**MAC 438 – Programação Concorrente**
Instituto de Matemática e Estatística – Segundo Exercício Programa

# Controlador de Máquinas

Neste EP faremos um controlador de máquinas de uma fábrica. A capacidade de produção e produtos disponíveis de uma fábrica são determinados pelo seu conjunto de máquinas. As características da fábrica serão fornecidas através de um arquivo texto, no formato abaixo:

Máquina;Produto;CapacidadeDeProduçãoPorIntevalo

Vamos convencionar que todos os intervalos de tempo que utilizaremos serão dados em milisegundos.

Um exemplo de tal arquivo:

```
1;1;100
2;1;300
2;2;100
3;2;500
4;3;2000
```

Observações:

- Mais de uma máquina pode produzir o mesmo produto
- As capacidades de produção são sempre maiores que zero
- Uma mesma máquina pode produzir mais de um produto
- Uma máquina produz apenas um produto por vez
- Quando ocorre o pedido de mais de um produto do mesmo tipo, o início da produção de todas as unidades do mesmo tipo devem ocorrer simultaneamente
- A produção pode ser dividida em várias máquinas (inclusive, do mesmo tipo).

A máquina será modelada pela classe ep2.base.Maquina presente no arquivo disponível junto com o enunciado deste ep. Todas as classes dentro do pacote ep2.base.* e subpacotes NÃO podem ser alteradas.

O seu objetivo neste EP é o de implementar um WebService que recebe pedidos de produção, e os despacha para a fábrica. Para cada pedido seu EP deve gerar um número que servirá para efetuar consultas sobre o andamento da sua produção. Os pedidos de produção recebidos de forma online serão especificados através de uma lista de pares de inteiros com os produtos desejados e respectivas quantidades. Todos os pedidos devem ser atendidos e possuir um identificador único. A geração e devolução do número do pedido deve ocorrer antes da produção, de forma que o cliente possa verificar o andamento de seu pedido através de um método específico do WebService. No arquivo base, estão presentes os diretórios ep2.base.server e o diretório ep2.base.client. Os arquivos dentro destes diretórios, como já foi dito, não devem ser alterados, mas no entanto devem ser utilizados como base para implementar o seu WebService. Para isto, você deve "rechear os métodos" presentes na classe ep2.FabricaImpl com o código necessário para o seu funcionamento.

No pacote ep2.base.exemplos estão classes que demonstram como criar um servidor e um cliente de WebService de forma fácil. Não se preocupe em utilizar um servidor Web mais robusto, o servidor embutido do próprio Java deve bastar.

Seu EP deverá receber como parâmetro o nome do arquivo que contém a lista das máquinas (no formato descrito acima). A partir deste arquivo, o seu programa deve criar cada uma das maquinas. A classe que representa cada máquina da fábrica (ep2.base.Maquina) não é thread-safe, ou seja, o seu programa deve cuidar para que não hajam pedidos simultâneos para cada uma das máquinas e assim não causem problemas. Note também que uma vez que uma das máquinas tenha começado a produção de um item, ela não pode ser interrompida, ou seja, começada a produção de um item ela vai até o final. O numero de máquinas e de tarefas não possui um limite pré estabelecido, mas normalmente o número de pedidos é muito maior que o número de máquinas. Seu EP deve ser capaz de lidar com um número arbitrariamente grande de máquinas e tarefas.

O objetivo do ep é atender a todas as requisições no menor tempo possível, não necessariamente obedecendo uma política FIFO mas que evite starvation.

## Sobre a entrega

Você deve entregar um arquivo tar.gz contendo os seguintes itens:

- Um arquivo README que explica como compilar e rodar o seu programa;
- Makefile
- Arquivo(s) fonte(s)
- Um conjunto de testes (JUnit) que demonstrem o funcionamento do seu programa
- Relatório sucinto explicando a sua solução

O desempacotamento do seu arquivo tar.gz deverá produzir um diretório contendo esses itens. O diretório e o arquivo tar.gz deve ter nome da forma ep2-membros-da-equipe (exemplo: ep2-joaoAugusto-mariaAparecida). A entrega deverá ser feita no paca. Se o EP for feito em dupla, **somente** uma pessoa deverá enviar o EP.

## Critério para a nota

Serão avaliados os seguintes itens na composição da nota:

- Corretude
- Uso da concorrência
- Qualidade do código
- Qualidade dos testes

**Importante**: para cada item que não for seguido exatamente como está descrito na seção *Sobre a entrega* será descontado nota. EPs com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.

O seu programa deve iniciar um serviço web na porta 8080 (tal qual o exemplo) e deve também receber o nome do arquivo de configuração das máquinas a ser carregado durante a fase de inicialização

## Atualização do enunciado.

- As funções da classe *FabricaImpl* devem trabalham com **long**.
- Você deve criar a sua própria classe com o seu main que recebe um arquivo e levanta o webservice.
- Você não deverá alterar a implementação do webservice, definido pela classe ep2.base.server.Fabrica. Apesar de não ser alterada, é a classe que deve ser usada pela sua própria classe criada no main para levantar o servidor.
- Note que apesar de mudanças nas classes do pacote ep2.base.* estarem proibidas, você tem total liberdade para alterar a classe ep2.FabricaImpl. Atente-se para o fato que a única restrição é que esta classe continue compatível com o resto do código base, e para isto basta manter o nome da classe e a assinatura dos seus dois métodos static. Tirando isso, você pode recheá-la como bem entender.
- Para verificar o funcionamento do EP vamos levantar o serviço e fazer diversas requisições. Naturalmente para que você saiba se está tudo funcionando será necessário fazer as suas requisições e ver o comportamento do seu servidor.
- Resp: No caso de mais de um pedido, o segundo pedido não necessariamente deve esperar que o primeiro termine. Caso existam máquinas disponíveis ele não só pode como deve começar antes dos outros pedidos, que eventualmente estiverem em produção, terem terminado.

## Exemplos

Vou tentar esclarecer as dúvidas através de alguns exemplos. Se faltar alguma coisa, ou algo não ficar claro, mandem mais dúvidas.

Imaginem o seguinte cenário, com uma fábrica definida pelo seguinte arquivo:

1;1;1000

Uma máquina, que produz o produto 1, 1000 unidades por intervalo de tempo. Para os pedidos dos tamanhos abaixo, ela levaria:

$30 -> 1$ unidade de tempo
$500 -> 1$ unidade de tempo
$999 -> 1$ unidade de tempo
$1000 -> 1$ unidade de tempo
$1001 -> 2$ unidades de tempo
$1999 -> 2$ unidades de tempo
$2000 -> 2$ unidades de tempo
$2001 -> 3$ unidades de tempo

Ou seja, como alguém já citou na lista, a máquina leva teto(quantidadePedida/capacidade) unidades de tempo para atender um pedido.

Pense nisso como uma máquina que é capaz de produzir uma caixa cheia de produtos a cada unidade de tempo. Se a caixa está lotada de produtos ou se tem apenas um produto o tempo total para que ela produza esta caixa é o mesmo. Logo, para o cenário onde:

1;1;300 2;1;200

E se fosse feito um pedido de 500 unidades do produto 1, então o tempo caso a máquina 1 seja escolhida seria 2 unidades de tempo, caso a máquina 2 seja escolhida seria 3 unidades de tempo.

Se a sua capacidade total da sua fábrica para a produção de um produto do tipo 1 for 2000 por unidade de tempo, tal qual especificado por um arquivo de entrada assim:

1;1;1000 2;1;500 3;1;500

O esperado é que o seu ep seja capaz de atender um pedido de tamanho 2500 em 2 unidades de tempo. Perceba que isso descarta a possibilidade de deixar toda a produção deste pedido a cargo da máquina 1, que neste caso consumiria 3 unidades de tempo. Às vezes não vai ser tão fácil chegar a este ótimo, mas a idéia é tentar chegar o mais próximo possível.

Já a idéia contida na frase do enunciado: Quando ocorre o pedido de mais de um produto do mesmo tipo, o início da produção de todas as unidades deve ocorrer simultaneamente"é a de que você NÃO pode começar um pedido a não ser que TODOS os itens tenham máquinas disponíveis para serem produzidos imediatamente. Em outras palavras, você não pode produzir um produto 1, enquanto espera uma máquina ficar livre para produzir um produto 2 do mesmo pedido. Ou todos os itens de um pedido começam, ou nenhum começa.

**Ajudando a esclarecer:**

Máquinas podem receber mais trabalho do que sua capacidade. Como temos que iniciar todo o pedido simultaneamente, então o simultâneo não se aplica ao início de todos os produtos mas sim ao início do trabalho de todas as máquinas que estão designadas para atender esse pedido.

Ou seja, devo alocar um número de máquinas que seja suficiente para produzir todos os ítens do pedido (podendo separar um ítem em várias máquinas) e iniciá-las ao mesmo tempo.

**Exemplo 2**

Voltando a um exemplo já dado, cuja entrada é:
1;1;30
2;1;30

Um pedido de 70 produtos do tipo 1 PODE ser atendido e levaria 2 unidades de tempo?
Isso mesmo. 30 unidades em cada máquina na primeira unidade de tempo, e mais 10 e uma das máquinas na segunda unidade de tempo.

**Exemplo 3**

1;1;30 1;2;30

Um pedido de: (10 produtos do tipo 1) e (10 produtos do tipo 2) NÃO PODE ser atendido, uma vez que a produção não poderia começar simultaneamente? Esta é uma variação do caso que o Alfredo havia respondido anteriormente. Vocês devem assumir que não serão feitas requisições ao web service de pedidos que não possam ser cumpridos. Isso foi dito indiretamente pela frase "Todos os pedidos devem ser atendidos e possuir um identificador único" do enunciado. Assim, o exemplo acima deve poder ser atendido pois a máquina 1 é capaz de produzir os itens necessários.

Como o Alfredo disse, vocês podem assumir que a produção de itens em uma mesma máquina começa simultaneamente. O tempo mínimo seria alcançado se fosse possível dividir a produção em várias máquinas distintas de modo a diminuir o makespan, mas isso nem sempre será possível ou fácil.

**Exemplo 4**

A entrada a seguir é inválida.
1;1;1000
2;1;500
3;1;500