

**Atenção:** Como a AD é individual, caso seja constatado que provas de alunos distintos são cópias uma das outras, independentemente de qualquer motivo, a todas será atribuída nota ZERO. As soluções para as questões podem sim, ser buscadas por grupos de alunos, mas a redação final das respostas para as questões da prova tem que ser individual!

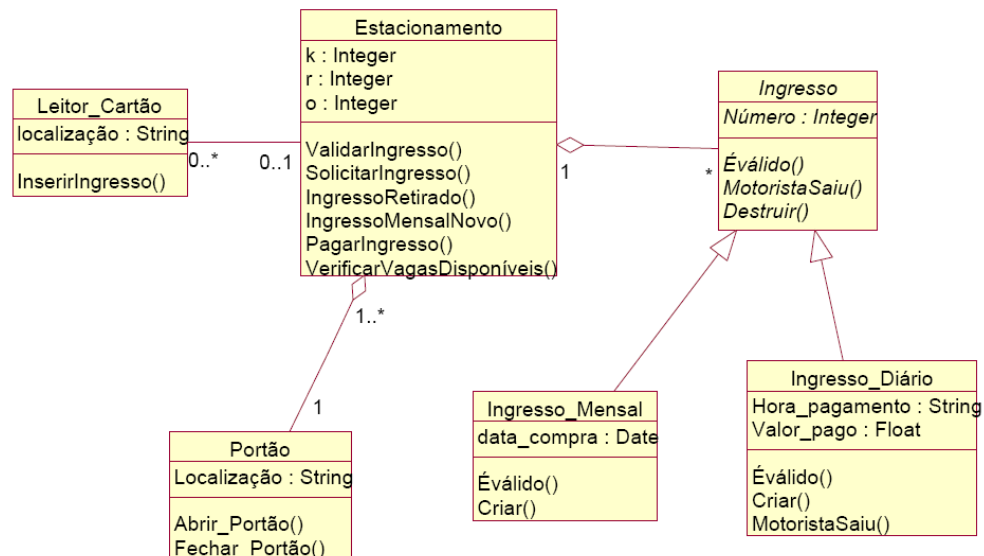
**ADS enviadas pelo correio, devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.**

1. Utilizando a notação UML e considerando o documento de requisitos para o Sistema de Controle de Estacionamento (em anexo), apresente (7,0):

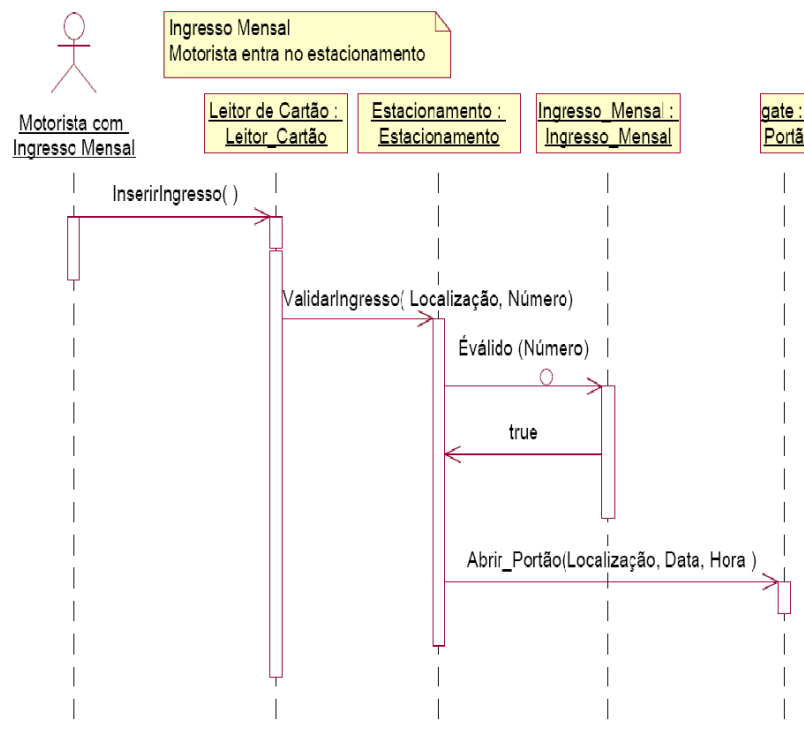
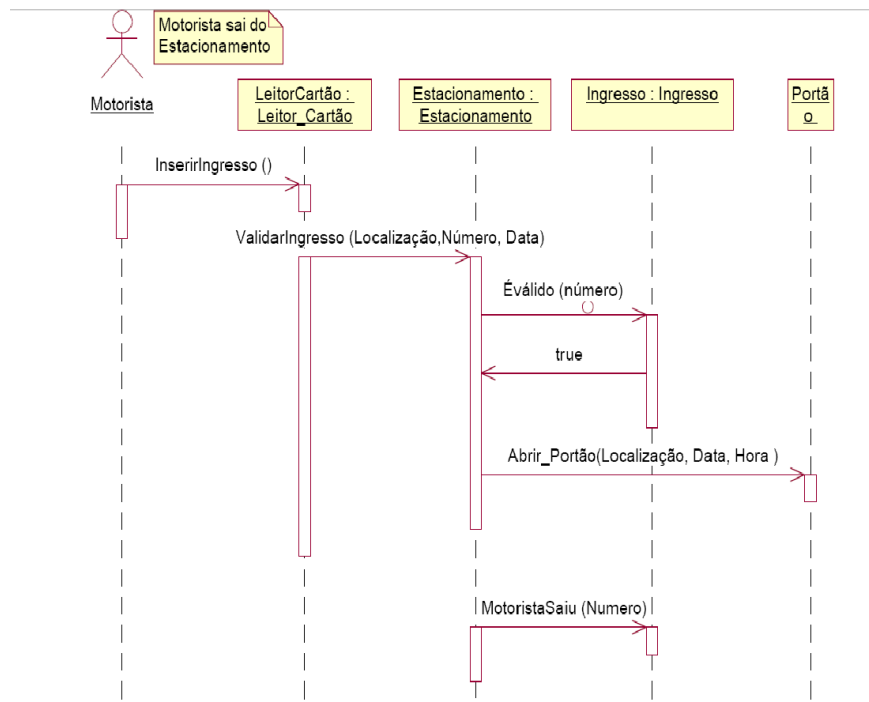
O gabarito para esta questão é dependente do modelo produzido pelos alunos. Portanto, as respostas abaixo representam apenas uma das possíveis soluções. O mais importante deste tipo de questão é permitir ao aluno aplicar os conceitos de modelagem ao mesmo tempo que pode discutir com o tutor sobre sua solução

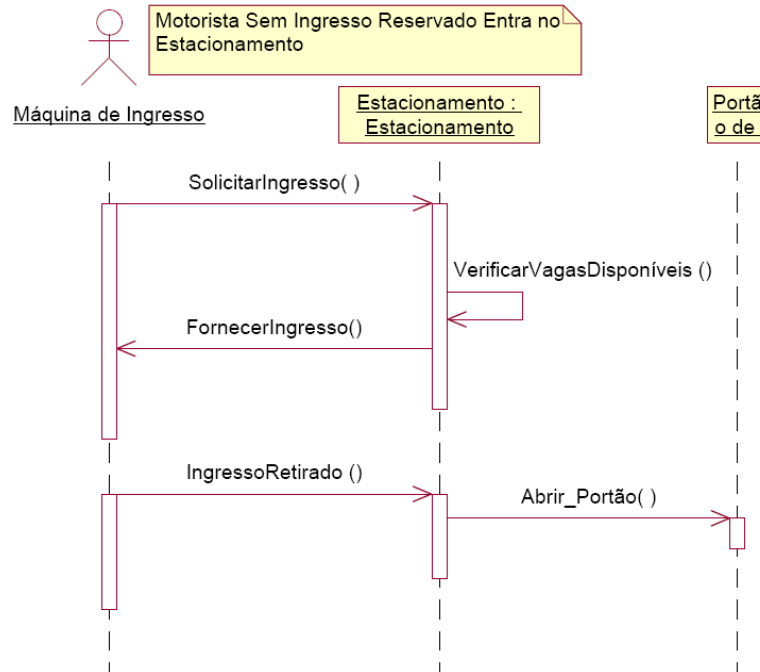
- a. o diagrama de classes para o sistema (2,0 pontos)

**Possível solução:**



- b. os diagramas de seqüência para 3 casos de uso (1,5 pontos) (**POSSIVEL SOLUCAO**)





- c. Indique que técnicas de inspeção podem ser usadas para revisar estes diagramas, descrevendo os critérios que você usou para escolher e os tipos de defeito que podem ser encontrados. De um exemplo aplicando a técnica nos diagramas que você criou (1,0 ponto)

Neste caso o projetista poderia utilizar técnicas ad-hoc ou então técnicas de leitura orientadas a objetos, OORTs, que se aplicam a inspeção de diagramas orientados a objetos descritos em UML. Os defeitos que poderiam ser considerados são omissão, fato incorreto, inconsistência, Informação estranha e ambigüidade. Por exemplo, observando os diagramas apresentados neste gabarito e aplicando OORTs (leitura horizontal) seria possível revisar o diagrama de seqüência 3 com o diagrama de classes, podemos observar um possível defeito de fato incorreto (maquina de ingresso x leitor de cartão)

- d. Apresente o conjunto de casos de testes para realizar o teste de Manutenção do Sistema, ou seja, entrada e saída de veículos, indicando qual abordagem foi utilizada para gerar esta informação (1,5 pontos)

#### Manutenção e teste do sistema

Para testar e manter o sistema é possível entrar o número total de vagas e o número de vagas reservadas com a ajuda da unidade de controle.

Neste caso devemos usar as informações existentes na exigência funcional 1 e nos requisitos funcionais 2-5:

**k:** número máximo de vagas disponíveis no estacionamento: 1000 (default)

**r:** número de vagas reservadas no estacionamento;

**a = k – r:** número de vagas não reservadas e disponíveis;

**o:** número de vagas não reservadas e ocupadas.

Daí se pode aplicar alguma abordagem para derivar os casos de testes. Análise de Valor limite pode ser uma abordagem interessante neste caso. Particionamento por equivalência também poderia ser usado.

Assim, teríamos, usando particionamento por equivalência:

Válidos:  $0 < k \leq 1000$ ,  $r \leq k$ . Resultado:  $a \geq 0$  ( $a = k - r$ )

Inválidos:  $0 < k \leq 1000$ ,  $r > k$

Inválidos:  $k \leq 0$ ,  $K > 1000$ , qualquer  $r$

- e. Utilize as métricas NOC, DIT e CBO para identificar as classes que poderiam ser mais propensas à falha no seu projeto. Indique as classes e o motivo desta indicação (1,0 ponto)

	Leitor Cartão	Estacionamento	Ingresso	Portão	Ingresso Mensal	Ingresso Diário
NOC	0	0	2	0	0	0
DIT	0	0	0	0	1	1
CBO	1	3	1	1	1	1

Neste caso é possível que a classe Estacionamento seja mais propensa à falha devido a ser a classe mais acoplada do projeto, fazendo com que ela dependa de 3 outras classes.

2. Quais são as três características fundamentais de um bom programa? Apresente e justifique a relação (ou ordem) de importância entre estas características. (valor: 1,0 ponto; máximo: 5 linhas).

Um bom programa deve ser correto, fazendo o que lhe foi requisitado e não fazendo o que não lhe foi requisitado; ser legível e fácil de manter; e ser racional no uso dos recursos computacionais, como memória e disco. A primeira característica é a mais importante, posto que um programa incorreto não tem valor, ainda que seja legível e consuma poucos recursos computacionais. A segunda característica é importante porque um programa usado geralmente sofre alterações constantes – desta forma, ele deve estar preparado para ser facilmente alterado.

3. Relacione o gerenciamento de tempo com o gerenciamento de escopo em projetos de desenvolvimento de software. Como as atividades de cada processo de gerenciamento geram insumos que são utilizados por outras atividades dos mesmos processos? Que insumos são estes e como são utilizados? (Valor: 2,0 pontos; máximo: 10 linhas)

No gerenciamento do escopo de um projeto de desenvolvimento de software identificamos o trabalho que deve ser feito no contexto do projeto. Este trabalho é registrado na forma de uma estrutura analítica de projeto, que representa os pacotes de trabalho que devem ser entregues. No gerenciamento de tempo, estes pacotes são refinados em atividades, que são organizadas em uma rede de atividades de acordo com suas dependências. A partir da rede de atividades podemos gerar um cronograma para o projeto.

ESE/COPPE/UFRJ