

Considere um estacionamento com N vagas, e um conjunto de 10 estações entradas/saídas. Seu objetivo é construir um sistema distribuído (**middleware + app**) para controlar as N vagas, ou seja, quando um veículo requisitar entrada, sua aplicação deverá:

- Localizar uma vaga livre;
- Existindo vaga livre, alocar essa vaga ao veículo, marcando a vaga como ocupada (não livre);

Características do estacionamento:

- a) Nas estações de entrada os veículos estão em fila por ordem de chegada para atendimento;
- b) Ao entrar no estacionamento, o veículo recebe uma lista de estações ativas;
- c) Os veículos permanecem na vaga por um tempo aleatório, ao final selecionam uma estação de saída (da lista recebida) para indicar a liberação da vaga e registro de saída. A vaga deve retornar ao conjunto de vagas livres gerenciado pela estação.
- d) As estações de entrada/saída podem ser ativadas e desativadas, inicialmente todas estão desativadas, ao iniciar (ativar) uma estação:
 - O middleware realiza contato inicial com um “gerente” em endereço/porta conhecidos (e fixos);
 - O middleware da nova estação deve estabelecer uma conexão com uma das estações, formando uma “árvore de encaminhamento”;
 - A comunicação entre middlewares deve ser realizada através dessa árvore de encaminhamento;

As vagas do estacionamento são numeradas de 1 a N.

O objetivo do estacionamento é manter a maior quantidade de vagas ocupadas e atender os clientes (definir uma vaga no estacionamento) no menor tempo possível.

As vagas do estacionamento devem ser controladas pelas estações ativas, **não há uma base de dados única em nosso sistema**. As vagas devem ser divididas entre as estações ativas.

Em caso de falha de uma estação, as vagas controladas pela estação devem ser repassadas a outra estação, através de uma eleição.

Caso não exista vaga disponível em uma estação, pode ser utilizada vaga disponível de outra estação.

A liberação da vaga pode ocorrer em qualquer estação disponível, sua implementação deve retornar a vaga disponível para a estação original. Existe a possibilidade de falha na estação original da vaga, nesse caso, defina uma solução para a vaga livre.

Deve ser fornecido um arquivo texto contendo a **lista das 10 estações** no seguinte formato: nome#IP#porta. Onde:

- nome: representa o nome da estação com no máximo 30 caracteres;
- IP: endereço IP do App;
- porta: número da porta para recebimento de comandos (no App).

Serão 10 estações, cada estação em uma linha do arquivo.

As mensagens apresentam o seguinte formato básico: (“código”, lista de parâmetros)
Mensagens enviados pelo Carro ao App:

- Requisitando uma vaga: simulando o carro chegando para entrada no estacionamento, código “RV”
 - Parâmetros enviados para ação “requisitar vaga”:
 - código: RV
 - Identificação do carro: ID do carro (ID gerado thread carro)
- Liberando a vaga: simulando um carro saindo do estacionamento, código “LV”
 - Parâmetros enviados para ação “liberar vaga”:
 - código: LV
 - Identificação do carro: ID do carro (ID gerado quando o carro chegou na estação)

Mensagens enviadas do App ao Carro:

- Envia um “OK” quando existir vaga disponível (a thread carro está programada para aguardar o “ok”).
- Não envia mensagem se não houver vaga disponível.

A fila de atendimento no App do estacionamento será por ordem de chegada de carros.

Outro elemento do sistema é a **central de controle** da simulação, esse módulo será responsável por gerar threads que simulam os carros e repassar comandos de controle para as estações, a seguir algumas mensagens geradas pela central de controle:

- Da central de controle para o App:
 - **Ativar Estação**: o software da estação já estará ativo, no entanto, ainda não responde a nenhuma requisição, ao receber esse comando o App solicita ao middleware para iniciar seu trabalho, ou seja, inicia interação com as demais estações (ações do passo “d”), código: “AE”:
 - Parâmetros enviados para ação “ativar estação”:
 - código: AE
 - número da estação: 1 a 10 (somente um número)

Exemplo: considere a seguinte lista de estações:

```
Station1 127.0.0.1 8881
Station2 127.0.0.1 8882
Station3 127.0.0.1 8883
Station4 127.0.0.1 8884
Station5 127.0.0.1 8885
Station6 127.0.0.1 8886
Station7 127.0.0.1 8887
Station8 127.0.0.1 8888
Station9 127.0.0.1 8889
Station10 127.0.0.1 8890
```

O comando “AE 2” envia mensagem de ativação para a Station2 (em seu endereço IP:porta).
 - **Falha na estação**: simular uma falha, após receber essa mensagem a estação não deve processar nenhuma outra requisição, ou seja, o App envia msg ao middleware para parar toda comunicação (lembre-se isso vale para middleware + app), o middleware simplesmente não responde a nenhuma requisição, o mesmo vale para o App. A única exceção é a mensagem de **ativar estação**. Código dessa mensagem “FE”
 - Parâmetros enviados para ação “falha na estação”:
 - código: FE
 - número da estação ativa: 1 a 10 (somente um número)

- **Vagas disponíveis na estação:** ao receber essa mensagem a estação deve recuperar a o total de vagas disponíveis, as vagas ocupadas e livres em cada estação ativa (on-line) no momento. Código da mensagem “VD”. A estação que recebe essa mensagem responde com uma mensagem com código “AV”, seguido por uma lista de tuplas contendo: (nome da estação, total de vagas disponíveis, vagas ocupadas e vagas livres).
 - Parâmetros enviados para ação “vagas disponíveis na estação”:
 - código: VD
 - número da estação ativa: 1 a 10 (somente um número)
 - Exemplo: considere a seguinte lista de estações, as estações ATIVAS estão marcadas com *:
Station1 127.0.0.1 8881*
Station2 127.0.0.1 8882*
Station3 127.0.0.1 8883
Station4 127.0.0.1 8884
Station5 127.0.0.1 8885
Station6 127.0.0.1 8886*
Station7 127.0.0.1 8887
Station8 127.0.0.1 8888
Station9 127.0.0.1 8889
Station10 127.0.0.1 8890
 - O comando “VD 2” será processado pela central de controle, gerando uma mensagem com o código “VD” para a Station2 (127.0.0.1:8882). O middleware da estação 2 deve responder com a seguinte conteúdo:
Station1:2-3, Station2:4-5, Station6:7-8.
 - Ou seja, a Station1 possui 2 vagas livres e 3 ocupadas.

****** Lembre-se o APP não envia mensagens diretamente a outro APP, e sim, solicita um serviço do middleware.

****** O **gerente** apresenta tolerância a falhas (ou seja, não apresenta falha).

****** podem surgir novas mensagens no decorrer do desenvolvimento do projeto.

Tolerância a falhas: em caso de falha de uma estação, outra deve assumir o controle das vagas;
{detectar falha, eleição}

Sempre que um middleware detectar uma falha, deve iniciar uma eleição para determinar a estação que será responsável pelas vagas da estação com falha.

Sempre que uma nova estação é ativada, receberá uma quantidade de vagas para gerenciar (essas vagas são fruto da redistribuição do total de vagas existentes).