

TikTok Ads API → BigQuery

API para extração automatizada de dados de campanhas publicitárias do TikTok Ads e carga no Google BigQuery.

📋 Sumário

- [Visão Geral](#)
- [Arquitetura](#)
- [Dados Extraídos](#)
- [Configuração](#)
- [Deploy](#)
- [Uso](#)
- [Airflow DAG](#)
- [Desenvolvimento Local](#)
- [Troubleshooting](#)

Visão Geral

Objetivo

Automatizar a extração diária de métricas de campanhas publicitárias do TikTok Ads para centralização e análise no BigQuery.

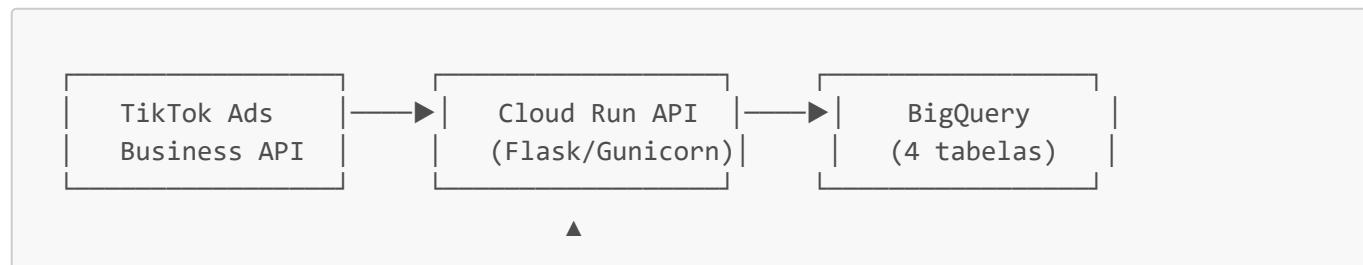
Por que esta API foi desenvolvida?

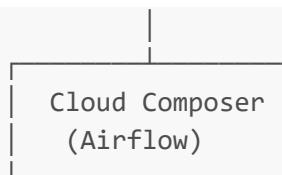
- **Centralização de dados:** Consolida dados de múltiplas contas TikTok Ads em um único data warehouse
- **Automação:** Elimina processos manuais de extração e carga
- **Análise avançada:** Permite análises cross-platform com outras fontes de mídia
- **Histórico:** Mantém histórico completo para análises de tendências

Autor

- **Equipe:** Data Engineering Team
- **Data de criação:** Janeiro 2025
- **Versão:** 1.0.0

Arquitetura





Componentes

Componente	Tecnologia	Descrição
API	Flask + Gunicorn	Serviço HTTP containerizado
Runtime	Cloud Run	Execução serverless
Orquestração	Cloud Composer (Airflow)	Agendamento diário
Data Warehouse	BigQuery	Armazenamento e análise
Container Registry	Artifact Registry	Imagens Docker

Dados Extraídos

Tabelas no BigQuery

A API gera **4 tabelas** no dataset **RAW**:

Tabela	Dimensões	Descrição
TKT001_TIKTOK ADS ADVERTISER	advertiser_id, stat_time_day	Métricas agregadas por conta
TKT002_TIKTOK ADS CAMPAIGN	campaign_id, stat_time_day	Métricas por campanha
TKT003_TIKTOK ADS ADGROUP	adgroup_id, stat_time_day	Métricas por grupo de anúncios
TKT004_TIKTOK ADS AD	ad_id, stat_time_day	Métricas por anúncio individual

Métricas Disponíveis

Core Metrics

- spend, impressions, clicks, ctr, cpc, cpm, reach, frequency

Engagement

- profile_visits, likes, comments, shares, follows, engagements

Video Metrics

- video_play_actions, video_watched_2s, video_watched_6s
- video_views_p25, video_views_p50, video_views_p75, video_views_p100

Conversion

- `conversions, conversion_rate, cost_per_conversion`
- `results, result_rate, cost_per_result`

In-App Events

- `app_install, registration, purchase, checkout, view_content`
- E muitas outras métricas de eventos in-app

Campos de Metadados

Cada registro inclui:

- `_advertiser_id`: ID do anunciante
- `_extracted_at`: Timestamp da extração (UTC)
- `_report_type`: Tipo de relatório
- `date`: Data do dado (derivada de `stat_time_day`)

Configuração

Variáveis de Ambiente

Variável	Obrigatória	Default	Descrição
<code>PORT</code>	Não	8080	Porta do servidor
<code>DAYS_REPROCESS</code>	Não	3	Dias de reprocessamento

Variáveis Airflow

Configure no Cloud Composer:

```
# Variables
TIKTOK_ADS_CLOUD_RUN_URL = "https://tiktok-ads-api-xxxxx.run.app"
TIKTOK_ACCESS_TOKEN = "seu_access_token"
TIKTOK_ADVERTISER_IDS = ["1234567890123456789"]
GCP_PROJECT = "cadastra-yduds-prod"
GCP_REGION = "southamerica-east1"
```

Obtenção do Access Token TikTok

1. Acesse [TikTok for Business](#)
2. Vá em **Assets** → **Events** → **App Settings**
3. Em **Marketing API**, gere um Access Token
4. O token não expira, mas pode ser invalidado se o anunciante revogar

Deploy

Pré-requisitos

- Google Cloud SDK instalado
- Projeto GCP com billing habilitado
- APIs habilitadas: Cloud Run, Artifact Registry, Cloud Build

Build e Deploy (PowerShell)

```
# Variáveis
$PROJECT = "cadastra-yducs-prod"
$REGION = "southamerica-east1"
$REPO = "api-repo"
$SERVICE = "tiktok-ads-api"
$TAG = "1.0.0"
$IMAGE = "$REGION-docker.pkg.dev/$PROJECT/$REPO/${SERVICE}:$TAG"
$SA = "sa-cloud-run@$PROJECT.iam.gserviceaccount.com"

# Configuração
gcloud config set project $PROJECT
gcloud config set run/region $REGION

# Habilita APIs
gcloud services enable ` 
    run.googleapis.com ` 
    artifactregistry.googleapis.com ` 
    cloudbuild.googleapis.com

# Cria repositório (se não existir)
gcloud artifacts repositories create $REPO ` 
    --repository-format=docker ` 
    --location=$REGION

# Build da imagem
gcloud builds submit --tag $IMAGE

# Deploy no Cloud Run
gcloud run deploy $SERVICE ` 
    --image $IMAGE ` 
    --no-allow-unauthenticated ` 
    --service-account $SA ` 
    --port 8080 ` 
    --region $REGION ` 
    --memory 1Gi ` 
    --timeout 600 ` 
    --set-env-vars DAYS_REPROCESS=3
```

Permissões IAM

A Service Account do Cloud Run precisa:

```
# BigQuery  
roles/bigquery.dataEditor  
roles/bigquery.jobUser  
  
# Para invocar (Service Account do Composer)  
roles/run.invoker
```

Uso

Endpoints

Método	Endpoint	Descrição
GET	/health	Health check
POST	/run	Executa extração

Exemplo de Requisição

```
# Health check  
curl -H "Authorization: Bearer $(gcloud auth print-identity-token)" \  
      "https://tiktok-ads-api-xxxxx.run.app/health"  
  
# Execução  
curl -X POST \  
      -H "Authorization: Bearer $(gcloud auth print-identity-token)" \  
      -H "Content-Type: application/json" \  
      -d @sample_request.json \  
      "https://tiktok-ads-api-xxxxx.run.app/run"
```

Payload da Requisição

```
{  
  "access_token": "YOUR_TIKTOK_ACCESS_TOKEN",  
  "advertiser_ids": ["1234567890123456789"],  
  "project_id": "cadastra-yducs-prod",  
  "dataset_id": "raw",  
  "if_exists": "append",  
  "report_types": ["advertiser", "campaign", "adgroup", "ad"],  
  "start_date": "2025-01-01",  
  "end_date": "2025-01-15"  
}
```

Parâmetros

Parâmetro	Tipo	Obrigatório	Descrição
access_token	string	<input checked="" type="checkbox"/>	Token de acesso TikTok
advertiser_ids	array	<input checked="" type="checkbox"/>	Lista de IDs de anunciantes
project_id	string	Não	Projeto GCP (default: config)
dataset_id	string	Não	Dataset BigQuery (default: raw)
if_exists	string	Não	append/replace (default: append)
report_types	array	Não	Tipos de relatório (default: todos)
start_date	string	Não	Data inicial YYYY-MM-DD
end_date	string	Não	Data final YYYY-MM-DD

Resposta

```
{
  "status": "Ok",
  "message": "Extração concluída",
  "request_id": "uuid",
  "start_date": "2025-01-01",
  "end_date": "2025-01-15",
  "total_inserted_rows": 1500,
  "results": [
    {
      "advertiser_id": "1234567890123456789",
      "report_type": "campaign",
      "destination_table": "RAW.TKT002_TIKTOK_ADS_CAMPAIGN",
      "inserted_rows": 350,
      "status": "success"
    }
  ],
  "errors_count": 0
}
```

Airflow DAG

Localização

Copie o arquivo `dags/dag_tiktok_ads_to_bigquery.py` para o bucket do Cloud Composer:

```
gsutil cp dags/dag_tiktok_ads_to_bigquery.py \
  gs://BUCKET_COMPOSER/dags/
```

Agendamento

- **Schedule:** 0 8 * * * (diariamente às 08:00 UTC)
- **Período:** Últimos 3 dias (D-3 a D-1)
- **Retries:** 2 tentativas com 5 minutos de intervalo

Fluxo de Tasks

```
validate_config → prepare_payload → call_cloud_run_api → validate_result
```

Desenvolvimento Local

Setup

```
# Clone o repositório
cd tiktok-ads-api

# Crie ambiente virtual
python -m venv venv
source venv/bin/activate # Linux/Mac
.\venv\Scripts\activate # Windows

# Instale dependências
pip install -r requirements.txt
```

Teste Local com CSV (Recomendado para validação)

Para testar se a extração está funcionando **sem precisar do BigQuery**:

1. Abra o arquivo `run_local_csv.py`
2. Preencha suas credenciais:

```
CONFIG = {
    "access_token": "SEU_TOKEN_AQUI",
    "advertiser_ids": ["SEU_ADVERTISER_ID"],
    ...
}
```

3. Execute:

```
python run_local_csv.py
```

4. Os arquivos CSV serão gerados na pasta `output/`

Saída esperada:

```
📅 Período: 2025-01-13 a 2025-01-15
📁 Pasta de saída: /path/to/output

⌚ Processando Advertiser: 1234567890
    ✅ Extraiendo: ADVERTISER
    ✅ Salvo: TKT001_TIKTOK_AMS_ADVERTISER_123..._20250116.csv (50 linhas)
    ✅ Extraiendo: CAMPAIGN
    ✅ Salvo: TKT002_TIKTOK_AMS_CAMPAIGN_123..._20250116.csv (120 linhas)
    ...
    📋 RESUMO DA EXTRAÇÃO
    ✅ ADVERTISER | 50 linhas | success
    ✅ CAMPAIGN | 120 linhas | success
    ✅ ADGROUP | 340 linhas | success
    ✅ AD | 890 linhas | success
```

Execução Local (API Flask)

Para testar a API completa com BigQuery:

```
# Configure credenciais GCP
export GOOGLE_APPLICATION_CREDENTIALS="path/to/credentials.json"

# Define variáveis
export PORT=8080
export DAYS_REPROCESS=3

# Executa
python main.py
```

Teste da API

```
curl -X POST http://localhost:8080/run \
      -H "Content-Type: application/json" \
      -d @sample_request.json
```

Troubleshooting

Erros Comuns

Erro	Causa	Solução
401 Unauthorized	Token inválido	Verifique access_token
40100 Rate Limit	Muitas requisições	Aguarde e retry (automático)

Erro	Causa	Solução
BigQuery permission denied	IAM incorreto	Adicione roles à SA
Connection timeout	API TikTok lenta	Aumente timeout

Logs

```
# Cloud Run logs
gcloud logging read "resource.type=cloud_run_revision \
    AND resource.labels.service_name=tiktok-ads-api" \
    --limit 50

# Airflow logs
# Acesse UI do Composer → DAG → Task → Logs
```

Rate Limits TikTok

- 10 requisições por segundo por app
- 600 requisições por minuto por app
- A API implementa backoff automático

Estrutura do Projeto

```
tiktok-ads-api/
├── config/
│   ├── __init__.py
│   └── settings.py      # Configurações do projeto
├── controller/
│   ├── __init__.py
│   └── TikTokAdsController.py  # Lógica de extração
├── database/
│   ├── __init__.py
│   └── BigQuery.py      # Conexão BigQuery
├── dags/
│   └── dag_tiktok_ads_to_bigquery.py  # DAG Airflow
├── main.py              # Entry point Flask
└── __main__.py          # Entry point módulo
├── Dockerfile
├── .dockerignore
├── requirements.txt
└── README.md
```

Changelog

v1.0.0 (Janeiro 2025)

- Release inicial
 - Suporte a 4 níveis de relatório
 - Integração com BigQuery
 - DAG Airflow com Taskflow API
-

Contato

Para dúvidas ou sugestões, entre em contato com a equipe de Data Engineering.