UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
Bacharelado em Ciência da Computação / Engenharia da Computação
INF 01127 – Engenharia de Software N
Profa. Karin Becker

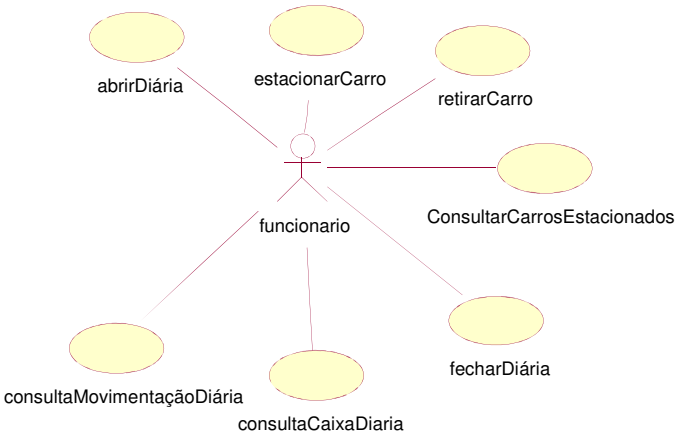
Exercício – Modelo de projeto

Objetivos: para o modelo de casos de uso/análise descritos abaixo

- Elaborar o diagrama de classes de projeto;
- Mostrar a realização dos casos de uso através de diagramas de interação

Enunciado:
Um estacionamento tem como negócio a permanência de carros por durações curtas de tempo (horas)[1]. Ele deseja poder controlar seu negócio por dia, em um conceito que denomina DIARIA. Quando inicia um dia de trabalho, um funcionário cria um novo registro de diária. A partir deste momento, deseja-se poder controlar os carros que estão no estacionamento, em função do tempo de permanência, controlando o caixa e o número de carros estacionados. Ao final do expediente, a diária é fechada. O dono do negócio deseja poder fazer consultas sobre as diárias abertas e fechadas. Os Casos de Uso abaixo explicam de forma mais detalhada o que o sistema faz.

Modelo de Casos de Uso



Caso de Uso: AbrirDiária
Ator: Funcionário
Pré-condição: não há diária aberta
Pós-condição: é criado um controle para o dia (Diaria) com a data atual e valor/hora cobrado, sendo que não há carros estacionados, e o caixa está vazio.
Seqüência típica de eventos:

1. O caso de uso inicia quando o funcionário solicita a criação de uma diária.
2. O sistema informa o dia atual.
3. O funcionário informa o valor/hora a ser cobrado, confirma a data (ou fornece outra data).
4. O sistema cria um registro de estacionamento para o dia (diária), com a data e valor hora, sendo que o caixa é inicializado com 0 e nenhum carro encontra-se estacionado.

Fluxo Alternativo:
4.a Já existe registro para o dia fornecido
Cancelar a operação.

Caso de Uso: EstacionarCarro
Ator: Funcionário
Pré-condição: existe uma Diária aberta

[1] Não existem assim neste estacionamento carros mensalistas, ou que pernoitam.

Pós-condição: o carro é incluído no registro da Diária, sendo registrada sua placa e hora de entrada no estacionamento.
Seqüência típica de eventos:
1. O caso de uso inicia quando o funcionário informa a placa do carro.
2. O sistema registra a placa do carro, junto com a hora atual.
Fluxo Alternativo:
2.a Um carro com a placa fornecida já está localizado no estacionamento
Cancelar a operação.

Caso de Uso: RetirarCarro
Ator: Funcionário
Pré-condição: existe uma Diária aberta
Pós-condição: é excluído o registro de estacionamento do carro, é informada a duração de sua estadia, bem como o preço a pagar. As informações da diária são atualizadas: o caixa é acrescido do valor cobrado (pago), e a informação do número de carros que já deixaram o estacionamento é atualizada.
Seqüência típica de eventos:
1. O caso de uso inicia quando o funcionário informa a placa do carro a ser retirado
2. O sistema calcula quantas horas o carro permaneceu estacionado (é cobrada hora cheia), o preço da estadia (dependente do valor cobrado na diária), atualizando as informações de movimentações de carro do dia (número de carros estacionados correntemente, e número de carros que já saíram), acrescentando o preço pago ao caixa diário.

Fluxo Alternativo:
2.a Um carro com a placa fornecida não está localizado no estacionamento
Cancelar a operação.

Caso de Uso: FecharDiária
Ator: Funcionário
Pré-condição: existe uma Diária aberta
Pós-condição: a diária é registrada como encerrada.
Seqüência típica de eventos:
1. O caso de uso inicia quando o funcionário solicita fechamento da Diária aberta.
2. O sistema verifica que todos os carros foram retirados, encerrando a diária..

Fluxo Alternativo:
2.a Estacionamento não está vazio
Cancelar a operação.

Caso de Uso: ConsultarCarrosEstacionados
Ator: Funcionário
Pré-condição: existe uma diária aberta.
Pós-condição: é informado número total de carros que se encontram dentro do estacionamento.
Seqüência típica de eventos:
1. O caso de uso inicia quando o funcionário solicita o total de carros estacionados.
2. O sistema informa o número de carros existentes dentro do estacionamento.

Caso de Uso: ConsultarCaixaDaDiaria
Ator: Funcionário
Pré-condição:. existem diárias registradas.
Pós-condição: é informado total obtido com pagamento de estacionamentos
Seqüência típica de eventos:
1. O caso de uso inicia quando o funcionário informa para qual data deseja fazer a consulta.
2. O sistema informa o valor registrado do caixa.

Fluxo Alternativo:
2.a Não existe uma diária para a data fornecida
Cancelar a operação.

Caso de Uso: ConsultarMovimentaçãoDaDiaria
Ator: Funcionário
Pré-condição:. existem diárias registradas.
Pós-condição: é informado número total de carros que utilizaram o estacionamento:
1. O caso de uso inicia quando o funcionário informa para qual data deseja fazer a consulta.

2. O sistema informa o número de carros que utilizaram o estacionamento (já pagos).

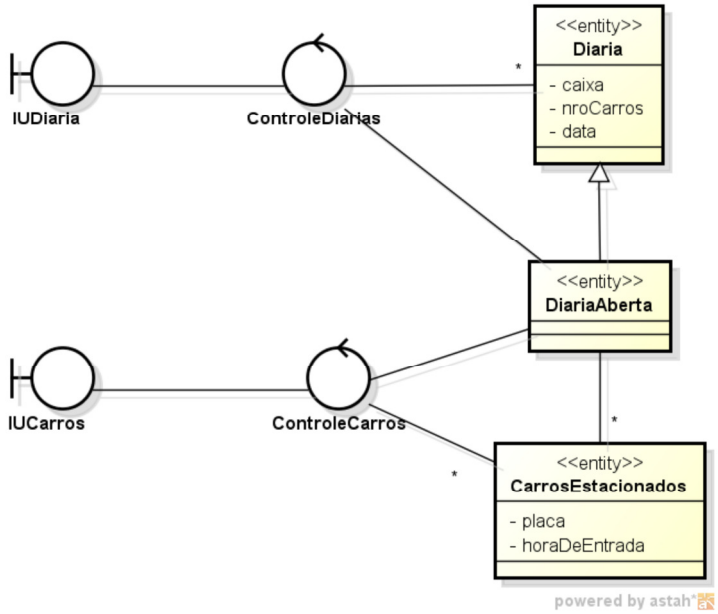
Fluxo Alternativo:

2.a Não existe uma diária para a data fornecida

Cancelar a operação.

Modelo de Análise

O modelo de análise correspondente é representado abaixo. A realização dos casos de uso sobre diárias e sobre carros foram agrupadas com um único controlador/fronteira.



Modelo de Projeto

a) Projete um conjunto de classes procurando distribuir as responsabilidades de forma homogênea. Isto envolve definir:

- as classes
- atributos (com tipo e visibilidade)
- operações das classes (com argumentos e visibilidade)
- o sentido de navegação das associações

b) Ilustre com um diagrama de interação a realização de cada caso de uso. Considerar apenas a semântica da aplicação, iniciando pelos eventos de sistema, desprezando o projeto da interface. Faça no mínimo a realização dos seguintes casos de uso:

- abrirDiária
- estacionarCarro
- retirarCarro
- fecharDiária

c) Sugestão: use a técnica de Larman (eventos de sistema, padrões GRASP). Inicie detectando os eventos de sistema (eventos que são gerados via a interface). Pense nas verificações que devem ser feitas, e nas modificações geradas pelo sistema para cada evento. Para cada evento, você deve decidir qual classe é o controlador do evento externo, o especialista da informação, e o criador dos objetos. Seu projeto deve buscar baixo acoplamento e a alta coesão.