde la base de datos para evitar en caso de cambio tocar el código.

**Clase Comisiones**

La clase Comisiones es la clase principal encargada de coordinar todo el flujo de trabajo: carga de datos, cálculo de comisiones, aplicación de descuentos y envío de resultados.

**Atributos**

* **directorio**: Ruta del directorio donde se encuentran los archivos necesarios para el procesamiento (base de datos, archivos SQL, etc.).
* **data\_loader**: Instancia de la clase DataLoader, encargada de cargar los datos de la base de datos.
* **parametros**: Diccionario que almacena parámetros importantes como fechas, direcciones de correo, credenciales, etc.
* **email\_sender**: Instancia de la clase EmailSendOutlook, encargada de enviar los correos electrónicos con los resultados.

**Métodos**

* **\_\_init\_\_(self, directorio)**: Constructor que inicializa las instancias de DataLoader y EmailSendOutlook, y carga los parámetros desde la base de datos.
* **load\_data(self)**: Carga los datos necesarios desde la base de datos. Se cargan las tablas commissions y discounts, así como los parámetros de configuración. También convierte las fechas de inicio y fin en un formato más legible.
* **calculate\_commissions(self)**: Calcula las comisiones de las empresas según los contratos. Utiliza consultas SQL para obtener las estadísticas de las peticiones y luego aplica las reglas de los contratos (comisiones fijas o variables). Asegura que los valores nulos se llenen con los valores máximos de las peticiones exitosas y no exitosas.
* **save\_results(self, df\_calculo\_comisiones)**: Guarda los resultados del cálculo de comisiones en un archivo Excel. El archivo se guarda en una carpeta **result** dentro del directorio especificado. Si la carpeta no existe, se crea automáticamente.
* **send\_results(self, file\_path, df\_calculo\_comisiones)**: Prepara y envía un correo electrónico con los resultados de las comisiones. El correo incluye un resumen de las comisiones calculadas en formato HTML y adjunta el archivo Excel con los resultados.
* **run(self)**: Método principal que coordina el flujo completo: carga de datos, cálculo de comisiones, guardado de resultados y envío del correo electrónico.

**Clase DataLoader**

La clase DataLoader es responsable de cargar los datos desde la base de datos SQLite, así como de manejar la conexión con la misma.

**Atributos**

* **directorio**: Ruta del directorio donde se encuentra la base de datos y otros archivos relacionados.
* **db\_path**: Ruta completa a la base de datos database.sqlite.
* **connection**: Conexión a la base de datos SQLite.

**Métodos**

* **\_\_init\_\_(self, directorio)**: Constructor que inicializa el directorio y la conexión a la base de datos.
* **get\_db\_connection(self)**: Establece la conexión a la base de datos SQLite. Si la base de datos no existe, lanza una excepción.
* **load\_table\_as\_dataframe(self, table\_name)**: Carga una tabla completa desde la base de datos como un DataFrame de Pandas.
* **load\_parameters(self)**: Carga los parámetros desde la tabla paremeter de la base de datos, que contiene configuraciones como las fechas de inicio y fin.
* **load\_sql(self, sql\_file, params)**: Carga y ejecuta una consulta SQL desde un archivo .sql y devuelve el resultado como un DataFrame. Los parámetros para la consulta son pasados dinámicamente.
* **close\_connection(self)**: Cierra la conexión a la base de datos.

**Clase EmailSendOutlook**

La clase EmailSendOutlook se encarga de la autenticación y el envío de correos electrónicos mediante la API de Outlook (Microsoft Graph API) utilizando el flujo de autenticación OAuth2.

**Atributos**

* **client\_id**: ID de cliente de la aplicación registrada en Azure Active Directory.
* **tenant\_id**: ID de inquilino en Azure Active Directory.
* **username**: Nombre de usuario de Outlook (correo electrónico).
* **password**: Contraseña de Outlook.
* **authority**: URL de autorización para obtener el token de acceso.
* **scopes**: Alcances requeridos para la API de Outlook.
* **access\_token**: Token de acceso obtenido tras la autenticación.

**Métodos**

* **\_\_init\_\_(self, client\_id, tenant\_id, username, password)**: Constructor que inicializa los valores necesarios para la autenticación de Outlook.
* **authenticate(self)**: Realiza la autenticación mediante OAuth2 y obtiene un token de acceso. Si la autenticación falla, lanza una excepción.
* **send\_email(self, recipient, subject, body)**: Envía un correo electrónico con los resultados de las comisiones. El cuerpo del correo incluye un resumen de las comisiones y un archivo adjunto con los resultados en formato Excel.

**Archivo SQL y Consultas**

El código hace uso de consultas SQL almacenadas en archivos .sql para interactuar con la base de datos:

1. **count\_call\_api.sql**: Consulta que obtiene el número de peticiones exitosas y no exitosas para las empresas durante el mes de julio y agosto de 2024.
2. **cc.sql**: Consulta que calcula las comisiones según los contratos y las condiciones especificadas en el planteamiento del problema.

**Flujo Completo de Ejecución**

1. **Carga de datos**: Se cargan los datos desde la base de datos SQLite utilizando la clase DataLoader. Esto incluye las tablas de comisiones y descuentos, así como los parámetros de configuración (fechas, correo, credenciales).
2. **Cálculo de comisiones**: Se calcula el total a cobrar según las peticiones exitosas y no exitosas. Las comisiones son determinadas por las condiciones de cada contrato (fijo o variable). También se aplica el IVA (19%) y los descuentos, si es necesario.
3. **Guardado de resultados**: Los resultados son guardados en un archivo Excel, que se almacena en una carpeta de resultados dentro del directorio especificado.
4. **Envío de resultados**: Un correo electrónico es enviado a los destinatarios con el resumen de las comisiones calculadas y un archivo adjunto con los resultados en formato Excel.

**Contextualización del proceso**

1. **Autenticación con OAuth2 en lugar de autenticación básica**: Microsoft desactivó el método de autenticación básica, que anteriormente permitía que aplicaciones de terceros autenticaran a los usuarios simplemente utilizando su nombre de usuario y contraseña. Este cambio se realizó debido a que la autenticación básica es menos segura que el flujo de OAuth2, que es más robusto y proporciona una mayor seguridad al manejar el acceso a los servicios de Microsoft 365.
2. **Obtención de credenciales**: Para autenticarte y obtener las credenciales necesarias para enviar correos electrónicos a través de Outlook con OAuth2, debes seguir estos pasos:

**Pasos para obtener las credenciales necesarias:**

**1. Instalación de los módulos necesarios en PowerShell si asi lo quieres.**

Desde PowerShell, puedes instalar el módulo de Microsoft Graph, que es la interfaz para interactuar con los servicios de Microsoft 365, como Outlook. La documentación oficial de Microsoft explica cómo realizar la instalación.

**Enlace de la documentación:**

* [Instalación del módulo de Microsoft Graph en PowerShell](https://learn.microsoft.com/en-us/powershell/microsoftgraph/installation?view=graph-powershell-1.0)

Aquí se especifican los pasos para instalar el módulo y configurar el entorno para interactuar con Microsoft Graph utilizando PowerShell.

**¡SI no!:**

**2. Crear una aplicación en Azure Active Directory (Azure AD)**

Para utilizar OAuth2, es necesario crear una aplicación en Azure AD que pueda solicitar permisos para enviar correos en nombre del usuario. Esta aplicación se puede crear desde la consola gráfica de Azure, o alternativamente, puedes usar PowerShell para gestionarla.

**Enlace de la documentación:**

* [Tutorial para aplicaciones de solo autenticación con Python y PowerShell](https://learn.microsoft.com/en-us/graph/tutorials/Python-app-only?tabs=aad&tutorial-step=1)

Este tutorial te guía paso a paso para registrar una aplicación en Azure AD, establecer los permisos necesarios y obtener las credenciales pero desde la parte grafica.

**3. Obtener el token de acceso**

Una vez que hayas registrado la aplicación en Azure AD, podrás obtener un token de acceso mediante el flujo de autenticación OAuth2. Este token es el que usarás para autenticar las solicitudes que la aplicación haga a la API de Microsoft Graph (como el envío de correos).

El token de acceso se obtiene al interactuar con el servicio de autenticación de Microsoft y se debe incluir en las solicitudes a Microsoft Graph para enviar correos electrónicos.

**4. Uso de la API de Microsoft Graph para enviar correos**

Con el token de acceso en mano, puedes usar la API de Microsoft Graph para enviar correos electrónicos a través de Outlook. El proceso de envío de correos es manejado mediante solicitudes HTTP que incluyen el token de acceso en los encabezados de la petición.

**Resumen del flujo**

1. **Instalar el módulo de Microsoft Graph**: Usar PowerShell o la interfaz gráfica de Azure.
2. **O**
3. **Registrar la aplicación en Azure AD**: Establecer los permisos de acceso y configurar la aplicación para usar OAuth2.
4. **Obtener el token de acceso**: Autenticarse mediante OAuth2 y obtener el token necesario para realizar acciones en nombre del usuario.
5. **Enviar correos electrónicos**: Usar la API de Microsoft Graph con el token de acceso para enviar correos a través de Outlook.

Este proceso reemplaza la antigua autenticación básica, mejorando la seguridad y la integridad de las aplicaciones que interactúan con los servicios de Microsoft.  
  
  
**Documentación de la Base de Datos**

**1. Tabla: apicall**

La tabla apicall registra cada llamada realizada a la API, incluyendo detalles como la fecha, el comercio asociado y el estado de la solicitud. Además, contiene un campo que indica si la llamada está relacionada con otros datos.

* **date\_api\_call**: Fecha de la llamada a la API en formato texto.
* **commerce\_id**: Identificador único del comercio asociado a la llamada.
* **ask\_status**: Estado de la solicitud (por ejemplo, "exitosa" o "fallida").
* **is\_related**: Indicador de relación (1 si está relacionada, 0 si no lo está).

**2. Tabla: commerce**

La tabla commerce almacena información detallada sobre los comercios registrados en el sistema.

* **commerce\_id**: Identificador único del comercio.
* **commerce\_nit**: Número de Identificación Tributaria (NIT) del comercio.
* **commerce\_name**: Nombre del comercio.
* **commerce\_status**: Estado del comercio, como "Activo" o "Inactivo".
* **commerce\_email**: Correo electrónico asociado al comercio.

Esta tabla se relaciona con otras tablas a través del campo commerce\_id.

**3. Tabla: commissions**

La tabla commissions define las reglas para el cálculo de comisiones según el número de solicitudes exitosas procesadas.

* **trade\_name**: Nombre comercial asociado a las comisiones.
* **min\_successful\_requests**: Mínimo de solicitudes exitosas requeridas para aplicar la comisión.
* **max\_successful\_requests**: Máximo de solicitudes exitosas permitidas para aplicar la comisión.
* **commission\_cost**: Costo fijo de la comisión por transacción.
* **percentage\_vat**: Porcentaje de IVA aplicable a la comisión.

Esta tabla permite determinar los costos y condiciones de las comisiones en función de las transacciones.

**4. Tabla: csc**

La tabla csc consolida información sobre el rendimiento de los comercios en cuanto a solicitudes exitosas y fallidas, además de proporcionar datos básicos del comercio.

* **commerce\_id**: Identificador único del comercio.
* **commerce\_nit**: Número de Identificación Tributaria del comercio.
* **commerce\_name**: Nombre del comercio.
* **commerce\_email**: Correo electrónico asociado al comercio.
* **formatted\_date**: Fecha en un formato estandarizado para reportes.
* **successful\_count**: Número de solicitudes exitosas realizadas.
* **unsuccessful\_count**: Número de solicitudes fallidas realizadas.

Esta tabla se relaciona con commerce a través del campo commerce\_id.

**5. Tabla: discounts**

La tabla discounts establece las reglas para otorgar descuentos basados en el número de solicitudes fallidas.

* **trade\_name**: Nombre comercial asociado al descuento.
* **min\_unsuccessful\_requests**: Mínimo de solicitudes fallidas requeridas para aplicar el descuento.
* **max\_unsuccessful\_requests**: Máximo de solicitudes fallidas permitidas para aplicar el descuento.
* **discount\_percentage**: Porcentaje de descuento aplicable.

Los descuentos están diseñados para incentivar una mejor gestión de las solicitudes fallidas.

**6. Tabla: parameters**

La tabla parameters contiene configuraciones clave del sistema que son utilizadas para personalizar y automatizar ciertos procesos.

* **parameter\_name**: Nombre del parámetro (por ejemplo, "mail\_to" o "password").
* **value**: Valor asociado al parámetro.

**Ejemplo de parámetros:**

* mail\_to: Dirección de correo electrónico para notificaciones (ejemplo: example@gmail.com).
* subject: Asunto de las notificaciones por correo (ejemplo: Cuenta de cobro comisiones).
* date\_init: Fecha de inicio del período (ejemplo: 20240701000000).
* date\_end: Fecha de finalización del período (ejemplo: 20240831235959).
* password: Contraseña utilizada para acceso seguro.

**7. Tabla: sqlite\_master**

Esta tabla es gestionada automáticamente por SQLite y no debe ser modificada manualmente. Contiene información sobre la estructura de las tablas, vistas e índices.

**Relación entre Tablas**

* **commerce\_id**: Es la clave principal para relacionar las tablas apicall, commerce y csc.
* **trade\_name**: Relaciona las tablas commissions y discounts, conectando las reglas de comisiones y descuentos con un nombre comercial específico.