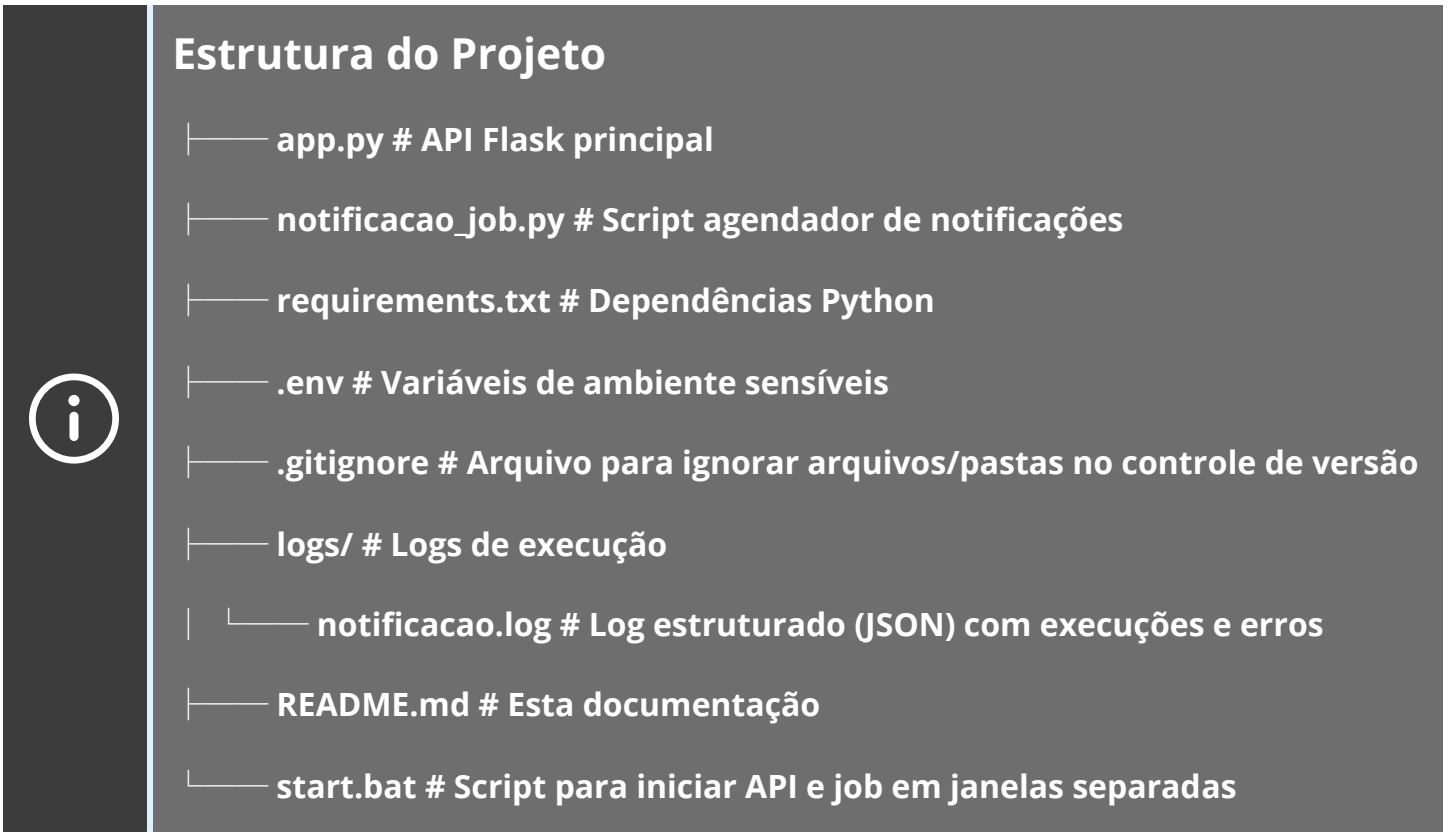
Documentação da API de Notificações via BLiP

Visão Geral

Esta API tem como objetivo enviar notificações via WhatsApp utilizando a plataforma BLiP. As mensagens são disparadas automaticamente a partir de registros em um banco Oracle, onde os contatos e mensagens são previamente cadastrados. A aplicação possui dois componentes principais:

- **API Flask (app.py)**: expõe um endpoint /send_notification que consulta o banco e dispara notificações via BLiP usando templates.
- **Script de job (notificacao_job.py)**: rotina agendada que chama periodicamente o endpoint da API para enviar notificações automaticamente.

Estrutura do Projeto



Tecnologias Utilizadas

- Python 3.9+
- Flask (API web)
- cx_Oracle (conexão Oracle)
- schedule (agendamento de tarefas)
- BLiP API (disparo de mensagens WhatsApp via template)
- dotenv (carregamento de variáveis de ambiente)
- logging (logs estruturados em JSON via notificacao_job.py e log padrão via app.py)

Variáveis de Ambiente (arquivo .env)

Configure as variáveis abaixo no arquivo .env na raiz do projeto:

env

DB_USER=usuario_oracle

DB_PASSWORD=senha_oracle

DB_DSN_HOST=host_oracle

DB_DSN_PORT=porta_oracle

DB_DSN_SERVICE=servico_oracle

URLBLIP_COMMANDS=https://http_blip_commands_url

URLBLIP_MESSAGES=https://http_blip_messages_url

KEYBLIP=TokenDeAutenticacao_BLIP

TEMPLATE_NAME=nome_template_mensagem

NAMESPACE=namespace_do_template

PORT=5000

Material de apoio - Blip Community

Como Rodar a API e o Job

1. **Instale as dependências:**

bash/powershell

pip install -r requirements.txt

1. **Configure o arquivo .env** com as variáveis corretas.
2. **Execute o script start.bat** (Windows):

.bat

start cmd /k "cd C:\Caminho até a pasta\notificacao_api && python app.py"

timeout /t 20

start cmd /k "cd C:\Caminho até a pasta\notificacao_api && python notificacao_job.py"

Isto abrirá duas janelas de prompt: uma com a API rodando e outra com o job agendado disparando notificações a cada 5 minutos.

Funcionamento Interno

API Flask (app.py)

- Endpoint: POST /send_notification
- A cada chamada, executa:
  - Consulta no banco Oracle por notificações pendentes (CD_STATUS = 1).
  - Para cada registro, normaliza o número de telefone.
  - Consulta a API BLiP para obter o identificador WhatsApp do cliente.
  - Envia mensagem usando template BLiP.
  - Atualiza o status do registro para 2 (enviado) em caso de sucesso.
  - Registra logs detalhados para monitoramento.

Job de Notificações (notificacao_job.py)

- Roda uma rotina que:
  - Verifica se a API está disponível.
  - Se disponível, chama o endpoint /send_notification a cada 5 minutos.
  - Produz logs estruturados em JSON para facilitar análise e monitoramento.
  - Em caso de erro na conexão, aborta e loga a falha.

Logs

- Logs da API (app.py) são gravados em logs/notificacao.log com formato tradicional.
- Logs do job (notificacao_job.py) são exibidos no console em formato JSON, contendo campos: timestamp, nível, mensagem, etc.

Erros Comuns e Troubleshooting

Erro	Possível Causa	Solução
Timeout na conexão com API	API não está rodando ou inacessível	Verifique se o app.py está ativo e endereço correto
Erro ao conectar Oracle	Dados DB incorretos ou indisponível	Confirme variáveis .env e status do banco Oracle
Falha no envio BLiP	Token inválido ou template errado	Confira KEYBLIP, TEMPLATE_NAME e NAMESPACE
Mensagens não são enviadas	Status do registro não atualizado	Verifique commits no banco e logs para erros

Considerações Finais

- Mantenha o .env seguro e nunca o envie para repositórios públicos.
- Aumente o tempo do timeout no job se o banco ou API forem lentos.
- Logs JSON podem ser importados em ferramentas como ELK para monitoramento avançado.