

**Disciplinas:** Fundamentos de Engenharia de Software  
Algoritmos e Estruturas de Dados I  
Laboratório de Computação I  
**Curso:** Engenharia de Software  
**Profs.:** Maria Augusta Nelson, Roberto Felipe Rocha e Ivre Marjorie R. Machado  
**Entrega:** 26/11/2018  
**Valor:** 10 pontos (FES) – 7 pontos (AEDI) – 7 pontos (LCI)

**Observações:**

- O trabalho poderá ser feito em **grupos de até 5 alunos**.
- Copias de trabalho receberão a nota **ZERO**
- O trabalho será avaliado em **07 pontos**
- O programa deve ser feito na linguagem de programação C
- O programa deverá ser entregue pelo SGA até o dia **26/11/2018** as **23:59 horas**
- Deverá ser entregue o **projeto completo** do **programa** e a **documentação**
- Em caso de dúvida, envie email para seu professor

### Locadora LocaMais

LocaMais é uma locadora de veículos que tem como objetivo atender bem e fidelizar seus clientes. A princípio a LocaMais está trabalhando apenas com locações baseadas em diárias, ou seja, a quilometragem usada pelo cliente é livre. A locadora está localizada em diversas cidades do estado de Minas Gerais, no entanto, a filial de Confins está com sérios problemas, pois ainda não implantou um sistema para realizar seus controles e gerenciamentos de locações de veículos. Muitas vezes um cliente liga solicitando a locação de um veículo e a atendente não consegue saber se será possível atendê-lo. Sem falar que não é possível ter um controle de quantas locações foram feitas por um determinado cliente e premia-lo de alguma forma. Diante dos problemas vividos pela LocaMais, a locadora resolveu contratar uma empresa desenvolvedora de sistemas (vocês). Sendo assim, é necessário compreender a real necessidade da locadora e desenvolver um software específico. A seguir foi descrito como deverá ser o sistema, bem como suas restrições.

### O sistema

Deseja-se cadastrar os clientes, os funcionários, os veículos e as locações da locadora. As informações que devem ser cadastradas são:

CLIENTE = código, nome, endereço, telefone

FUNCIONARIO = código, nome, telefone, cargo, salario

LOCACAO = código da locacao, data de retirada, data de devolução, seguro, quantidade de dias, código do cliente, código do veiculo

VEICULO = código do veiculo, descrição, modelo, cor, placa, valor diária, quantidade de ocupantes, status

Considere as seguintes restrições: para cadastrar uma locação, primeiro é necessário que o cliente e o veículo estejam cadastrados. As locações devem ser cadastradas apenas para veículos com status disponível, sendo que os possíveis status para o veículo são: alugado, disponível e em manutenção. Além disso, não devem ser feitas mais de uma locação para um mesmo veículo em

um mesmo período (data de retirada e data de devolução). As locações podem ser feitas no horário comercial (de 8:00 as 18:00 horas).

1. Implemente uma função para cadastrar um cliente. Esta função deve garantir que não haverá mais de um cliente com o mesmo código. Se quiser pode gerar o código automaticamente.
2. Implemente uma função para cadastrar um funcionário. Esta função deve garantir que não haverá mais de um funcionário com o mesmo código. Se quiser pode gerar o código automaticamente.
3. Implemente uma função que cadastre uma locação. Para cadastrar a locação, o sistema deve receber do usuário o nome do cliente que deseja alugar o veículo, a data de retirada, a data de devolução do veículo e a quantidade de ocupantes necessários. A partir disso, o sistema deve encontrar um veículo que esteja disponível para a necessidade do cliente e calcular a quantidade de diárias com base na data de retirada e devolução. O cliente também pode contratar um seguro para o veículo alugado, caso queira, o valor do seguro é de R\$ 50 e será acrescido ao valor total pago ao final.
4. Implemente uma função que dê baixa em uma determinada locação, calcule e mostre o valor total a ser pago por um determinado cliente. Lembre-se de alterar o status do veículo para disponível. Além disso, será necessário solicitar a data de entrega do veículo para validar com a data de devolução prevista. Também será preciso calcular multa por atraso caso o cliente não entregue o veículo no dia combinado (considere como multa o valor de 5% do total + R\$ 30,00 por dia de atraso).
5. Implemente funções para realizar pesquisa tanto para clientes quanto para funcionários, ou seja, digitando o nome apresenta na tela todas as informações do cliente ou funcionário.
6. Implemente uma função que mostre na tela todas as locações de um determinado cliente (a pesquisa vai se basear no nome do cliente).
7. Implemente uma função que calcula a quantidade de pontos de fidelidade de um cliente. Para cada dia de locação, o cliente ganhará 10 pontos no programa de fidelidade. Lembre-se que é possível ter mais de uma locação para um mesmo cliente. Implemente também uma função para pesquisar os clientes que atingiram um total de 500 pontos no programa de fidelidade, esses clientes serão premiados e receberam um kit da LocaMais.
8. Implemente uma função extra, criada pelo grupo. Sejam criativos!

Para fazer este programa pode ser necessário criar mais funções do que as que estão descritas. Finalmente, faça uma função `main()` que teste o sistema acima. A função `main()` deve exibir um menu na tela, com as opções de cadastrar um cliente, um funcionário, uma locação e um veículo. Além disso, permitir realizar as pesquisas. Este menu deve ficar em loop até o usuário selecionar a opção SAIR. Além disso, todas as informações deverão ser persistidas/armazenadas em arquivo(s) texto, portanto, deverá ser feita leitura e escrita em arquivos.

## Metodologia

Este é um trabalho interdisciplinar onde você vai planejar, analisar, projetar e implementar uma

solução de software para o problema de controle do salão de festas utilizando o Scrum para gerenciar o seu progresso.

Inicialmente organize o seu *backlog* de produto contendo as funções básicas do sistema. Cada uma das funções será responsabilidade de um membro do grupo e será feita em *sprints* de 3-4 dias. Seguem algumas sugestões de atividades a serem feitas nas *sprints*:

- 1- Definir a assinatura da(s) função(ões). Reflita sobre os parâmetros de entrada e saída da função e comunique aos seus colegas de projeto.
- 2- Documente os parâmetros da sua função. O nome da função deve ser escolhido sob o ponto de vista de quem usa a função ou de quem vai chamar a função e deve refletir o que a função faz.
- 3- Implemente o caso de sucesso da função.
- 4- Selecione casos de testes para verificar o funcionamento da sua função. Um caso de teste deve conter os valores de entrada para a função, a saída esperada.
- 5- Execute os casos de testes, criando um relatório de execução de testes que contem os casos de testes, a saída retornada durante a execução e uma indicação se o teste passou ou não na função. Isso é feito comparando a saída esperada documentada no caso de testes com a saída retornada durante a execução (esperado x real).
- 6- Implemente os casos especiais, exceções que possam existir na sua função. Em seguida execute os casos de testes anteriores para garantir que as mudanças não quebraram o código anterior que já funcionava. Pense também nos novos casos de testes necessários para a nova versão da função.

### O que deve ser entregue para os professores no SGA

- 1- A evolução do *backlog* de produto a cada semana. Indique quais tarefas encontram-se no *backlog* do produto, as que foram alocadas nas *sprints* (indique em qual *sprint* cada tarefa foi alocada).
- 2- A documentação das funcionalidades do software
- 3- A documentação de casos de testes com o relatório de execução dos testes.
- 4- O código em C das funções e do programa principal, juntamente com o **projeto completo** do programa.