

**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;CapacidadeDeProduçãoPorIntervalo

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

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 deve ocorrer simultaneamente

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 “recheiar 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 máquinas. 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 número 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