
M2C Digital - Teste Backend Senior.

Questionário

1. Descreva com suas próprias palavras quais rotinas tem o menor desempenho em termo de performance no momento do processamento de um dado.
2. Quais linguagens são mais utilizadas para processamento de dados em alta performance?
3. Quais linguagens realmente executam um paralelismo real e quais não?
4. Descreva com suas palavras o fluxo de uma aplicação que deve enviar milhões de SMS em um curto espaço de tempo (Por exemplo 5 minutos) e gerar os relatórios destes envios para apresentar para o cliente. Não preciso de código. Quero uma análise detalhada de como faz isso.

Proficiência em Desenvolvimento de Aplicações.

1. Crie uma aplicação em NodeJS.
 - a. Será utilizado para esta aplicação NestJS para a API, NodeJS para o consumer de uma fila em rabbitmq. Será utilizado Mongo DB e PostgreSQL.
 - b. PostgreSQL será utilizado para armazenar os dados relacionais. Mongodb dados não relacionais.
-



c. Tabelas dos dados Relacionais: Users, Companies, Campaigns

users

id: string (gerado com library cuid2)
email: string
password: string (criptografado com crypto por exemplo)
created_at: Date
updated_at: Date
deleted: bool

companies

id: string (gerado com library cuid2)
name: string
document: string
created_at: Date
updated_at: Date
deleted: bool

campaigns

id: string (gerado com library cuid2)
name: string
created_at: Date
updated_at: Date
deleted: bool

d. Tabelas dos dados Não relacionais: Messages

messages

identifier: string (gerado com library cuid2)
phone_number: string
message: string
campaign_id: string
created_at: Date
updated_at: Date
deleted: bool

e. Funcionamento do Sistema:

- i. O sistema (**API NestJS**) deve ter um crud para usuários, empresas onde serão cadastrados os dados (**PostgreSQL**).
- ii. O sistema (**API NestJS**) irá criar um sistema de autenticação para que possa acessar a rota de crud de campanha.
- iii. O sistema (**API NestJS**) deve permitir delete desde que seja soft delete.
- iv. A partir do cadastro da empresa e usuário, o sistema irá criar uma campanha e enviará um arquivo texto com os números. (**API NestJS**)
- v. Quando a (**API NestJS**) receber esse dado em seu crud, ela irá criar essa campanha no banco de dados **PostgreSQL** e irá fazer um loop para enviar essas mensagens a uma fila do rabbitmq para que o sistema de envio (consumer construído em **NodeJS** puro com typescript) receba essa mensagem e grave no banco de dados **MongoDB** e atualize o registro da campanha no **PostgreSQL** para enviado.
- vi. As mensagens da campanha devem ser enviadas ordenadamente e criadas no banco apenas 1 vez para cada campanha. Se enviou para 5 números, você terá 5 mensagens no Mongoddb.

2. Crie uma aplicação em Rust

Para a aplicação em Rust, quero que seja feito a mesma coisa do consumer em nodejs.

- a. Quando a **(API NestJS)** receber esse dado em seu crud, ela irá criar essa campanha no banco de dados **PostgreSQL** e irá fazer um loop para enviar essas mensagens a uma fila do rabbitmq para que o sistema de envio (consumer construído em **NodeJS** puro com typescript) receba essa mensagem e grave no banco de dados **MongoDB** e atualize o registro da campanha no **PostgreSQL** para enviado.

3. Crie uma aplicação em Go Lang

Para a aplicação em Go Lang, quero que seja feito a mesma coisa do consumer em nodejs.

- a. Quando a **(API NestJS)** receber esse dado em seu crud, ela irá criar essa campanha no banco de dados **PostgreSQL** e irá fazer um loop para enviar essas mensagens a uma fila do rabbitmq para que o sistema de envio (consumer construído em **NodeJS** puro com typescript) receba essa mensagem e grave no banco de dados **MongoDB** e atualize o registro da campanha no **PostgreSQL** para enviado.

4. Entrega

Não é preciso frontend para esta tarefa. A entrega se dará ao seguinte formato:

- Repositórios no Github
- Collection do Postman com todos os endpoints e seus devidos payloads.
- Aplicação rodando em ambiente local.
 - Deverão ser criados os banco de dados usando docker para eu rodar isto localmente.
 - As aplicações consumers deverão criar a fila automaticamente a partir do Rabbit DSN (String de conexão do rabbit) que eu passar. Isto é, quando eu iniciar a aplicação ela deve ler meu .env para que eu possa estar rodando a aplicação e criando a fila no CloudAMQP por exemplo.
 - Deixe as linhas de comandos de cada aplicação escritas no readme de cada uma delas. Quero saber tudo que preciso para rodar seu código no readme.



- Envio da conclusão e respostas via Email.