

Desafio **dito**

A Dito coleta bilhões de eventos de comportamento on/offline e um dos desafios diários é gerar valor para os usuários através de informações consistentes e acessíveis.

Esse desafio é composto por 2 partes: um **Serviço de Autocomplete** e um problema de **Manipulação de Dados**.

Serviço de Autocomplete

O serviço deve conter uma **API Coletora** de dados e o mecanismo de **Autocomplete** propriamente dito.

API Coletora:

Você deverá construir uma API para coletar e armazenar os dados.

Esta API deverá receber informações de navegação dos usuários em um site. Um exemplo seria:

```
{
  "event": "buy",
  "timestamp": "2016-09-22T13:57:31.2311892-04:00"
}
```

Autocomplete:

O mecanismo de autocomplete deve ser implementado e disponibilizado através de uma API, contendo um campo de busca que deverá completar o nome dos eventos a partir da segunda letra que o usuário digitar.

Manipulação de Dados

O objetivo é criar uma timeline de compras a partir dos eventos disponíveis neste endpoint: <https://storage.googleapis.com/dito-questions/events.json>.

Um evento representa um comportamento de uma pessoa, seja no mundo online ou offline. Quando uma pessoa faz uma compra, um evento **comprou** é gerado contendo o total de receita gerada e o nome da loja. Para cada produto dessa

compra é gerado um evento **comprou-produto**, contendo o nome e preço do produto.

Você deve implementar uma função, em qualquer linguagem de programação, que consuma esse endpoint e agrupe as compras pelo campo **transaction_id**. Cada item da timeline deve representar uma compra em uma determinada loja e deve conter uma **lista** com os produtos comprados.

A timeline deve ser **ordenada** pelo campo **timestamp** na ordem decrescente.

A resposta esperada dessa função é a seguinte:

```
{
  "timeline": [
    {
      "timestamp": "2016-10-02T11:37:31.2300892-03:00",
      "revenue": 120.0,
      "transaction_id": "3409340",
      "store_name": "BH Shopping",
      "products": [
        {
          "name": "Tenis Preto",
          "price": 120
        }
      ]
    },
    {
      "timestamp": "2016-09-22T13:57:31.2311892-03:00",
      "revenue": 250.0,
      "transaction_id": "3029384",
      "store_name": "Patio Savassi",
      "products": [
        {
          "name": "Camisa Azul",
          "price": 100
        },
        {
          "name": "Calça Rosa",
          "price": 150
        }
      ]
    }
  ]
}
```

Premissas

- A solução deve ser escalável, preparada para receber um grande volume de requisições;
- As APIs do serviço de autocomplete devem responder num tempo satisfatório, que não comprometa a experiência dos usuários, mesmo com um grande volume de dados;
- O desenvolvimento pode ser feito utilizando qualquer linguagem de programação e qualquer sistema de banco de dados, cabendo ao desenvolvedor escolher o que for melhor para a situação;
- Serão avaliados a arquitetura, os padrões utilizados para o desenvolvimento e a entrega no prazo.