Desafio Engenheiro de Dados

Este desafio consiste em **3** questões divididas em **questões teóricas** e **questões práticas**.

# Questão Teórica

## Arquitetura de Dados

### Questão 1

Projete uma solução para uma plataforma de comunicação que computa a **utilização** de **notificações Whatsapp** de milhares de clientes **a cada minuto**.

A interface gráfica é um painel simples, apresentando alguns dados conforme o exemplo abaixo:

**Painel de Notificações**

Cliente: Acme

Notificações Whatsapp: 5000

Última atualização: 2020-02-03T15:00:00+00:00

As notificações são processadas **a cada minuto** e podem ser exportadas em um relatório detalhado conforme exemplo abaixo:

**Relatório Detalhado de Notificações**

= Cliente: Acme =

Mensagem ID: 5B81FF24A1

Payload: “Olá XXXXX, seja muito bem-vindo! Precisa de alguma ajuda?”

Data hora: 2020-02-03T14:00:01.001+00:00

...

Mensagem ID: A61D178E73

Payload: “Não identificamos a sua última mensagem. Por favor, entre em contato”

Data hora: 2020-02-03T14:00:01.099+00:00

...

Além disso, a solução também deve possibilitar a geração de um **relatório de cobrança uma vez por dia**, conforme exemplo abaixo:

**Relatório de Cobrança**

Cliente: Acme

Notificações Whatsapp: 25000

Valor: R$ 1000,00

Data: 2020-02-03

A origem dos dados é um tópico no **Kafka**, com um volume de **100 milhões de mensagens/dia**, com o seguinte payload:

+-----------+----------+-------------+---------+------------------------------------------+-----------------------------+

|customer\_id|message\_id|customer\_name|channel |payload |event\_datetime |

+-----------+----------+-------------+---------+------------------------------------------+-----------------------------+

|7494212 |5B81FF24A1|Acme |Whatsapp |Olá XXXXX, seja muito bem-vindo! Precis...|2020-02-03T14:00:01.001+00:00|

|7494212 |A61D178E73|Acme |Whatsapp |Não identificamos a sua última mensagem...|2020-02-03T14:00:01.099+00:00|

...

Proponha uma arquitetura que contemple cada um dos casos de uso: painel de notificação, relatório detalhado de notificações e relatório de cobrança.

Nota: Esta é uma questão teórica, não é necessário implementação.

# Questões Práticas

* Compartilhe um zip ou um repositório privado no GitHub com o código fonte
* Descreva como executar cada serviço em um arquivo Markdown
* Implemente os serviços na linguagem de sua preferência

## Serviços de Agregação de Dados com Entradas CSV ou JSON

### Questão 2

Com base no arquivo **invoices.csv**, escreva um serviço que computa **a média de faturamento de cada conta (account) nos últimos três e seis meses** retroativos à **Jan/2020**. Quando não há dados suficientes na janela de 3 ou 6 meses o serviço registra a entrada como null.

Resultado esperado:

+--------+-------+--------------------------+--------------------------+

|customer|account|avg\_invoices\_last\_3\_months|avg\_invoices\_last\_6\_months|

+--------+-------+--------------------------+--------------------------+

|C1000 |A1100 |56.333,33 |41.416,66 |

|C1000 |A1200 |50.666,66 |null |

|C1000 |A1300 |null |null |

|C2000 |A2100 |55.400,00 |null |

+--------+-------+--------------------------+--------------------------+

### 

### 

### Questão 3

Uma plataforma de comunicação fornece fluxos de conversação (chatbots) entre outras funcionalidades. O data lake desta plataforma armazena valores informados pelos usuários em um formato semiestruturado (JSON) particionado por hora:

* hour=13.json
* hour=14.json

Considere que a seção **content** mantém as respostas de usuários. Ou seja, os valores preenchidos pelos usuários durante uma conversa. Para isto, ela armazena mapas onde a chave e valor são do tipo String.

Implemente um serviço que gera um relatório consolidado das últimas respostas informadas pelos usuários na hora 13 e hora 14 no seguinte formato:

+--------+-----+-------+-------------------+-------------------+-----+--------------+------------------+

|customer|flow |session|first\_answer\_dt |last\_answer\_dt |name |cpf |delivery\_confirmed|

+--------+-----+-------+-------------------+-------------------+-----+--------------+------------------+

|C1000 |F1000|S1000 |2019-12-16T13:59:58|2019-12-16T14:00:01|maria|305.584.960-40|true |

|C1000 |F1000|S2000 |2019-12-16T13:59:59|2019-12-16T14:00:00|joao |733.600.420-26|false |

+--------+-----+-------+-------------------+-------------------+-----+--------------+------------------+

Os campos **first\_answer\_dt** e **last\_answer\_dt** representam, respectivamente, a primeira e última interações válidas (diferente de vazio). No exemplo acima, os campos *name*, *cpf* e *delivery\_confirmed* são as respostas do usuário.

Para concluir, **o serviço deve ser agnóstico de conversa**, ou seja, deve suportar conteúdo de qualquer fluxo. Exemplo:

+--------+-----+-------+-------------------+-------------------+---------+----+

|customer|flow |session|first\_answer\_dt |last\_answer\_dt |recomenda|nota|

+--------+-----+-------+-------------------+-------------------+---------+----+

|C2000 |F2000|S3000 |2019-12-16T13:59:59|2019-12-16T14:00:01|Simmmmmmm|9 |

+--------+-----+-------+-------------------+-------------------+---------+----+

Neste exemplo, *recomenda* e *nota* são as respostas do usuário.