

Este desafio consiste em **15** questões divididas em:

* ​**11 questões teóricas**​
* **4 questões práticas**​

Sumário

[1. Questões Teóricas 3](#_Toc57109083)

[1.1 Engenharia de Dados 3](#_Toc57109084)

[ Questão 1 3](#_Toc57109085)

[ Questão 2 3](#_Toc57109086)

[ Questão 3 3](#_Toc57109087)

[ Questão 4 3](#_Toc57109088)

[ Questão 5 3](#_Toc57109089)

[ Questão 6 3](#_Toc57109090)

[ Questão 7 4](#_Toc57109091)

[ Questão 8 4](#_Toc57109092)

[ Questão 9 4](#_Toc57109093)

[1.2 Arquitetura de Dados 4](#_Toc57109094)

[ Questão 10 4](#_Toc57109095)

[ Questão 11 6](#_Toc57109096)

[2. Questões Práticas 6](#_Toc57109097)

[2.1 Serviços de Exploração de Dados 7](#_Toc57109098)

[ Questão 1 7](#_Toc57109099)

[ Questão 2 7](#_Toc57109100)

[2.2 Serviços de Agregação de Dados com Entradas CSV ou JSON 9](#_Toc57109101)

[ Questão 3 9](#_Toc57109102)

[ Questão 4 10](#_Toc57109103)

# Questões Teóricas

Por favor, envie para por email ‘rh@stitdata.com’ as respostas das seguintes questões teóricas:

## Engenharia de Dados

### Questão 1

Quais são as responsabilidades de um Engenheiro de Dados ?

R: O engenheiro de dados tem como principais responsabilidades a criação de processos de ingestão de dados, bem como a criação de pipelines que transformam dados brutos que podem estar nos mais diversos formatos (Estruturados, Semi-estruturados, não-estruturados) vindos de fontes de dados como banco de dados, arquivos de logs ou redes sociais em um formato para consumo dessas informações. Assim como garantir as execuções desses pipelines de acordo com a necessidade do negócio e promover soluções de formar escalável.

### Questão 2

Quais são as habilidades necessárias para um Engenheiro de Dados ?

R: O engenheiro de dados necessita ter as habilidades para arquitetar sistemas distribuídos, elaborar pipelines confiáveis; combinar diversas fontes de dados, criar e arquitetar soluções big data; Coloborar com os times de data analysis e data scients para gerar valor para o negócio, seja para tratar de dados para gerar informação para tomada de decisão ou criação de informações para trabalhar com modelos preditivos.

### Questão 3

Que tipos de arquiteturas você conhece para DW e Big Data?

R: DW – Camada única, Dupla camada e tripla camada.

Big Data – Arquitetura Lambda e Arquitetura Kappa

### Questão 4

Qual o diferencial de um DW e um EDW ?

R: Data Warehouse (DW) e Enterprise Data Warehouse (EDW) são termos utilizados para se tratar da tecnologia referente armazenamento de dados oriundos de sistemas OLTP, porém eles se diferenciam no conceito aonde são aplicados. DW tem como essência ter um repositório de dados de transações históricas e atuais de uma organização com a interpretação de um determinado setor. Já EDW é um armazém centralizado, que fornece o serviço para toda a empresa, a ideia é apresentar uma única fonte de dados da “Verdade” que contem informações de toda empresa (marketing, vendas, finanças, Recursos Humanos).

### Questão 5

O que uma sand-box no ambiente de Big Data e qual seu diferencial ?

R: Um sandbox, no contexto de big data, é uma plataforma escalonável de desenvolvimento usada para explorar conjuntos de dados de uma organização por meio de interação e colaboração. Oferecendo uma capacidade computacional necessaria para que, por exemplo, um cientista de dados possa lidar com uma carga de dados complexa.

### Questão 6

Quais são as camadas de um Data Lake?

R: Para a construção de um data lake são necessárias quatro camadas, e organizadas por assunto (exemplo: RH, Marketing, Vendas e etc.), são elas:

1º Camada - Staging – Essa camada tem com intuito armazerna dados de formar temporária, logs, spools de streaming e outros dados de curta duração. O é ideial criar uma rotina de expurgo de dados.

2º Camada – Raw Data – Essa camada tem com objetivo receber o dado de formar bruta, ou seja, dados Estruturatos, semi-estruturas e não-estrurados como XML, EXCEL, CSV, export de banco de dados e etc.

3º Camada – Trusted Data – Nessa camada são armazenados os dados que foram tratados para um entidade centralizada e aplicado o data quality e a melhor nomenclatura para os campos.

4º Camada – Refined – Ultima camada de armazenamento do data lake, nessa camada são armazenados os dados já enriquecido e prontos para o consumo.

### Questão 7

Em que fases atuam o Arquiteto de Dados , Data Science, Engenheiro de Dados e Consultor Analytics ?, e como os perfis se se integram ?

R: Arquiteto de dados é o responsável por visualizar e projetar a estrutura de gerenciamento de dados de uma organização. Geralmente esse profissional coordena e planeja todo o ciclo de vida do dado.

Já o Engenheiro de dados tem com principais atividades criar e desenvolver rotinas de ingestão de dados, e criação de pipeline automatizados com rotinas de tratamento e enriquecimento de dados seguindo estrutura arquitetada pelo arquiteto de dados e disponibilizando essas informações para consultas para as área de Data  Analytics e Data Science.

O Consultor de Data Analytics tem como objetivo analisar os dados do presente e passado para responder perguntas do negócio e ajudar na tomada de decisão.

A área de Data Science possui diversas ramificações de atuação, porém de modo objetivo e simplista, tem como objetivo aplicação de análise exploratória e criação de modelos preditivos.

### Questão 8

Qual a diferença de um Data Lake e um DW?

R: Data lake é uma estrutura de Big Data que tem como objetivo receber o dado bruto de diversos tipos e origens. Com objetivos de criar e tratar para o consumo das áreas de Data Analytics e Data Science. O Data Warehouse tem como objetivo armazenar de forma dedicada uma grande volume de dados que foram tratados, padronizados e higienizados, trazendo assim uma proposta de oferecer visões da organização que possam contribuir na melhor tomada de decisão.

### Questão 9

O que é arquitetura Lambda e Kappa?, como pode ser feito um mix destas duas? exemplifique um cenário em que possa ser aplicado as duas.

R: Arquitetura Lambda e Kappa são arquiteturas desenhadas para um ecossistema de Big Data, ou seja, arquiteturas para trabalhar-se com uma quantidade massiva de dados, sendo eles dos mais diversos tipos e formatos.

A arquitetura Lambda é uma técnica de processamento de dados capaz de lidar com uma quantidade enorme de dados (Terabytes e Petabytes) de maneira eviciênte e escalável. A implementação desse modelo de processamento de dados é aplicado aonde as consultas do usuário devem ser atendidas em uma base ad-hoc usando armazenamento de dados imutável. Além de garantir formar de seguras de atualização da base de dados.

A arquitetura Lambda é composta por Batch Layer, Speed ​​Layer e Serving Layer.

Batch Layer: é camada que recebe os dados de formar bruta ou revinados e enriquecidos, antes da ingestão desses dados na camada de Serving Layer.

Speed Layer: Camada que recebe o fruto de algum evento programado de processamento da camada Batch que resulta na atualização de algum processo Delta ou MapReduce em um tempo quase real (near-real time).

Serving Layer: Camada de saída dos lotes de processamento de Batch e Speed Layer na formar de visualizações.

A arquitetura Lambda trás grandes benefícios são ele alta disponibilidade, escalabilidade, tolerância a falha, armazenamento distribuído, confiabilidade, alta velocidade de processamento e baixa latência. Como pontos contra a esse modelo de arquitetura temos alta complexidade na criação dos scripts, processos automatizados ou migração. Construção de código verbosos e complexos.

Arquitetura Kappa tem como objetivo trabalhar com dados em tempo real e ainda possibilita uma arquitetura mais simples que utiliza menos uso de código.

A arquitetura Kappa é composta por Streaming Layer e Serving Layer.

Streaming Layer – Essa camada recebe os dados oriundos de eventos que são acionados por triggers. Nessa camada são realizadas os tratamentos das informações que são entendidas com relavantes.

Serving Layer – Nessa camada são disponibilizados as informações tratados em forma de visualizações.

Arquitetura Kappa tem com pontos positivos a possibilidade de criação de sistema online que não precisam de camada de lote, alta disponibilidade de informações (in-memory), ou seja, o data é persistido em memória não necessita de tantos recursos. Como contra a ausência da camada de Batch pode ocasionar erros durante o processamento ou atualização do banco de dados o que necessita de um controle maior sobre o tratamento de exceções.

É possível tem uma solução que integre ambos os modelos, por exemplo, pode ser construído uma arquitetura Lambda pensando em processos massivos de dados sofre as informações de negócio e em outra ponta uma arquitetura Kappa que recebe informações de rede social em tempo real. Ou seja, pensando num negócio como varejo, que possua uma campanha promocional numa semana temos a possibilidade de ter as informações sobre venda dos itens dessa campanha através de visões de dados coletas pela arquitetura Lambda e podemos analisar o que estão dizendo sobre a marca ou da campanha nas redes sociais para realizar uma analise de sentimento e entender melhor o consumidor ou algum processo que necessite de melhoria.

## Arquitetura de Dados

### Questão 10

Projete uma solução big data para uma plataforma de comunicação que computa a **utilização**​de​ **notificações Whatsapp** de milhares de clientes **a**​ **cada minuto**.​ Suponha que a interface gráfica seja um painel bem simples. Exemplo:

**Painel de Notificações**

Cliente: Acme

Notificações Whatsapp: 5000

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

Assuma que as notificações processadas **a**​ **cada minuto** possam ser exportadas em um relatório detalhado:

**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, considere que a solução também gera um **relatório**​ **de cobrança uma vez por dia**.​ Exemplo:

**Relatório de Cobrança**

Cliente: Acme

Notificações Whatsapp: 25000

Valor: R$ 1000,00

Data: 2020-02-03

Assuma que a origem dos dados é o **Kafka**​ ​. Mais especificamente, a entrada de dados é um tópico com **100**​ **milhões de mensagens/dia** com o seguinte payload:​

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

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

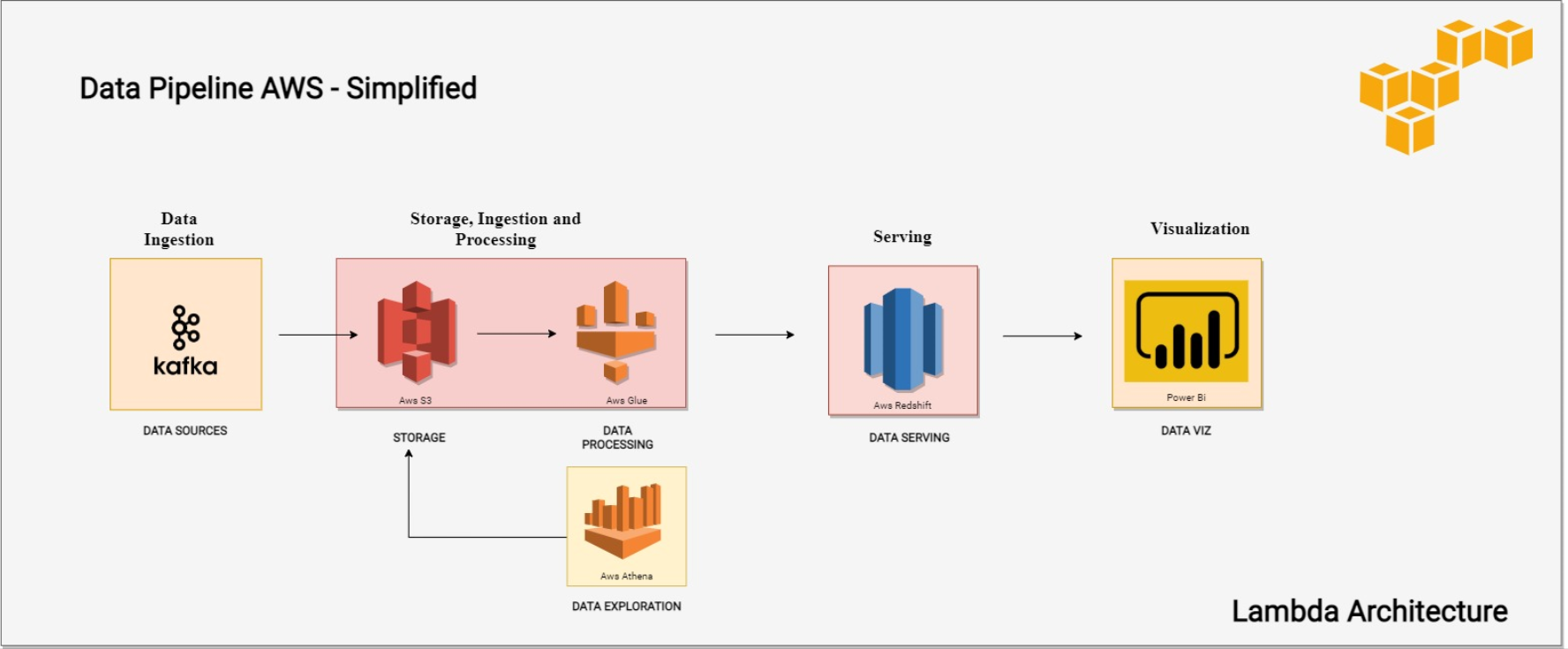
+-----------+-----------+-------------+---------+------------------------------------------+-----------------------------+

|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| ...

Em suma, proponha **uma arquitetura com data lake, serviços analíticos e bases de dados**​ .​

Idealmente desenhe um pipeline de dados para cada caso de uso: painel de notificação, relatório detalhado de notificações e relatório de cobrança.

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



### Questão 11

Com base **na**​ **resposta** dada para a Questão​ 10 especifique o(s) esquema(s) necessário(s) para armazenar as informações nos bancos de dados.

R: Criaria dois esquema, um schema de Stage e outro de Prod. Cada um com três tabelas (painel\_notificacao, notificação\_detalhada e notificacao\_cobracao), o schema Stage seria usado como área temporária para validação de duplicidade de linhas no schema de Prod, e em caso verdadeiro seria aplicado o replace desses dados para a versão mais atual e o resto dos dados seria feito insert para o schema de Prod. Após finalização da carga no schema Prod os dados do schema Stage seriam apagados.

# Questões Práticas

Por favor, envie um zip com o código fonte que resolve os seguintes desafios.

Outras Instruções

* Descreva como executar cada serviço em um arquivo Markdown (.md)
* Implemente os serviços (scripts) em Pandas, Dask ou Spark

## Serviços de Exploração de Dados

### Questão 1

Um plataforma de comunicação registra eventos de execuções de chatbots. Escreva um serviço que computa **quantos clientes executaram entre 1000 e 2000 chatbots.**

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

|customer\_id|chatbot\_id|event\_date\_time|

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

|7494212 |1000 |1535308430 |

|7494212 |2000 |1535308433 |

|1475185 |3000 |1535308444 |

|6946725 |4000 |1535308475 |

|6946725 |5000 |1535308476 |

|6946725 |6000 |1535308477 |

…

### Questão 2

Suponha que você está explorando dados de cobrança de chatbots de Jan/2020 e tem em mãos dois conjuntos de dados pré-processados. O primeiro armazena os clientes e respectivos chatbots utilizados naquele mês.

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

|customer\_id|chatbot\_id|chatbot\_type |

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

|7494212 |1000 |Pesquisa de satisfação |

|7494212 |2000 |Confirmação de agendamento|

|1475185 |3000 |Negociação de dívida |

|6946725 |4000 |Segunda via de fatura |

|6946725 |5000 |Pesquisa de satisfação |

...

O segundo mantém os valores de cobrança dos chatbots:

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

|chatbot\_id|cost |

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

|1000 |200,0 |

|2000 |100,0 |

|3000 |1000,0|

|4000 |50,0 |

|5000 |400,0 |

...

Escreva um serviço que imprime os clientes os quais **a média do custo é maior que R$ 500,00.**​

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

Você deve ter recebido três arquivos:

* invoices.csv
* hour=13.json
* hour=14.json

### Questão 3

Dado o 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 NaN (null). A saída esperada é a seguinte:

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

|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 |NaN |

|C1000 |A1300 |NaN |NaN |

|C2000 |A2100 |55.400,00 |NaN |

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

### Questão 4

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. Pragmaticamente falando, ela armazena mapas onde chave e valor são String.

Implemente um serviço que gera um relatório consolidado das últimas respostas

+--------+-----+-------+-------------------+-------------------+-----+--------------+------------------+ |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|sim |

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

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

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*​ ​ aparecem como respostas do usuário.