Estudo de caso para um problema de integração

# Charles Guimarães Cavalcante - RM 334409

#### **Problema a ser resolvido**

A empresa conta com um aplicativo de compras domésticas mensais, onde o usuário insere a sua lista de compras, indicando a quantidade e periodicidade de compra de cada item. O sistema irá verificar o preço de cada produto nas lojas parceiras e indicar o melhor plano de compra, minimizando o custo total da compra, por exemplo, em uma lista de 30 itens, indicar a compra de 15 deles na loja X, 10 na loja Y e 5 na loja Z.

Os dados precisam ser disponibilizados para o time de BI o mais rápido possível, para que sejam tomadas decisões para o marketing digital.

#### **Fontes de dados**

Bancos de dados relacionais e NoSQL, hosteados em ambiente cloud.

Gravações de ligações no atendimento, hosteadas em repositório em ambiente cloud.

São gerados cerca de 50 Gb de dados por dia.

#### **Arquitetura de dados Lambda ou FastData?**

O ideal para este projeto é a arquitetura lambda, pois trabalhar com conjuntos de dados muito grandes, pode levar muito tempo para executar a classificação de consultas de que os clientes precisam. Essas consultas não podem ser executadas em tempo real e geralmente exigem algoritmos como MapReduce, que operam em paralelo em todo o conjunto de dados. Os resultados são então armazenados separadamente dos dados brutos e usados para consulta.

Todos os dados recebidos pelo sistema passam por dois caminhos:

Uma camada de lote (batch layer) armazena todos os dados de entrada em sua forma bruta e executa o processamento em lotes nos dados. O resultado desse processamento é armazenado como uma exibição de lote.

Uma camada de velocidade (speed layer) analisa os dados em tempo real. Essa camada é projetada para baixa latência, em detrimento da precisão.

A camada de lote alimenta uma camada de serviço que indexa a exibição de lote para uma consulta eficiente. A camada de velocidade atualiza a camada de serviço com atualizações incrementais de acordo com os dados mais recentes.