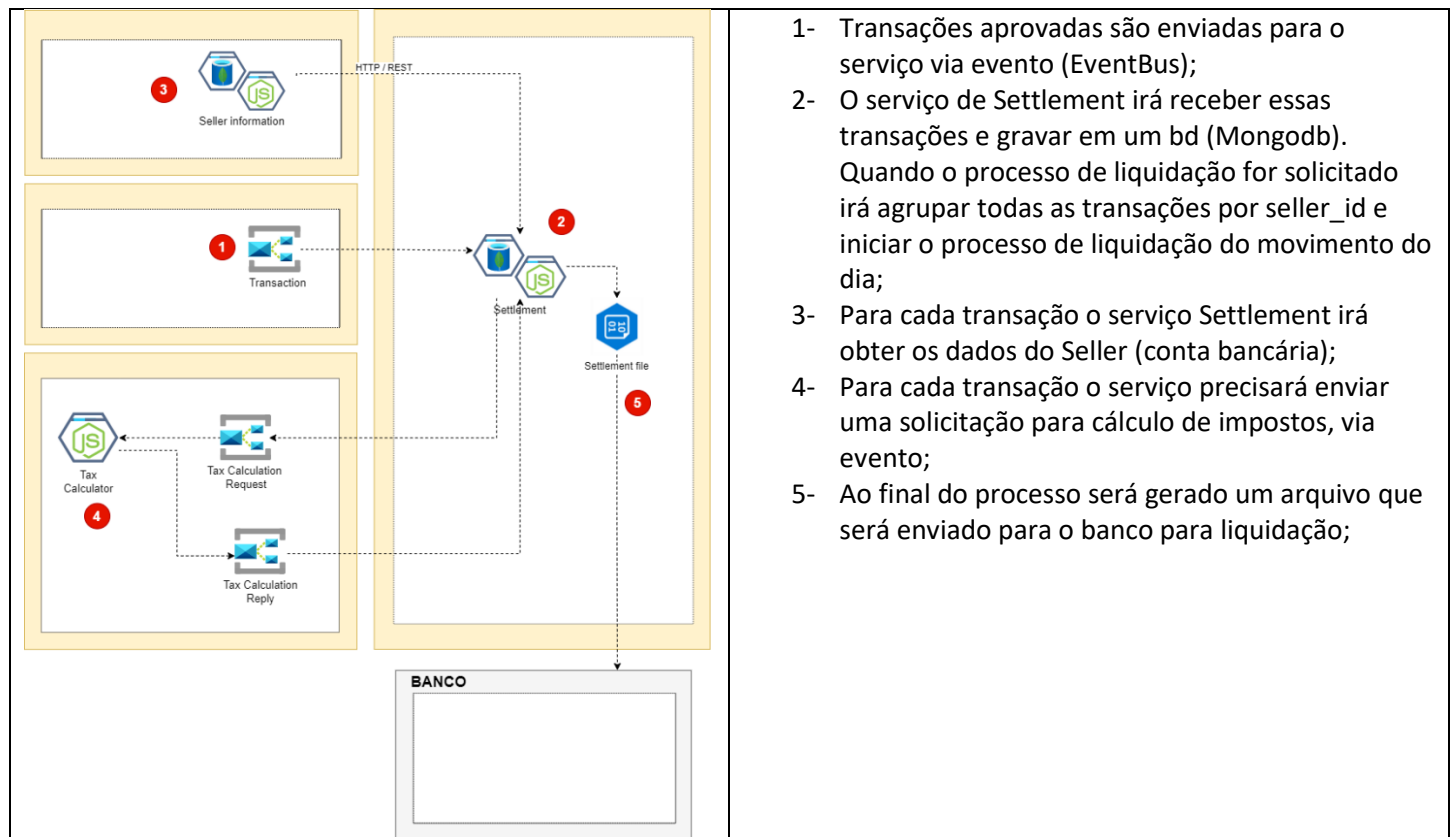


Nosso novo objetivo agora é escrever o Microserviço Settlement, quando esse Microserviço receber uma solicitação via API, deve consultar a lista de transações no Mongodb agrupado por seller\_id, para cada seller\_id realizar o processo para cálculo de imposto (via tópico usando o tax\_calculator), obter o número da conta bancária do seller (via API utilizando a API seller\_information) e então escrever um arquivo para enviar para o banco, para a liquidação desse movimento.

## Cenário

Lembrando, que o desenho da solução é o seguinte:



**NA AULA:  $2+2=4$   
LIÇÃO DE CASA:  $700+300=1000$**



**NA PROVA:  
EXISTEM 2 CAVALOS VOADORES  
UM AZUL, E O OUTRO FOI PARA A DIREITA,  
QUAL O PESO DA GIRAFA, SABENDO  
QUE A FORMIGA ANDA ?**



NÃO ENTRE AQUI

## 1- Criar o projeto

Criar um novo projeto dentro do repositório, chamado settlement.

### MongoDb

Lembrando que cada Microserviço deve ter o seu próprio Banco de dados, não compartilhe o mesmo mongo com o outro serviço. Nesse mongo vamos ter as seguintes collections:

#### **/transactions**

Representa as transações de venda que vão ser processadas

```
{
  "transactionId": "c1e42054-c08f-4cb0-b893-1ebfdbad5719",
  "timeStamp": "2022-03-01 08:00",
  "sellerId": 1,
  "settlementDate": "2022-03-01",
  "amount": 2000
},
{
  "transactionId": "c1e42054-c08f-4cb0-b893-1ebfdbad5719",
  "timeStamp": "2022-03-01 08:00",
  "sellerId": 1,
  "settlementDate": "2022-03-01",
  "amount": 3000
}
```

#### **/settlements**

Representa o lote de liquidação que está sendo processado

```
{
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "settlementDate": "2022-03-01",
  "startDate": "2022-03-24 08:00",
  "endDate": "2022-03-24 09:10",
  "sellersCount": 10,
  "transactionsCount": 100,
  "elapsedMilliseconds": 19292
}
```

#### **/sellerSettlements**

Representa o total de transações, o imposto calculado e o número da conta para o seller

```
{
  "sellerId": 1,
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "amount": 5000,
  "taxValue": 300,
  "bankCode": 033,
  "bankAccount": 23456
}
```

## Massa de testes

Vamos criar endpoints que carregam o Mongo com dados de teste.

Na API **seller\_information**, criar o método: POST **/v1/sellers/dummy-data**

Quando executado deve criar 10 sellers (seller\_id de 1 até 10), com dados gerados

Futuramente vamos fazer isso com 100 sellers para testar.

Na API aqui de **settlement**, criar o método: POST **/v1/transactions/dummy-data**

Quando for executado deve gerar 100 transações para cada um dos sellers, totalizando 1.000 (gerar o transaction\_id utilizando guid). Preencha todas as transações com settlementDate: “2022-03-01” é a data fixa que vamos utilizar para testar.

Inclua no README esses passos, por exemplo: Depois de rodar “docker-compose up”, abrir o browser e rodar a chamada 1, e depois a chamada 2, ...

NOTA: A ideia é depois rodar esses métodos com uma massa maior: 1K sellers ao invés de 100 e 1M de transações ao invés de 1K.

## 2- Iniciar a liquidação

### POST /v1/settlements

Essa solicitação, deve fazer o seguinte:

- Criar um guid que será a settlementId
- Obter a data atual (startDate)
- Gravar no mongo (collection settlements) os dados e retornar eles, assim:

```
{
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "settlementDate": "2022-03-01",
  "startDate": "2022-03-24 08:00",
  "endDate": null
}
```

Esse registro identifica qual a chave do processo e quando ele começou.

Essa chamada da API continua com as etapas a seguir.

### Consolidar as transações por seller\_id

Obter do mongo (collection transactions) todas as transações com os seguintes parâmetros: settlementDate: 2002-03-01. E deve agrupar o movimento total por essa chave, por exemplo se tiver duas transações para o seller\_id 1:

```
{
  "transactionId": "c1e42054-c08f-4cb0-b893-1ebfdbad5719",
  "timeStamp": "2022-03-01 08:00",
  "sellerId": 1,
  "settlementDate": "2022-03-01",
  "amount": 2000
},
{
  "transactionId": "c1e42054-c08f-4cb0-b893-1ebfdbad5719",
  "timeStamp": "2022-03-01 08:00",
  "sellerId": 1,
  "settlementDate": "2022-03-01",
  "amount": 3000
}
```

Deve agrupar o total assim:

```
{
  "sellerId": 1,
  "amount": 5000
}
```

Grave no mongo isso (coleção sellerSettlements), assim:

```
{
  "sellerId": 1,
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "amount": 5000,
  "taxValue": 0,
  "bankAccount": null
}
```

## Solicitar o imposto

Para cada consolidado de seller, enviar uma mensagem para o serviço **tax\_calculator** utilizando a fila **tax\_calculation\_request**, assim:

```
{
  "sellerId": "1",
  "amount": 5000
}
```

### 3- Recebe o imposto

#### Receber a resposta do imposto

Note que esse fluxo está desconectado do primeiro, o primeiro fluxo começou com a requisição da API (igual a API seller\_information), esse aqui começou com a leitura da fila (igual ao tax\_calculator).

A aplicação deve ficar escutando a fila de resposta do serviço **tax\_calculator** (tax\_calculation\_response), ele vai receber mensagens assim:

```
{
  "sellerId": "1",
  "amount": 5000,
  "taxValue": 300
}
```

#### Obter o número da conta

Fazer uma chamada na API seller\_information para obter o número da conta para o seller.

A chamada será algo assim:

GET /v1/sellers/1

E irá retornar:

```
{
  "sellerId": 1,
  "name": "McDonalds",
  "cnpj": "90891366000190",
  "bankCode": 033,
  "bankAccount": 23456,
  "notes": ""
}
```

#### Atualizar imposto e a conta

Atualizar o valor do imposto na coleção sellerSettlements, então ela era assim:

```
{
  "sellerId": 1,
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "amount": 5000,
  "taxValue": 0,
  "bankAccount": null
}
```

E ficará assim:

```
{
  "sellerId": 1,
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "amount": 5000,
```

```
"taxValue": 300,  
"bankCode": 033,  
"bankAccount": 23456  
}
```

Nessa etapa aqui, verificar se já obteve o imposto para todos os sellers. Consultar a coleção `sellerSettlements` e verificar se tem algum registro com `tax_value: 0`, se você não encontrar nenhum registro sem imposto terminou a consulta de imposto, então siga para o próximo passo.

## Evento gerar arquivo de liquidação

Produza um evento em uma nova fila **settlement\_file\_request** com uma mensagem assim:

```
{  
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",  
  "settlementDate": "2022-03-01"  
}
```



## 4- Montar o arquivo de liquidação

### Escutar a fila

Escutar a solicitação para gerar o arquivo de liquidação

```
{
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "settlementDate": "2022-03-01"
}
```

### Escrever o arquivo

E criar um arquivo .txt com a data de início da liquidação, ex: **"2022-03-01\_08-00-00.txt"** nesse arquivo, gravar uma linha para cada registro da coleção sellerSettlements. Por exemplo, se tiver esse registro:

```
{
  "sellerId": 1,
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "amount": 5000,
  "taxValue": 300,
  "bankCode": 033,
  "bankAccount": 23456
}
```

Escreva uma linha com como abaixo, separando por TAB os dados:

- sellerId
- amount
- taxValue
- bankCode
- bankAccount

1	5000	300	033	23456
---	------	-----	-----	-------

### Registrar o resultado

Escreva na coleção settlements a data fim e quantos registros foram processados, assim:

```
{
  "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
  "settlementDate": "2022-03-01",
  "startDate": "2022-03-24 08:00",
  "endDate": "2022-03-24 09:10",
  "sellersCount": 10,
  "transactionsCount": 100,
  "elapsedMilliseconds": 19292
}
```

## 5- API de consulta do resultado

Criar um endpoint adicional, assim:

**GET /v1/settlements**

Que deve retornar a lista desses registros de liquidação (coleção settlements), assim:

```
[
  {
    "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
    "settlementDate": "2022-03-01",
    "startDate": "2022-03-24 08:00",
    "endDate": "2022-03-24 09:10",
    "sellersCount": 10,
    "transactionsCount": 100,
    "elapsedMilliseconds": 19292
  },
  {
    "settlementId": "4ee4cd0d-73e6-4d41-9d34-3a2618b22092",
    "settlementDate": "2022-03-01",
    "startDate": "2022-03-24 10:00",
    "endDate": "2022-03-24 10:30",
    "sellersCount": 10,
    "transactionsCount": 100,
    "elapsedMilliseconds": 19292
  }
]
```

E é só isso! 😊