



UNIVERSIDADE ESTADUAL DE CAMPINAS

ALINE HESLEY SILVA SOUSA
RA 248490

FRANCISCO VINICIUS SOUSA GUEDES
RA 260440

MARIA EDUARDA VIEIRA RANGEL
RA 241035

THIAGO MOTA MARTINS
RA 223485

PROJETO FINAL

MC322 - Programação Orientada a Objetos

Campinas
2023

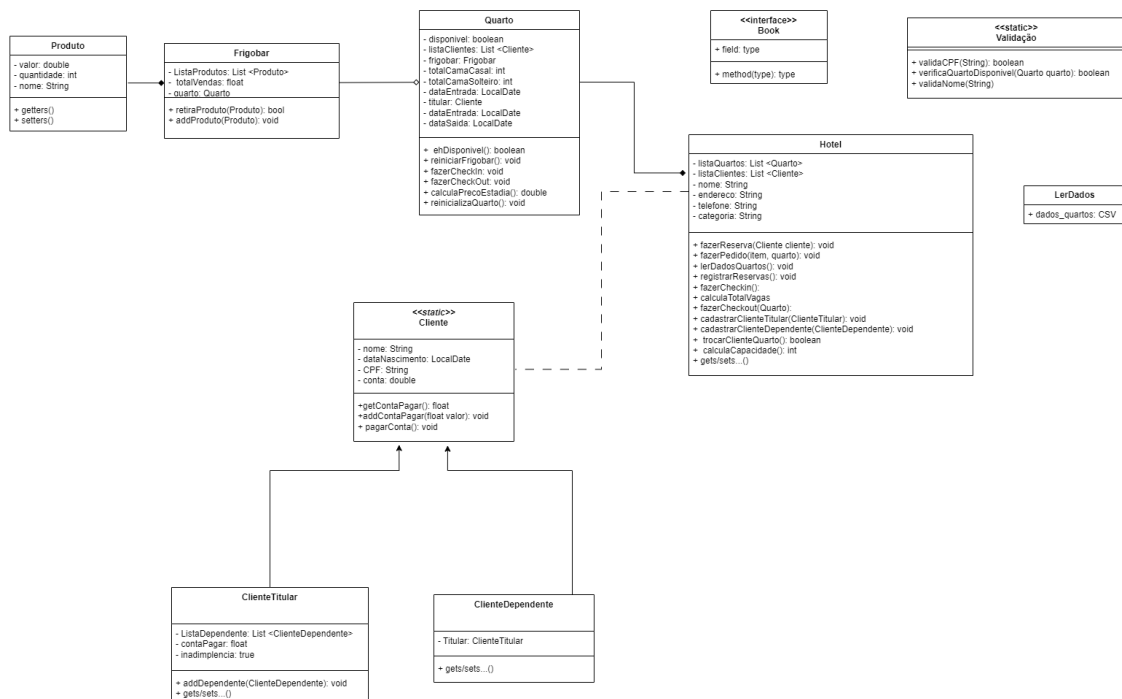
GERENCIAMENTO DE UM HOTEL

Objetivo do projeto

O objetivo do projeto consiste em criar e construir um sistema que, conforme definido pelo grupo, trata-se do gerenciamento de um hotel, no qual o atendente faz o controle dos quartos do hotel, verificação dos preços da hospedagem, e todas as demais atividades necessárias para o funcionamento pleno do mesmo. Contemplando a programação orientada a objetos, o sistema foi descrito e implementado em sua versão completa, de modo que pode ser manipulado por parte do usuário.

Visão geral do sistema

Para melhor compreensão do sistema, é possível visualizar o UML do programa a seguir:



Conforme visto acima, cada classe consta com seus atributos e principais métodos. Para visualização completa das estruturas de cada uma, recomenda-se olhar os códigos disponíveis no github, acessível através do link: <https://github.com/alinehesley/HotelManager>

Funcionamento do sistema

Para a criação do sistema para o gerenciamento de um hotel, as funções que constam na interface gráfica Book para manipulação do programa são:

- Cadastro de clientes;
- Criação de reserva, com a checagem da capacidade dos quartos a serem reservados com o número de clientes a serem alocados;
- Cálculo da capacidade disponível do hotel;
- *Check-in*;

- Leitura dos dados de um quarto;
- Troca de quarto;
- *Check-out*;
- Reinicialização de quartos e frigobares; e
- Cálculo da conta final, considerando tempo de estadia e consumo de alimentos no frigobar.

Tais funções foram exibidas através de um menu principal, com o objetivo de coletar as informações necessárias e realizar as funcionalidades básicas para o funcionamento pleno de um hotel.

- *Relacionamentos*

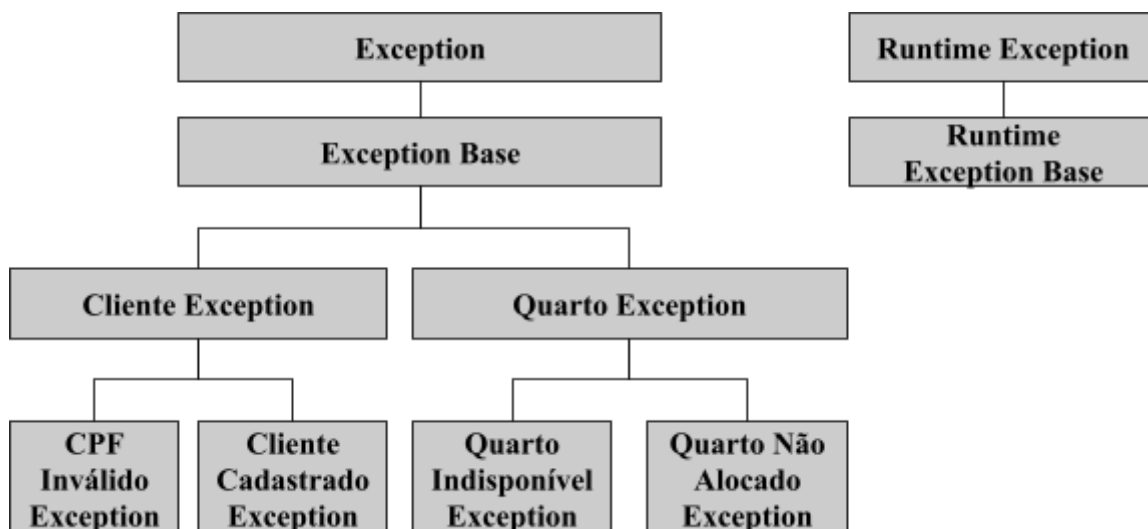
Conforme visto no UML, a classe Hotel contém relacionamentos com as classes Cliente e Quarto, para gerenciamento e manipulação dos clientes cadastrados no hotel, através de uma associação, e também do hotel com o quarto, por meio da composição, no qual o hotel necessita de quartos para existir.

Além disso, também há a agregação do Quarto com o Frigobar e a composição do Frigobar com a classe Produto, no qual cada quarto contém um frigobar, mas não necessita dele para existir, enquanto o frigobar só existe com uma lista de produtos limitada que pode ser consumida pelos hóspedes alocados.

Por fim, também há o caso de herança, utilizado entre a classe-mãe Cliente e as classes-filhas Cliente Titular e Cliente Dependente. Assim, será possível associar o quarto e a conta com um cliente principal e todos os demais hóspedes associados ao cliente titular ainda constarem na lista de hóspedes do hotel, mas sem terem que ter reservas individuais para cada um.

- *Tratamento de Exceção e Validações*

O tratamento de exceção ocorreu por meio de uma cascata de operações, de modo a tratar todos os possíveis erros que possam ocorrer no processo. Tem-se, portanto:



- *QuartoNaoLocadoException*: será lançado quando o gerenciador tentar realizar atividades envolvendo um hóspede, como o cálculo do seu tempo de estadia, em um

quarto que não está alocado. Nesse caso, o usuário será informado sobre o erro e solicitará uma nova entrada.

- *ClienteCadastradoException*: alerta quando o atendente tentar cadastrar um cliente que já possui registro no hotel.
- *CPFInvalidoException*: checagem importante para pesquisas envolvendo hóspedes; de modo a informar quando um CPF pesquisado não estiver disponível nos registros.
- *QuartoIndisponivelException*: ocorre na tentativa de fazer *check-in* em um quarto que já está ocupado.
- *RuntimeExpectionBase*: usada para prevenir uma possível exceção não prevista; comumente usada em erros de programação que só podem ser verificados em tempo de execução, como *ArrayIndexOutOfBoundsException* e *NullPointerException*.

Já na classe Validação também ocorrem verificações, como a validação do CPF e nome inseridos e também da capacidade da reserva, por meio da comparação do número de camas com o número de hóspedes a serem alocados.

- *Arquivos*

O projeto inicia com um arquivo pronto de quartos, com seus números para identificação e a quantidade de camas de solteiro e de casal em cada um. Assim, o usuário não precisa cadastrar toda a estrutura do hotel para iniciar o programa.

O sistema também conta com o tratamento de arquivos para armazenar os detalhes dos clientes, quartos reservados e pedidos em um arquivo, quando o programa é encerrado. Desse modo, ao reiniciar o programa, essas informações podem ser lidas para restaurar o estado anterior do hotel.

Implementação completa do sistema

Para visualização da implementação do programa e teste das funcionalidades, um vídeo está disponível a seguir mostrando a manipulação do programa desenvolvido. O mesmo está disponível no Github apresentado anteriormente.