

# Teste de Lógica e Programação – IB Front

## Introdução

A ideia de pedirmos para você realizar esse teste é para conhecer melhor o seu trabalho e também mostrar um pouco do que fazemos no dia a dia. Por isso ele envolverá diversos pontos que são relevantes e importantes para a vaga!

## Instruções

- Todos os detalhes estão dentro da sessão “Caso”. Se surgir alguma dúvida em relação aos detalhes, desenvolva da forma que achar melhor e deixe um comentário no código com o seu entendimento;
- Você pode escolher se prefere implementar a solução em C++ 11/14/17 ou em Java 8;
- Não utilize o nome de nenhuma empresa nesse teste! Não use o nome da sua empresa atual nem o nome da nossa empresa;
- Apesar de não definirmos quais recursos de programação você deve usar, lembre-se que a ideia do teste é entendermos o seu nível de conhecimento e capacidade de desenvolvimento;
- Sinta-se à vontade para usar libs e dependências que ajudem a construção do projeto (uma das mais utilizadas para o tratamento de FIX é a lib QuickFix para C++: <http://www.quickfixengine.org/> e QuickFIX/J para Java: <https://www.quickfixj.org/>);
- O projeto deve ser desenvolvido sempre pensando em uma alta performance com o menor tempo de processamento possível;
- Leia o documento inteiro antes de começar a desenvolver! Há algumas informações relevantes no fim do documento;
- Não é um pré-requisito, mas sinta-se a vontade para utilizar threads onde você achar importante!

## Como nos enviar o projeto

- Nossa expectativa é receber o código pronto e conseguir compilar e executar usando apenas linha de comando;
- Preencher as instruções de como rodar no README – é importante que o arquivo tenha as instruções necessárias;
- Entregar o projeto em um repositório público. Recomendamos o GitHub ou o BitBucket – **lembrando novamente o pedido para não usar nome de empresas no repositório ou código.**

## Caso

### Introdução

Nosso caso está ligado a mensagens FIX ([FIX Protocol](#)), isso significa que o teste exigirá criação e manipulação de massa de dados em FIX. Ainda sobre FIX, [iremos usar o padrão da B3](#) de estrutura de mensagens.

### Descrição do caso

O caso será dividido em três tarefas; (1) **criação de uma massa de dados** de negociação, (2) **modificar mensagens** FIX e (3) **calcular e comparar médio das operações** respeitando alguns parâmetros.

## Parte 1

A sua primeira tarefa será gerar uma massa de dados pensando no mercado de AÇÕES (Bovespa) no formato FIX. Na sessão “Anexo” você encontrará um modelo de Execution Report. Iremos gerar apenas Execution Reports com execuções parciais e totais, ou seja, **teremos apenas execuções neste arquivo**.

Utilizando-o como base, você deve gerar 5000 mensagens FIX, sendo que pelo menos 2500 devem ser **execuções totais** (atenção para os campos LastQty, LastPx, CumQty, AvgPx, ExecType, OrdStatus).

Essas novas mensagens devem possuir os seguintes campos com valores aleatórios: **conta, quantidade, preço, lado, instrumento, horário, order id, client order id e entering trader (para identificar qual tag refere ao valor, consultar o manual da B3)**. Lembrando que os valores aleatórios devem respeitar os domínios e a especificação do *Entry Point* da B3.

- Os campos **order id** e **client order id** não podem se repetir;
- O campo **conta** deve estar limitado a, no máximo, 10 valores diferentes;
- O campo **instrumento** deve estar limitado a, no máximo, 10 valores diferentes.

Por favor manter o código fonte gerador de dados aleatórios no repositório. A massa de dados pode manter estática uma vez gerada. Com a massa de dados FIX pronta em um arquivo de texto, você finalmente pode dar segmento ao teste.

## Parte 2

Será necessário consumir e tratar essa massa de dados FIX. O maior objetivo aqui é enriquecer esta massa de dados de acordo com algumas regras. No final, teremos como resultado 2 arquivos: **AllMsgs.csv** e **FullFill.txt**. Eles serão usados na Parte 3.

A sua aplicação *handler* deve ler a massa de dados que está armazenada em um arquivo de texto gerado no passo anterior (Parte 1) e considerar as seguintes condições:

1. **AllMsgs.csv: Será uma reprodução da massa FIX em formato csv. Para todas as mensagens (considerando as totais e as parciais), você deve:**
  - 1.1. Calcular o **notional da ordem** (em Bovespa, preço \* quantidade)
  - 1.2. Calcular o **notional da execução atual** (considerar a quantidade desta execução, ou seja, o valor da tag 32 LastQty como quantidade e da tag 6 AvgPx como preço)
  - 1.3. Calcular o **notional da execução acumulada** (considerar a quantidade acumulada, ou seja, o valor da tag 12 CumQty como quantidade e da tag 6 AvgPx como preço)
  - 1.4. Para cada mensagem, transformar a mensagem em uma String delimitada por ponto e vírgula considerando apenas os campos: **Horário, Conta, Instrumento, Lado, Qtd Ordem, Qtd Execucao Atual, Qtd Executada Acumulada, Preço Executado, Notional da ordem, Notional da execução atual, Notional da execução acumulada, Entering Trader**.
  - 1.5. Escrever o resultado deste procedimento em um novo arquivo **AllMsgs.csv**
2. **FullFill.txt: se a mensagem for uma execução total (novamente atenção para OrdStatus e ExecType), a mensagem da massa de dados FIX deverá ser gravada neste arquivo, porém, modificando o conteúdo da mensagem** de acordo com as regras abaixo:
  - 2.1. Calcular o **notional da ordem** e adicionar o notional da ordem na tag 1010;
  - 2.2. Replicar o *entering trader* para a tag 1011;
  - 2.3. Escrever a mensagem FIX no arquivo **FullFill.txt**

### Parte 3

Agora que você possui dois arquivos diferentes que representam duas fontes de dados diferentes, é necessário comparar o conteúdo deles para garantir que os dados estão íntegros. Para isso, iremos ler AllMsgs.csv, FullFill.txt e criar um terceiro arquivo que representa o resultado desta comparação.

Você deve primeiramente encontrar **o preço médio das operações para a mesma conta, instrumento e lado**. Para isso basta ler o arquivo **FullFill.txt** e agregar as ordens baseando-se nos campos: **conta, instrumento e lado**.

Em seguida você deve repetir esse processo no arquivo **AllMsgs.csv** considerando **apenas as execuções totais**.

Por último você deve montar um arquivo que mostre a comparação dos preços médios do FullFill.txt com os do AllMsgs.csv (filtrado por execuções totais)

Fica a cargo de você a forma de exibição, seja em um Excel, CSV, TXT, HTML, PDF, etc.

#### Exemplo de apresentação de comparação:

TxtConta	TxtPapel	TxtLado	TxtPrecoMedio		CsvConta	CsvPapel	CsvLado	CsvPrecoMedio
123456	PETR4	C	10.000		123456	PETR4	C	10.000
234567	PETR4	C	10.200		234567	PETR4	C	10.200
345678	ITUB3	V	50.400		345678	ITUB3	V	50.400
123456	VIVT4	C	30.200		123456	VIVT4	C	30.200
234567	AZUL4	V	60.120		234567	AZUL4	V	60.120
345678	PETR4	C	100.550		345678	PETR4	C	100.550

### Lembretes

- **Preço médio = Soma dos notionals / Soma das quantidades**
- Na nossa realidade muitos parâmetros são modificados com frequência... E modificar um arquivo de configuração é mais fácil do que alterar o código fonte;
- Em um mesmo projeto esperamos receber as 3 partes do caso. Eles podem ser módulos diferentes;
- Considere o cenário apresentado apenas no mercado de ações, ou seja, o *notional* é definido por preço \* quantidade.

### Anexo

#### Mensagem para ser usada de base:

```
8=FIX.4.4^9=507^35=8^34=119848^49=SISTEMA^52=20200102-
14:32:25.120^56=DROPCOPY^128=GATEWAY^1=111111^6=0^11=009FDS8HGA^14=10300^17=2
8376823^22=8^31=10.34^32=7500^37=9823759^38=10300^39=2^40=2^44=10.34^48=20000
0581991^54=1^55=PETR4^59=0^60=20200102-
14:32:25.117^75=20200102^150=F^151=0^198=29385792^207=BVMF^336=1^581=39^625=1
7^1057=N^1180=GATEWAY^5149=MEMOVALUE^6032=62140^382=1^375=8^453=4^448=100^447
=D^452=7^448=DMA^447=D^452=36^448=XYZ^447=D^452=54^448=81274681^447=D^452=5^1
0=007^
```

Lembre-se que o delimitador dos campos FIX é o caractere especial "SOH": `\u0001`

É necessário substituir o caractere ^ pelo SOH.

Exemplo: `fixMessage.replace("^", "\u0001")`

#### Especificação do Entry Point da B3:

<http://www.b3.com.br/data/files/E1/24/3A/A1/177A8710C93D1687AC094EA8/EntryPointMessageSpecs2.18.pdf>