

SISTEMA DE HOTEL

O presente relatório tem por objetivo descrever o código desenvolvido pelo nosso grupo, do curso de Análise e desenvolvimento de sistemas, turma E, do Centro Universitário SENAC, para um sistema de reservas de um hotel.

1. INTRODUÇÃO

O código desenvolvido é um sistema de reserva de suítes para um hotel. Ele permite cadastrar hóspedes, listar hóspedes, fazer reservas e listar as reservas existentes. Existe um menu principal que orienta os usuários através das opções disponíveis. Com essa estrutura organizada, o sistema simplifica o gerenciamento de reservas de acomodações para o hotel e oferece uma experiência intuitiva para os usuários.

2. ESTRUTURA DO CÓDIGO

A linguagem utilizada para desenvolver o código foi a Java, e ele está organizado em diversas classes, contendo as principais funcionalidades do sistema:

2.1 Menu: É a classe principal que gerencia o fluxo do programa e a interação com o usuário.

2.2 Suítes: Classe que gerencia as reservas das acomodações.

2.3 lerDados: Classe utilizada para leitura de dados do console.

2.4 Arquivo: Classe utilizada para operações com arquivos.

3. FUNCIONALIDADE DA CLASSE PRINCIPAL

3.1 Menu

O código apresenta um menu com opções para o usuário:

- Cadastrar hóspede: Permite cadastrar um novo usuário (hóspede) fornecendo informações como Nome, Data de Nascimento, CPF/Passaporte/RNE, Nacionalidade, Endereço, Número, Complemento e CEP.
- Fazer reserva: Apresenta um segundo menu, que exibe as suítes que estão disponíveis e permite que o hóspede escolha e reserve a suíte desejada, além da opção de “voltar” que retorna ao menu principal.
- Listar hóspedes: Lista todos os hóspedes cadastrados, acompanhado das informações de cada um.
- Listar reservas: Lista todas as reservas feitas pelo sistema.
- Cancelar/Alterar Reserva: Permite cancelar ou alterar uma reserva existente em nosso sistema.

- Sair do programa: Encerra a execução do programa.

4. FLUXO DO PROGRAMA

- 4.1 Inicialização: O método `main` carrega os dados dos arquivos `Hospede.txt` e `Reservas.txt`, se existirem.
- 4.2 Menu Principal: O usuário é apresentado com o menu principal, onde pode escolher entre as opções disponíveis (cadastrar hóspede, fazer reserva, listar hóspedes, listar reservas, cancelar/alterar reserva e sair).
- 4.3 Cadastro de Hóspede: Coleta os dados do hóspede e os salva em `Hospede.txt`.
- 4.4 Reserva de Quartos: Permite ao usuário fazer reservas, verificar a disponibilidade de quartos e cancelar reservas. As reservas são salvas em `Reservas.txt`.
- 4.5 Listagem de Hóspedes e Reservas: Exibe os dados dos hóspedes e das reservas.

5. DETALHAMENTO DAS CLASSES

5.1 Menu:

- Atributos:

`hospedes`: Lista de hóspedes cadastrados.

- Métodos:

`main(String[] args)`: Método principal que inicia o programa, carregando dados dos arquivos se existirem, e chama o menu principal.

`menuPrincipal()`: Exibe o menu principal e gerencia a navegação entre as opções.

`cadaststrarHospede()`: Realiza o cadastro de um novo hóspede.

`listarHospedes()`: Lista todos os hóspedes cadastrados.

5.2 Suítes:

- Atributos:

`achei`: Referência ao hóspede encontrado durante a reserva.

`reservas`: Mapa que armazena as reservas, onde a chave é o número do quarto e o valor é uma instância da classe `Reserva`.

- Classes Internas:

`Reserva`: Representa uma reserva com nome do hóspede, data de check-in e data de check-out.

- Métodos:

`ReservaDeQuartos()`: Gerencia o processo de reserva de

`quartos.exibirReserva()`: Exibe o status das reservas dos quartos.

`fazerReserva()`: Realiza a reserva de um quarto.

`listar()`: Lista todas as reservas feitas.

`cancelarReserva()`: Cancela uma reserva existente.

```
transformarEmStrings(Map<Integer, Reserva> reservas):  
Transforma as reservas em uma lista de strings.  
lerDaLista(List<String> lista): Constrói um mapa de reservas a  
partir de uma lista de strings.  
salvarReservasEmArquivo(Map<Integer, Reserva> reservas): Salva  
as reservas em um arquivo.  
carregarReservasDoArquivo(): Carrega as reservas a partir de um  
arquivo.
```

5.3 lerDados:

- Métodos:

lerInt(): Lê um número inteiro do console

lerTexto(): Lê uma string do console.

lerData(): Lê uma data no formato dd/MM/uuuu do console.

Métodos adicionais para leitura de diferentes tipos de dados (inteiros, double, BigDecimal, data e hora).

5.4 Arquivo:

- Métodos:

arquivoExiste(String arquivo): Verifica se um arquivo existe.

lerLinhas(String arquivo): Lê as linhas de um arquivo.

escreverLinhas(List<String> linhas, String arquivo): Escreve uma lista de strings em um arquivo.

6. CONSIDERAÇÕES FINAIS

O sistema oferece uma interface de linha de comando para gestão de um hotel, com funcionalidades essenciais para cadastro e gerenciamento de hóspedes e reservas. A persistência de dados em arquivos de texto permite que as informações sejam mantidas entre execuções do programa, garantindo uma operação contínua e confiável. Este é um projeto de estudantes do primeiro semestre, e ainda visa implementar melhorias ao longo do curso e demais semestres, para oferecer uma melhor experiência aos usuários e garantir maior segurança de dados.