

1 Descrição

A aplicação *Bolsa Geométrica* disponibiliza os serviços de separação e entrega de formas geométricas, sob demanda, a clientes. O servidor modela um depósito automático de peças (pc_{Tipo}), representados por formas geométricas ($pc_{Tipo} = tipo1, tipo2, tipo3, etc$). O depósito possui E esteiras, p_i prateleiras e uma única plataforma de entrega. As esteiras são capazes de recuperar as peças nas prateleiras (p) vizinhas, representadas por $(p_i)E(p_j)$, e as deslocarem a plataforma de entrega. Uma prateleira pode ser servida por duas esteiras distintas. As prateleiras não são setorizadas, portanto uma peça A pode estar na mesma prateleira de uma peça B. Os clientes, por sua vez, são especializados, o que significa que possuem interesse em somente um tipo de peça (p_{cli}). Cada cliente solicita uma quantidade fixa de peças (Qtd_{cli}) de um tipo determinado. O número de clientes é determinado no lançamento da aplicação e representado por N_{cli} . O cliente finalizará suas atividades assim que recuperar todas as peças desejadas. O servidor, por sua vez, encerra suas atividades assim que todos os clientes forem atendidos completamente.

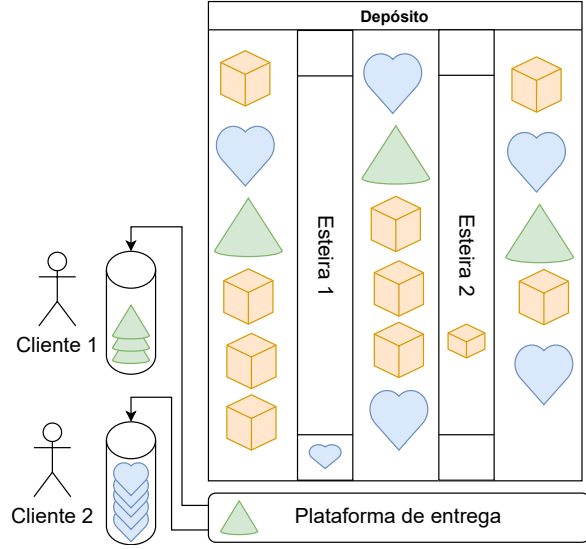


Figura 1: Funcionamento da *Bolsa Geométrica*.

2 Arquivo de Entrada

A única forma de entrada de dados é por meio do arquivo de entrada, respeitando o formato pré-definido. Parte da informação do arquivo de entrada será destinada ao servidor e o restante ao cliente. As peças são lidas sequencialmente e distribuídas igualmente nas prateleiras. No arquivo de entrada do exemplo, temos: 13 peças, de 5 tipos distintos, e 4 prateleiras, perfazendo a seguinte distribuição: $p1 = \{B E A J\}$, $p2 = \{H B B\}$, $p3 = \{J H H\}$ e $p4 = \{E A H\}$. O exemplo ainda informa a existência de 3 esteiras dispostas entre as prateleiras, uma esteira a cada duas prateleiras. Observa-se que a primeira esteira ($p1Ep2$) se localiza entre as prateleiras ($p1$ e $p2$), a segunda, ($p2Ep3$), serve as prateleiras ($p2$ e $p3$), ficando a cargo da terceira, ($p3Ep4$), as prateleiras ($p3$ e $p4$). Um exemplo de arquivo de entrada é o descrito na sequência.

```
Nro_clientes: 2
Processos: Client Client
Maquinas: ens2 ens3

## Servidor ##
NClientes = 2
NPecas = 13
pecas = B E A J H B B J H H E A H
NEsteira = 3
esteiras = p1Ep2 p2Ep3 p3Ep4
NPrateleira = 4
prateleiras = p1 p2 p3 p4

## Clientes ##
pCli1 = K
QtdCli1 = 1
pCli2 = J
QtdCli2 = 2
```

Figura 2: Exemplo de arquivo de entrada

3 Arquivo de Saída

O resultado final da aplicação *Bolsa Geométrica* é determinista. A análise do algoritmo se dará pela verificação do arquivo de saída (Figura 3).

O método de comparação é o textual caractere a caractere, portanto é essencial que o formato especificado seja respeitado.

Os clientes (N_{cli}) devem ser lançados sequencialmente para garantir a ordem correta do texto de saída. No caso do exemplo, as quatro primeiras linhas são geradas pelo *Cliente 1* e devem constar na saída padrão (stdout) do console do servidor. Em seguida, pode-se observar as quatro linhas da saída do *Cliente 2*. Em ambos os casos, aparece o estado final do atendimento (*atendido*) e o tipo de peça entregue. Caso houvesse mais clientes, as demais linhas representariam os dados desses clientes.

As últimas cinco linhas descrevem o estado final do servidor. O status *finalizado* só é atingido após a finalização do atendimento de todos os clientes. O estoque, peças que não foram entregues, deve dispor as peças por tipo e em ordem alfabética. Por fim, as entregas devem ser listadas por ordem alfabética também.

```
## Cliente (ens2) 1 ##
Status: nao atendido
pCli: K
#####
## Cliente (ens3) 2 ##
Status: atendido por ens1
pCli: J
#####
## Servidor ##
Status: finalizado
estoque: A A B B E H H H H
entregas: J J
rejeitadas: K
```

Figura 3: Exemplo de arquivo de saída.

4 Tecnologia

O projeto da *Bolsa Geométrica* deve ser desenvolvido na linguagem de programação Java e se apoiar no *middleware* de comunicação *Remote Method Invocation* (RMI). No Git da disciplina é possível baixar um código exemplo de RMI (<https://github.com/mauriciopillon/sdi0001/tree/master/03-HelloRMI>). As ferramentas de desenvolvimento estão instaladas na Plataforma de Desenvolvimento da disciplina (*Laboratório de Processamento Paralelo e Distribuído* (LabP2D)), e a aplicação deve, obrigatoriamente, ser validada na Plataforma de Submissão (BOCA) - (<http://boca-sdi.ddns.net/boca/src/>). O executável do servidor, **java -Djava.rmi.server.hostname=10.20.221.230 BGServer**, deverá ser lançado na máquina *ens1*, via *script*, e os N_{cli} (**java Client ens1 1, java Client ens2 2, ...**) na máquina *ens2* ... *ens4*. Apoiem-se nos *scripts* do problema teste, já disponível no BOCA, adequando o que for necessário.