

## Requisitos de funcionamento:

1. Python 3.7+
2. Postgresql 12.5

## Instruções para instalação:

1. Instale as dependências em requirements.txt  

```
pip install -r requirements.txt
```
2. Edite, no arquivo de configurações do Yaml (por default chamado de 'config.yaml'), as configurações de usuário e endereço da base de dados Postgres da sua preferência.
3. Adicione a variável de ambiente 'MIDDLEMAN\_CONFIG', apontando para seu arquivo de configuração do Yaml.  
Exemplo: 

```
set MIDDLEMAN_CONFIG=C:\Projetos\MiddleManTest\config.yaml
```
4. Garanta acesso total a esse usuário na base de dados que você especificou.
5. Na linha de comando, no diretório "MiddleMan" execute os seguintes comandos:
  - a. 

```
python src/manage.py db init
```
  - b. 

```
python src/manage.py db migrate
```
  - c. 

```
python src/manage.py db upgrade
```
6. Pronto! Para rodar o servidor execute no mesmo diretório o comando:
  - a. 

```
python src/app.py
```

# Decisões de Implementação

Essa aplicação foi criada pensando não só em manter a coerência entre seus serviços, mas também com as outras api's com que trabalha, para isso foram tomados os seguintes cuidados:

1. Ao receber cada request são validados os arquivos recebidos, para garantir que contenham todas e apenas as informações necessárias. Para isso foi gerado o schema json de cada uma das mensagens esperadas.
2. Na persistência de ofertas foi adicionado a data de expiração das mesmas, para evitar tráfego desnecessário.
3. Por considerar que as ofertas oferecidas por nossos parceiros são baseadas nas informações do cliente que passamos a eles, não permitimos a alteração de informações relevantes do cliente enquanto houverem ofertas válidas.

Exemplo, um cliente pode corrigir seu email e telefone na hora de fazer a proposta, mas não seu salário, nome e idade, uma vez que esses podem mudar a oferta do parceiro.

4. Para manter a consistência e histórico preciso as propostas têm todas suas informações guardadas diretamente em sua própria tabela do banco de dados. Dessa forma um registro de cliente pode ser atualizado sem perder a precisão do histórico de propostas.
5. Para evitar problemas de duplicação de propostas um cliente é limitado a apenas uma proposta por parceiro pelo período especificado de dias.
6. Para facilitar a gerência pode-se mudar no arquivo de configuração:
  - a. as informações de por quantos minutos uma oferta permanece válida e;
  - b. intervalo mínimo de dias entre propostas aprovadas de um mesmo cliente e parceiro.