

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, Roque Anderson e Ivre Marjorie R. Machado
Entrega:	25/06/2021
Valor:	10 pontos (FES) – 10 (AEDI)

Observações:

- O trabalho poderá ser feito em **grupos de até 5 alunos**.
- Cópias de trabalho receberão a nota **ZERO**
- O trabalho será avaliado em **10 pontos**
- O programa deve ser feito na linguagem de programação C
- O programa deverá ser entregue pelo SGA até o dia **25/06/2021** as **23:59 horas**
- Será realizada uma apresentação online através do Microsoft Teams no dia **29/06/2021** nos horários das aulas práticas e teórica, no qual o prof. poderá fazer perguntas individuais para todos os componentes do grupo
- Deverá ser entregue o **projeto completo** do programa e a **documentação**
- Em caso de dúvida, envie email para seu professor

Hotel Descanso Garantido

Descanso Garantido é um hotel que tem como objetivo atender bem seus clientes, além disso, procura fideliza lós. Está localizado no centro de Itacaré – BA e possui alguns funcionários com seguintes cargos (recepcionista, auxiliar de limpeza, garçom, gerente). Acontece que até hoje o Hotel Descanso Garantido fazia seus controles de estadias, clientes e funcionários em planilhas do excel e cadernos, o que tem gerado diversos problemas para a organização. Sem falar que muitas vezes um mesmo quarto é reservado para mais de um cliente. Diante dos problemas vividos pela Descanso Garantido, o hotel resolveu contratar uma empresa desenvolvedora de sistemas (vocês). Sendo assim, é necessário compreender a real necessidade do hotel 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 e as estadias do hotel. As informações que devem ser cadastradas são:

- **CLIENTE** = código, nome, endereço, telefone
- **FUNCIONARIO** = código, nome, telefone, cargo, salario
- **ESTADIA** = código da estadia, data de entrada, data de saída, quantidade de diárias, código do cliente, número do quarto
- **QUARTO** = número do quarto, quantidade de hospedes, valor diária, status

Considere as seguintes **restrições**: para cadastrar uma estadia, primeiro é necessário que o cliente e o quarto estejam cadastrados. As estadias devem ser cadastradas apenas para quartos com status

desocupado, sendo que os possíveis status para o quarto são: ocupado e desocupado. Além disso, não devem ser feitas mais de uma estadia para um mesmo quarto em um mesmo período (data de entrada, data de saída). As diárias têm início às 14h00 horas e findam às 12h00 horas do dia seguinte.

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 estadia. Para cadastrar a estadia, o sistema deve receber do usuário o nome do cliente que deseja se hospedar, a quantidade de hóspedes, a data de entrada e a data de saída. A partir disso, o sistema deve encontrar um quarto que esteja disponível para quantidade hóspedes desejados. Além disso, a quantidade de diárias deverá ser calculada com base na data de entrada e saída.
4. Implemente uma função que dê baixa em uma determinada estadia e calcule e mostre o valor total a ser pago por um determinado cliente. Além disso, lembre-se de alterar o status do quarto para desocupado.
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 estadias de um determinado cliente.
7. Implemente uma função que calcula a quantidade de pontos de fidelidade de um cliente. Para cada diária no hotel, o cliente ganhará 10 pontos no programa de fidelidade. Lembre-se que é possível ter mais de uma estadia para um mesmo cliente.

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 e uma estadia. 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 armazenadas em arquivo(s) texto, portanto, deverá ser feita leitura e escrita em arquivos.

** Não será obrigatório a implementação do programa usando Orientação a Objetos, mas caso o grupo queira usar será permitido.*

Metodologia

Este é um trabalho interdisciplinar onde você vai planejar, analisar, projetar e implementar uma solução de software para o problema apresentado 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 a sua função colocando o seu propósito, os parâmetros de entrada e saída 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, da sua função. Inicie fazendo a execução manual de alguns poucos casos de teste. Em seguida implemente a automatização dos testes da sua função usando a biblioteca munit. Especialmente as funções de cálculos de valores (preço, multa, desconto, total) devem ser testadas de forma automatizada.
- 6- Crie um relatório de execução de testes que contenha 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).
- 7- 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 Canvas

- 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, a implementação dos casos de testes automatizados e o relatório de execução dos testes.
- 4- O código em C das funções e do programa principal, juntamente com o **projeto completo** do programa.

Link para a biblioteca munit: <https://nemequ.github.io/munit/>