

Disciplinas: Fundamentos de Engenharia de *Software*
Algoritmos e Estruturas de Dados I

Professores: Eveline Alonso Veloso e Roberto Felipe Rocha

Entrega: 23/06/2024

Valor: 10 pontos (FES) – 7 pontos (AEDsI)

Observações:

- O trabalho poderá ser feito em **grupos de até 3 alunos, desde que esses alunos estejam simultaneamente nas turmas das 2 disciplinas ou fazendo apenas 1 disciplina.**
- Cópias de trabalho receberão nota **ZERO**.
- O programa deve ser feito na linguagem de programação C.
- As informações manipuladas neste trabalho deverão ser armazenadas em arquivo(s) de **acesso direto**. Portanto, deverá ser feita leitura e escrita em arquivos com a utilização de **bibliotecas em C**.
- O trabalho deverá ser entregue pelo Canvas até o dia **23/06/2024 às 23:59 horas**.
- O grupo deve preparar uma **apresentação gravada** com a participação de todos os seus componentes. Essa apresentação também deverá ser entregue no Canvas em todas as disciplinas participantes do trabalho interdisciplinar.
- Deverá ser entregue o **projeto completo** do programa, a **documentação**, os **arquivos** contendo os testes realizados e a apresentação gravada em todas as disciplinas participantes do trabalho interdisciplinar.
- Em caso de dúvida, entre em contato com seu professor.

Hotel Descanso Garantido

Descanso Garantido é um hotel que tem como objetivo atender bem seus clientes e fidelizá-los. Está localizado no centro de Itacaré – BA e possui alguns funcionários com os seguintes cargos: recepcionista, auxiliar de limpeza, garçom, gerente. Acontece que até hoje o Hotel Descanso Garantido faz 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 pelo Descanso Garantido, o hotel resolveu contratar uma empresa desenvolvedora de *software* (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, salário

Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática
Curso de Engenharia de *Software*

- 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 hóspedes, valor da diária, *status*

Considere as seguintes **restrições**: *** Não se esqueça de sempre validar essas 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 deve ser feita mais de uma estadia para um mesmo quarto em um mesmo período (data de entrada e data de saída). As diárias têm início às 14h00 e findam às 12h00 do dia seguinte.

1. Implemente uma função para cadastrar um **cliente**. Essa 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**. Essa função deve garantir que não haverá mais de um funcionário cadastrado com o mesmo código. Se quiser pode gerar o código automaticamente.
3. Implemente uma função para cadastrar um **quarto**. Essa função deve garantir que não haverá mais de um quarto cadastrado com o mesmo número.
4. Implemente uma função que cadastre uma **estadia**. Para cadastrar a estadia, o sistema deve receber do usuário o código 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 a quantidade de hóspedes desejada. Além disso, a quantidade de diárias deverá ser calculada com base nas datas de entrada e saída.
5. Implemente uma função que dê baixa em uma determinada estadia, calcule e mostre o valor total a ser pago por um determinado cliente. Lembre-se de alterar o *status* do quarto para desocupado.
6. Implemente funções para realizar pesquisas tanto por clientes quanto por funcionários, ou seja, digitando o nome ou código apresenta na tela todas as informações do cliente ou funcionário correspondente.
7. Implemente uma função que mostre na tela todas as estadias de um determinado cliente (a pesquisa deve se basear no nome ou código do cliente).
8. Implemente uma função que calcule 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 implementar este programa pode ser necessário criar mais funções do que as que estão descritas.

Finalmente, implemente 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, um quarto ou uma estadia. Além disso, permitir realizar as pesquisas e dar baixa em estadias. Este *menu* deve

ficar em *loop* até o usuário selecionar a opção SAIR. Além disso, todas as informações deverão ser **persistidas/armazenadas em arquivos binários** (em estruturas heterogêneas). Portanto, deverá ser feita leitura e escrita em arquivos. Utilize **bibliotecas** para organizar os módulos (funções e procedimentos) do *software*.

* Não é obrigatória a implementação do programa usando ***Orientação a Objetos***, mas caso o grupo queira usar, é permitida.

Metodologia

Este é um trabalho interdisciplinar em que você deve planejar, analisar, projetar, implementar e testar uma solução de *software* para o problema apresentado utilizando o Scrum para gerenciar seu progresso.

Inicialmente organize o *backlog* do produto com as funções básicas do sistema. Cada um dos módulos (procedimentos ou funções) será de responsabilidade de um membro do grupo e será desenvolvido em *sprints* de 3 a 4 dias. Seguem algumas sugestões de atividades a serem realizadas nas *sprints*:

1. Definir a assinatura do(s) módulo(s) (procedimento(s) ou função(ões)). Reflita sobre os parâmetros de entrada e saída do módulo e comunique aos seus colegas de projeto.
2. Documentar o módulo (procedimento ou função) indicando seu propósito, parâmetros de entrada e saída. O nome do módulo deve ser escolhido sob o ponto de vista de quem o usa ou de quem vai chamar o módulo e deve refletir o que ele faz.
3. Implementar o caso de sucesso do módulo (procedimento ou função).
4. Selecionar casos de teste para verificar o funcionamento de cada módulo. Um caso de teste deve conter os valores de entrada para o módulo e a saída esperada.
5. Executar os casos de teste planejados para o módulo.
6. Criar um relatório de execução de testes que contenha os casos de teste, a saída retornada durante sua execução e uma indicação se o módulo passou ou não no teste. Isso é feito comparando-se a saída esperada, documentada no caso de teste, com a saída retornada durante a execução do módulo (esperado x real).
7. Implementar os casos especiais, exceções que possam existir no módulo (procedimento ou função). Em seguida, executar os casos de teste anteriores para garantir que as mudanças não quebraram o código anterior que já funcionava. Pense também nos novos casos de teste necessários para a nova versão do módulo (procedimento ou função).

O que deve ser entregue para os professores no Canvas

1. A evolução do *backlog* do produto a cada *sprint*. Indique quais tarefas encontravam-se inicialmente no *backlog* do produto, e em qual *sprint* cada tarefa foi alocada, juntamente com seu responsável.
2. A documentação das funcionalidades do *software*.
3. O planejamento dos casos de teste (entradas, procedimento de teste e saídas esperadas) e o relatório de execução dos testes.
4. O código, em C, dos módulos (funções e procedimentos) em bibliotecas e do programa principal, juntamente com o projeto completo do *software*.
5. Arquivos contendo dados já incluídos para teste das funcionalidades.
6. Apresentação gravada em vídeo (*pitch*) mostrando todas as funcionalidades do sistema.