Documentação Técnica
Automação de Geração de Briefings via Integração com Google Workspace

---

Acessos:

Google Drive (link compartilhavel)

Repositório GitHub

1. Introdução

Este projeto tem como objetivo automatizar a criação de documentos de briefing com base em dados estruturados registrados em uma planilha do Google Sheets, utilizando um template previamente definido no Google Docs. A automação visa eliminar o processo manual de copiar, formatar e preencher documentos, reduzindo significativamente o tempo de execução e a incidência de erros humanos, além de padronizar os arquivos gerados.

Como ponto de partida, foi criado um arquivo de template no Google Docs contendo tags identificadoras (placeholders), como {{NOME_DO_PROJETO}}, {{RESPONSAVEL}}, entre outras. Essas tags funcionam como marcadores que serão substituídos dinamicamente pelos dados correspondentes obtidos da planilha. O código realiza a leitura dessas informações, verifica se o documento já foi gerado previamente e, caso contrário, cria uma cópia do template, faz as substituições necessárias e armazena o novo documento em uma pasta específica no Google Drive.

2. Objetivo

Automatizar, de forma segura e escalável, a geração de documentos no Google Docs a partir de dados armazenados em uma planilha do Google Sheets, garantindo:

- Redução de esforço manual
- Padronização do layout dos documentos de briefing
- Verificação de duplicidade de documentos antes da geração
- Integração segura com a conta Google do usuário (OAuth 2.0)
- Armazenamento centralizado em uma pasta do Google Drive

Caroline S. Amarante
2025

3. Tecnologias Utilizadas

A escolha da linguagem Python, em conjunto com as bibliotecas oficiais de integração com as APIs do Google Workspace, se deu pela flexibilidade, suporte robusto, capacidade de personalização da lógica de negócios e facilidade de autenticação via OAuth para uso com contas pessoais ou corporativas. A abordagem baseada em template com placeholders oferece desacoplamento entre lógica e layout do documento.

Tecnologias: Python 3.12, gspread, google-api-python-client, google-auth, google-auth-oauthlib, tkinter, tqdm.

4. Estrutura do Projeto

main.py	Script principal da automação
client_secret.json	Credenciais OAuth do projeto Google
config.json	IDs de recursos utilizados
token.pickle	Token gerado após autenticação OAuth
requirements.txt	Lista de dependências do projeto
run_automation.bat	Script para execução automatizada no Windows

5. Configuração

5.1. Google Cloud Console

1. Crie um projeto em Google Cloud Console.
2. Ative as APIs:
    - Google Drive API
    - Google Docs API
    - Google Sheets API
3. Configure a tela de consentimento OAuth:
    1. Tipo: Externa
    2. Escopos necessários:
        - https://www.googleapis.com/auth/drive
        - https://www.googleapis.com/auth/documents
        - https://www.googleapis.com/auth/spreadsheets.readonly
    3. Crie um **OAuth Client ID** (tipo Desktop App) e baixe o arquivo como client_secret.json.

5.2. Arquivo de Configuração: config.json

```
{
"sheetsID": "ID_DA_PLANILHA",
"templateID": "ID_DO_TEMPLATE_GDOCS",
"outputFolderID": "ID_DA_PASTA_NO_DRIVE"
}
```

6. Execução Local (Windows)

6.1. Ambiente Virtual:

- python -m venv venv
- call venv\Scripts\activate
- pip install -r requirements.txt
- run_automation.bat

7. Lógica da Automação

1. Leitura da planilha com gspread
2. Verificação se o documento já existe no Drive, se sim, o mesmo é atualizado
3. Cópia do template e substituição de campos no Google Docs
4. Abertura automática da pasta no navegador
5. Exibição de resumo via pop-up (tkinter)

8. Tags Substituídas no Template

Placeholder Template	Coluna Excel
{{NOME_DO_PROJETO}}	Nome do Projeto
{{RESPONSAVEL}}	Responsável
{{PRAZO_FINAL}}	Prazo Final
{{DESCRICAO_DO_PROJETO}}	Descrição do Projeto
{{OBSERVACOES_ADICIONAIS}}	Observações Adicionais
{{STATUS_DO_PROJETO}}	Status do Projeto
{{DIAS_ATE_FINALIZACAO}}	Dias até Finalização

9. Melhorias

- Exportação automática em PDF.
- Geração de logs detalhados (data, ID, status).
- Execução programada (via cron ou agendador de tarefas).
- Integração com Google Forms para preencher os dados de forma amigável.
- Interface gráfica completa para seleção de template/pasta.

10. Limitações

- A automação depende de conectividade com a internet e acesso autorizado às APIs do Google.
- A conta autenticada precisa ter permissão explícita nos arquivos e pastas usados.
- A estrutura da planilha deve estar padronizada, com nomes de colunas exatos.
- uso de OAuth com service accounts pode exigir quota de armazenamento específica.
- script está orientado a execução local e não possui controle de erros para exceções complexas (ex: falha parcial na substituição).

11. Links de Apoio

Como criar um proeto – Google Cloud Console

https://developers.google.com/workspace/guides/create-project?hl=fr

Como obter API Key – Google Cloud Console

https://www.youtube.com/watch?v=brCkpzAD0gc&ab_channel=JieJenn

Como utilizar OAuth para API's – Google Cloud Console

https://developers.google.com/identity/protocols/oauth2?hl=fr

Caroline S. Amarante
2025