Teste para back-end N2

Desenvolva uma API REST usando Golang com as seguintes características.

- Use um repositório privado no GitHub para fazer seu teste, entre em contato para saber com quem você deve compartilhar seu repositório.
- Dada a seguinte estrutura de dados para conter clientes:

```
    "uuid" string contendo UUID
    "nome" string contendo nome do cliente
    "endereco" string contendo endereço do cliente
    "cadastrado_em" string contendo time stamp com data e hora do cadastro (ISO 8601 sem time zone)
    "atualizado_em" string contendo time stamp com data e hora da atualização (ISO 8601 sem time zone)
```

- Crie uma estrutura no PostgreSQL para conter os dados de clientes, essa estrutura deve usar um campo UUID como chave primária. Para carregar a estrutura no banco use a ferramenta de migrations do GoSidekick https://github.com/gosidekick/migration
- Crie os seguintes endpoints para manipular a estrutura anterior.
    - POST /cliente (vai servir para cadastrar o cliente, deve retornar o UUID do cliente cadastrado)
    - GET /cliente/{UUID} (vai servir para recuperar um JSON com o cliente cadastrado.
    - PUT /cliente/{UUID} (vai servir para alterar um cliente cadastrado)
    - DELETE /cliente/{UUID} (vai servir para remover um cliente da base)
    - GET /cliente (vai ser usado para listar todos os clientes)
- Crie um mecanismo para colocar em uma fila RabbitMQ os dados do último cliente cadastrado. (você pode usar outros tipos de fila se preferir, desde que ela suba isoladamente em um container).
- Crie um micro-serviço que coleta a última entrada da fila e salva em um arquivo JSON em um diretório configurado pela variável de ambiente NOVOS_CLIENTES. Sendo um arquivo para cada novo cliente.
- Lembre de testar o seu código com testes unitários, esperamos pelo menos 80% de cobertura de testes.
- A stack deve rodar em containers Docker e idealmente ser carregada com docker-compose.
- Não esqueça de documentar seu código.