



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

JAIME DAVID VERDERESI

**SISTEMA EM JAVA WEB PARA GESTÃO DE HOTEL E
POUSADA**

**Assis/SP
2023**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

JAIME DAVID VERDERESI

SISTEMA EM JAVA WEB PARA GESTÃO DE HOTEL E POUSADA

Projeto de pesquisa apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a): Jaime David Verderesi
Orientador(a): Prof. Me. Guilherme de Cleve
Farto**

**Assis/SP
2023**

FICHA CATALOGRÁFICA

Verderesi, Jaime David

V483s Sistema em Java Web para gestão de hotel e pousada / Jaime David Verderesi. -- Assis, 2023.

79p.

Trabalho de Conclusão de Curso (Análise e Desenvolvimento de Sistemas) -- Fundação Educacional do Município de Assis (FEMA), Instituto Municipal de Ensino Superior de Assis (IMESA), 2023.

Orientador: Prof. Me. Guilherme de Cleve Farto.

1. Linguagem de programação. 2. Ambientes de programação.
3. Hotelaria. I Farto, Guilherme de Cleve. II Título.

CDD 003

SISTEMA EM JAVA WEB PARA GESTÃO DE HOTEL E POUSADA

JAIME DAVID VERDERESI

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Prof. Me. Guilherme de Cleva Farto

Examinador: _____
Inserir aqui o nome do examinador

**Assis/SP
2023**

DEDICATÓRIA

Dedico este trabalho às minhas duas filhas,
Renata e Paula e aos meus netos, Luisa (Luluca)
e Francisco (Tico) e às pessoas que
contribuíram para esta empreitada.

AGRADECIMENTOS

Primeiramente a Deus, autor da vida e fonte de sabedoria, que me concedeu forças para superar os desafios e orientou meus passos ao longo desta jornada acadêmica.

Aos professores que passaram pela minha vida, ensinando e orientando, durante a minha carreira acadêmica. Principalmente aos professores da FEMA que além de professores muitos deles eram colegas.

Gostaria de expressar meus sinceros agradecimentos a todas as pessoas que contribuíram para o sucesso da minha jornada acadêmica, à coordenadora do curso Prof. Me. Diomara Martins Reigato Barros, suas orientações, conhecimentos, dedicação, paciência e apoio constante tenham sido úteis para manter o curso uma referência.

Principalmente agradeço ao meu orientador Prof. Me. Guilherme de Cleve Farto, pela orientação, que com sua sabedoria, dedicação e paciência não deixou sem resposta nenhuma de minhas dúvidas desde as mais simples até as mais avançadas.

Obrigado a todos...

**Eu acredito que às vezes são as pessoas
que ninguém espera nada que fazem as
coisas que ninguém consegue imaginar.**

Alan Turing.

RESUMO

Com o surpreendente avanço nas áreas de Tecnologia da Informação (TI) a utilização das novas tecnologias que surgem é de fundamental importância no setor de hotelaria e pousada, pois novas tecnologias permitem que estas empresas se tornem mais competitivas frente aos seus concorrentes. O surpreendente avanço do *hardware* e a queda nos preços, sendo acessível para a maioria das pessoas e empresas, aliado a este fato tem-se ainda que a metodologia de desenvolvimento de *software* sofra avanços no que tange ao seu desenvolvimento com a adoção da metodologia ágil. O setor de hotelaria e pousadas não foge a estas regras e é considerada um dos mais importantes segmentos da economia nacional, compreendendo, deste modo, na base do setor de turismo, o que permite a geração de empregos, melhorando a qualidade de vida da população. O Sistema deste projeto é desenvolvido inicialmente utilizando a linguagem Java com a plataforma Java EE, em conjunto dos frameworks: JPA para mapeamento objeto e relacional, *PrimeFaces*, para o desenvolvimento da parte visual e JSF que disponibiliza um conjunto de componentes específicos para o desenvolvimento rápido.

Palavras-chave: *software*, ágil, java, JPA, JSF, JavaEE.

ABSTRACT

With the surprising progress in the areas of Information Technology (IT) the use of new technologies that arise is of fundamental importance in the hotel and inn sector, as new technologies allow these companies to become more competitive compared to their competitors. The surprising advancement of hardware and the drop in prices, being accessible to most people and companies, combined with this fact, the software development methodology still suffers advances in terms of its development with the adoption of the agile methodology . The hotel and inn sector is no exception to these rules and is considered one of the most important segments of the national economy, comprising, therefore, at the base of the tourism sector, which allows the creation of jobs, improving the quality of life of the population . The system of this project is initially developed using the Java language with the Java EE platform, together with the frameworks: JPA for object and relational mapping, PrimeFaces, for the development of the visual part and JSF that provides a set of specific components for rapid development .

Keywords: software, agile, java, JPA, JSF, JavaEE.

LISTA DE ILUSTRAÇÕES

- MySQL – SGBD	22
- PrimeFaces	23
- Popularidade dos frameworks através do Google Trends	24
- WBS.....	25
- Sequenciamento de Atividades	26
- Mapa Mental.....	27
- Caso de Uso Geral	30
- Caso de Uso Manter Hóspedes.....	31
- Caso de Uso Realizar Check-in.....	32
- Caso de Uso Realizar Check-out	35
- Caso de Uso Consultar Reserva	37
- Caso de Uso Efetuar Reserva	40
- Caso de Uso Consultar Hóspede	42
- Caso de Uso Consultar Apartamento	44
- Caso de Uso Manter Produto	46
- Caso de Uso Manter Usuário	48
- Caso de Uso Manter Apartamento	50
- Caso de Uso Emitir Relatórios.....	52
- Diagrama de Classe	56
- Diagrama de Entidade Relacionamento (DER)	59
- Diagrama de Atividade – Realizar Check-in	60
- Diagrama de Atividade – Manter Apartamento.....	60
- Diagrama de Atividade – Alterar Reserva.....	61
- Diagrama de Sequência Cadastrar Funcionário.....	62

- Diagrama de Sequência Cadastrar Cliente.	63
- Diagrama de Sequência Cadastrar Fornecedor.	64
- Diagrama de Sequência Cadastrar Produto.	65
- Diagrama de Sequência Cadastrar Cidade.	66
- Diagrama de Sequência Cadastrar Pagamento.	67
- Diagrama de Sequência Consultar Quartos.	68
- Organização dos Pacotes e Classes.....	69
- Organização das Páginas.....	70
- Tela Inicial do Sistema.....	71
- Tela do Sistema.....	72
- Tela de Cadastro de Usuários.....	73
- Tela de Consumo por Quarto.....	74
- Tela exibe Consumos e itens.....	75

LISTA DE TABELAS

Tabela 1 – Caso de Uso Manter Hospedes	32
Tabela 2 - Caso de Uso Realizar Check-in.....	34
Tabela 3 - Caso de Uso Realizar Check-out.....	36
Tabela 4 - Caso de Uso Consultar Reserva	39
Tabela 5 - Caso de Uso Efetuar Reserva	41
Tabela 6 - Caso de Uso Consultar Hospede	43
Tabela 7 - Caso de Uso Consultar Apartamento	46
Tabela 8 - Caso de Uso Manter Produto	48
Tabela 9 - Caso de Uso Manter Usuário.....	50
Tabela 10 - Caso de Uso Manter Apartamento	52
Tabela 11 - Caso de Uso Emitir Relatórios.....	54

SUMÁRIO

1. INTRODUÇÃO	13
1.1. OBJETIVOS	14
1.1.1. 1.1.1. Objetivo Geral	14
1.1.2. Objetivo Específico	15
1.2. JUSTIFICATIVAS.....	15
2. TECNOLOGIAS E FERRAMENTAS PARA DESENVOLVIMENTO DO SISTEMA	18
2.1.1. 2.1.6. <i>MySQL</i>	21
3. PLANEJAMENTO DO PROJETO.....	25
3.1. 3.1. WBS (WORK BREAKDOWN STRUCTURE /ESTRUTURA ANALÍTICA DE TRABALHO - EAP).....	25
3.2. SEQUENCIAMENTO DE ATIVIDADES.....	26
4. ANÁLISE E ESPECIFICAÇÃO DO TRABALHO	27
4.2. MAPEAMENTO DE REQUISITOS	28
4.4. DIAGRAMA DE CLASSES.....	55
4.6. DIAGRAMAS DE ATIVIDADES	60
4.7. DIAGRAMA DE SEQUÊNCIA	61
5. IMPLEMENTAÇÃO DO SISTEMA.....	69
5.3.1. Tela Inicial.....	71
5.3.2. Tela do Sistema	72
5.1.1. 5.3.3. Tela de Cadastro de Clientes.....	73
5.3.4. Tela de Consumos	74
6. CONCLUSÕES E TRABALHOS FUTUROS.....	76
6.2. TRABALHOS FUTUROS	76

1. INTRODUÇÃO

A gestão de hotéis e pousadas é uma atividade complexa que envolve muitas atividades, como reservas, *check-in* e *check-out* de hóspedes, gerenciamento de quartos, gestão financeira e contábil, entre outras. Para facilitar esse processo, um sistema em Java Web ajuda a automatizar essas tarefas e tornar a gestão mais eficiente.

É comum desenvolver sistemas Java para a Web seguindo uma arquitetura de três camadas, também conhecida como arquitetura *Model-View-Controller* (MVC).

O MVC é utilizado em muitos projetos devido à arquitetura que possui, o que possibilita a divisão do projeto em camadas muito bem definidas. Cada uma delas – *model* (modelo), *view* (visão) e *controller* (controlador) – executa o que lhe é definido e nada mais do que isso (PORTELA; QUEIRÓS, 2018).

Os objetivos centrais do MVC são a reutilização de código e a separação de conceitos. O MVC constitui-se, então, como um padrão de arquitetura de *software* que separa a representação da informação da interação do usuário com o próprio *software*, em que:

Model (modelo) – corresponde aos dados da aplicação, às regras de negócios, à lógica e às funções. O modelo gera o processo de negócio, responde a pedidos do controlador e apresenta os resultados na *view*.

View (visão) - pode ser qualquer saída da representação dos dados, como uma tabela ou um diagrama. É possível haver diversas vistas do mesmo dado, como um gráfico de barra ou uma tabela.

Controller (controlador) – faz a mediação da entrada, convertendo-a em comandos para o *model* ou a *view*. É baseado em comportamentos, podendo ser compartilhado por várias *views*, e responsável por determinar qual o tipo de exibição na *view* (PORTELA; QUEIRÓS, 2018).

O sistema para gestão de hotéis e pousadas pode ter funcionalidades como reservas de quartos, gerenciamento de *check-in* e *check-out* de hóspedes, gerenciamento de tarifas e disponibilidade de quartos, controle de estoque de produtos, gestão financeira, emissão de relatórios, entre outras funcionalidades.

Ao implementar um sistema em Java Web para gestão de hotéis e pousadas, é importante garantir que ele seja seguro, escalável e de fácil manutenção. Também é importante garantir que o sistema atenda às necessidades específicas do hotel ou pousada em questão, garantindo uma boa experiência tanto para os hóspedes quanto para os funcionários responsáveis pela gestão.

O grande desenvolvimento na área da tecnologia da informação e cada vez mais presente no cotidiano das pessoas, não justifica que empresas dos mais diversos setores continuem administrando seus negócios sem o auxílio destas novas tecnologias. A transformação digital e as inovações vieram para mudar o comportamento das pessoas e empresas ao realizar determinado processo.

O uso de novas tecnologias para administrar empresas de qualquer ramo de atividade, é um requisito fundamental a fim de tornar os diversos processos mais seguros, eficientes e realizá-los com maior rapidez.

Os hotéis e pousadas não fogem a esta regra, mesmo que sejam de pequeno ou médio porte, são obrigados a fazer uso de algum tipo de sistema informatizado no seu cotidiano. Mesmo que a satisfação e o conforto do hóspede ocupam lugar de destaque, temos que ter em mente que grande parte dos serviços de um hotel ou pousada pode ser diminuído ao adotar-se um sistema de gestão, e com isto aumentar a eficiência e tornar mais ágil a execução de tarefas ao mesmo tempo que aumenta o controle operacional.

1.1. OBJETIVOS

1.1.1. Objetivo Geral

O objetivo deste trabalho é apresentar a análise, o projeto e a implementação de um sistema Web de gerenciamento de hotéis e pousadas. Para o desenvolvimento foi utilizada a linguagem Java e diversas tecnologias presentes nos dias atuais com a finalidade de facilitar e tornar mais seguro o desenvolvimento.

1.1.2. Objetivo Específico

O objetivo específico deste trabalho é o desenvolvimento do controle de entrada (*check in*) e saída (*check out*) de hóspedes, consumo por apartamentos, emissão de relatórios diários, cadastro de clientes, funcionários e usuários, efetuar reservas.

Foi desenvolvido um sistema onde o ramo hoteleiro consiga estabelecer o contato necessário com seus clientes, e fazer a gerência dos principais fluxos do hotel ou pousada. Além disso, o sistema objetiva garantir a segurança das informações dos hóspedes, protegendo seus dados pessoais e financeiros.

1.2. JUSTIFICATIVAS

Desenvolver um *software* de gestão para hotéis ou pousada é fundamental para otimizar as operações diárias do estabelecimento, aumentar a eficácia e a qualidade dos serviços oferecidos, além de melhorar a experiência do cliente. Este software deve ter o mínimo de funcionalidades que atendam às necessidades dos usuários, e que seja suficiente o bastante para executar as tarefas essenciais da empresa.

Outras justificativas importantes são as seguintes: aumento da eficiência, melhoria da experiência dos clientes, redução de erros causados por humanos, *check-in* e *check-out* rápido e fácil, maior controle, competitividade no mercado, entre outras.

Essas são apenas algumas das justificativas para a criação de um sistema para gestão de hotel e pousada. É importante lembrar que cada hotel ou pousada tem suas particularidades e, portanto, os benefícios podem variar de acordo com as necessidades específicas de cada estabelecimento.

1.3. MOTIVAÇÃO

A principal motivação para o desenvolvimento deste projeto é a possibilidade de aprofundar os conhecimentos na linguagem Java e seus inúmeros *frameworks*.

Além desta, outra motivação para este empreendimento é tomar conhecimento de uma nova arquitetura que vem se tornando cada vez mais popular nos últimos anos, *microsserviços*.

Outra motivação que conduziu este empreendimento é aplicar os conhecimentos adquiridos e os que irei adquirir com a realização desta tarefa, de forma a desenvolver uma aplicação que tenha uma interface amigável e as operações seja simples o suficiente para realizar os trabalhos que propõe.

1.4. PÚBLICO ALVO

O público-alvo para um sistema em Java Web para gestão de hotel e pousada é composto por gestores e funcionários desses estabelecimentos, que buscam uma ferramenta para gerenciar e automatizar os processos do negócio.

Mais especificamente, o público-alvo pode incluir os gerentes e proprietários de hotéis e pousadas, funcionários responsáveis pela recepção, que precisam fazer *check-in* e *check-out* dos hóspedes, gerenciar reservas, controlar a ocupação dos quartos e atender solicitações dos clientes, funcionários do setor de limpeza, bem como outras áreas do estabelecimento, funcionários do setor financeiro, hóspedes, que podem usar o sistema para fazer reservas, consultar informações sobre o estabelecimento e interagir com a equipe de atendimento.

1.5. ESTRUTURA DO TRABALHO

Será dividido em seis capítulos, onde cada um descreve melhor cada passo do trabalho.

O capítulo 1 será Introdução, é nesta parte que estão descritos os objetivos, justificativas, motivações, público alvo e a estrutura do trabalho. Capítulo 2 está descrito as tecnologias e

ferramentas para desenvolvimento do sistema. As tecnologias utilizadas são, Plataforma Java, *JavascriptFaces* (JSF), Persistência com JPA, *Hibernate*, *Model, View, Controller* (MVC), *MySQL*, *Primefaces*. Capítulo 3 é o planejamento do projeto, estrutura analítica do projeto, sequenciamento das atividades. Capítulo 4 é a análise e especificação do trabalho, mapa mental, mapeamento de requisitos, diagrama de especificação de casos de uso, diagrama de classes, diagrama de entidade-relacionamentos (DER), diagrama de atividades, diagrama de sequência. Capítulo 5 é a implementação do sistema, organização dos pacotes e classes do sistema, organização das páginas e organização das telas: tela inicial, tela do sistema, tela de cadastro de clientes, tela de consumos, tela que relaciona os consumos e seus respectivos itens. Capítulo 6 consiste das conclusões e trabalhos futuros.

2. TECNOLOGIAS E FERRAMENTAS PARA DESENVOLVIMENTO DO SISTEMA

O presente capítulo descreve as tecnologias que eventualmente são utilizadas para o desenvolvimento do sistema.

2.1. TECNOLOGIAS UTILIZADAS

A ferramenta escolhida para o desenvolvimento foi o IDE Eclipse JEE-2023-03, que utiliza a linguagem de programação Java, por ser uma linguagem *open source* e Orientada a Objetos. O sistema proposto é desenvolvido para Web. Para a Análise Orientada à Objetos, são utilizados os conceitos do Padrão de Linguagem de Modelagem Unificada (UML).

O Banco de Dados utilizado é o MySQL que é um Sistema de Gerenciamento de Banco de Dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language* – Linguagem de consulta estruturada) como interface e o *framework* que fará o relacionamento com o banco de dados será o Hibernate.

2.1.1. Plataforma Java

A linguagem Java Web foi utilizada no desenvolvimento do sistema. Segundo Lockow et al. (2010, pg.25-26), a linguagem Java começou a surgir em 1991 na *Sun Microsystems*. Inicialmente era parte de outro projeto, chamado *Green Project*, que tinha como objetivo possibilitar a convergência entre computador, equipamentos eletrônicos e eletrodomésticos. Na época a linguagem era chamada de Oak (em português, “carvalho”), pois era essa árvore que James Gosling avistava da janela da sede do projeto, e o resultado desse projeto foi um controle remoto chamado “7(*Star Seven*), que possuía uma interface gráfica sensível ao toque e tinha a capacidade de interagir com diversos equipamentos.

Na época em que surgiu, a linguagem Oak foi desenvolvida pra controlar internamente um equipamento, mas o controle remoto não teve sucesso, já que estava muito avançado para o seu tempo, uma vez que as empresas de TV a cabo e vídeo por demanda não possuíam recursos para sustentar tal empreendimento. Posteriormente, James Gosling foi responsável por adaptar a linguagem Oak para a internet, resultando no surgimento da plataforma Java em 1995.

2.1.2. Javasever Faces (JSF)

O *JavaServer Faces* (JSF) é uma tecnologia padrão do J2EE 1.4 (ou superior) para criar aplicações web. Ele herda das tecnologias JSP e *Servlets* e estende seus conceitos com um ciclo de vida e um conjunto de componentes e recursos sofisticados e focados no desenvolvimento RAD para web (GOMES, 2008)

No caso do *JavaServer Faces*, o fato de ser uma especificação do JCP (www.jcp.org/en/jsr) significa que grandes empresas como Apache, BEA Systems, EDS, EMC, Fujitsu, Hewlett-Packard, IBM, Macromedia, Novell, Oracle, Siemens e Sun participaram na definição do *JavaServer Faces* e aprovaram sua especificação. O fato de ser uma especificação formal e segura permite que essas empresas e outras que se interessarem invistam na tecnologia e desenvolvam ferramentas e componentes para o desenvolvimento web com JSF (LUCKOV; MELO (2010, p.72).

2.1.3. Persistência com JPA (PERSISTÊNCIA/ORM)

A *Java Persistence API* (JPA) é um framework para persistência em Java, que oferece uma API de mapeamento objeto-relacional e soluções para integrar persistência com sistemas corporativos escaláveis.

Com JPA, os objetos são POJO (*Plain Old Java Objects*), ou seja, não é necessário nada de especial para tornar os objetos persistentes. Basta adicionar algumas anotações nas classes que representam as entidades do sistema e começar a persistir ou consultar objetos.

JPA é uma especificação, e não um produto. Para trabalhar com JPA, precisamos de uma implementação.

O projeto do *Hibernate* ORM possui alguns módulos, sendo que o *EntityManager* é a implementação da JPA que encapsula o *Core*. O *Core* é a base para o funcionamento da persistência, com APIs nativas e metadados de mapeamentos em arquivos XML. Possui uma linguagem de consultas chamada HQL (parecido com SQL), um conjunto de interfaces para consultas usando critérios (*Criteria* API), e outros (FARIA, 2015).

2.1.4 *Hibernate*

O projeto do *Hibernate* ORM possui alguns módulos, sendo que o *EntityManager* é a implementação da JPA que encapsula o *Core*. O *Core* é a base para o funcionamento da persistência, com APIs nativas e metadados de mapeamentos em arquivos XML. Possui uma linguagem de consultas chamada HQL (parecido com SQL), um conjunto de interfaces para consultas usando critérios (*Criteria* API), e outros (FARIA, 2015).

As interfaces `org.hibernate.Query` e `org.hibernate.Criteria`, ambas definem vários métodos para o controle de execução de uma consulta. Em adição, a *Query* fornece métodos para vincular valores concretos aos parâmetros da consulta. Para executar uma consulta na sua aplicação, você precisa obter uma instância de uma dessas interfaces usando a *Session* (BAUER; KING, 2007).

2.1.5. *Model, View, Controller* (MVC)

MVC é um conceito (paradigma) de desenvolvimento e design que tenta separar uma aplicação em três partes distintas. Uma parte, a *Model*, está relacionada ao trabalho atual que a aplicação administra, outra parte, a *View*, está relacionada a exibir os dados ou informações dessa aplicação e a terceira parte, *Controller*, em coordenar os dois anteriores exibindo a interface correta ou executando algum trabalho que a aplicação precisa completar.

Model: O model é o objeto que representa os dados do programa. Manipula os dados e controlam todas suas transformações, portanto, o Model é a classe que trabalha no armazenamento e busca de dados.

View: A *view* é o objeto que manipula apresentação visual dos dados representados pelo model.

Controller: O *controller* é o objeto que responde as ordens executadas pelo usuário, atuando sobre os dados apresentado pelo modelo, decidindo como o modelo deverá ser alterado ou deverá ser revisto e qual apresentação deverá ser exibida (GONÇALVES, EDSON, 2008).

2.1.6. MySQL

O MySQL surgiu a partir da necessidade da equipe que criou o SGBD, de utilizar algum mecanismo que permitisse a conexão de tabelas criadas na linguagem SQL para um determinado fim. A princípio, o grupo iria utilizar o MySQL, mas logo perceberam que esta ferramenta não era rápida o suficiente para atender às necessidades do projeto. O jeito foi criar uma solução própria. Nascia o MySQL. A figura 1 ilustra as características essenciais do MySQL. O MySQL é um banco de dados relacional gratuito, eficiente e otimizado para aplicações Web, é desenvolvido e mantido pela empresa MySQL AB, que também oferece uma versão comercial (paga). Esse SGBD também é multi-plataforma, sendo compatível com o Windows, Linux, BSDs, entre outros sistemas operacionais. As tabelas criadas podem ter tamanho de até 4 GB. Fora isso, o MySQL é compatível com várias linguagens de programação, tais como PHP, C, Java, Visual Basic, entre outros. MySQL é uma linguagem de propósito específico para acesso a banco de dados, e interpretada (SANTOS; SILVA S/D).

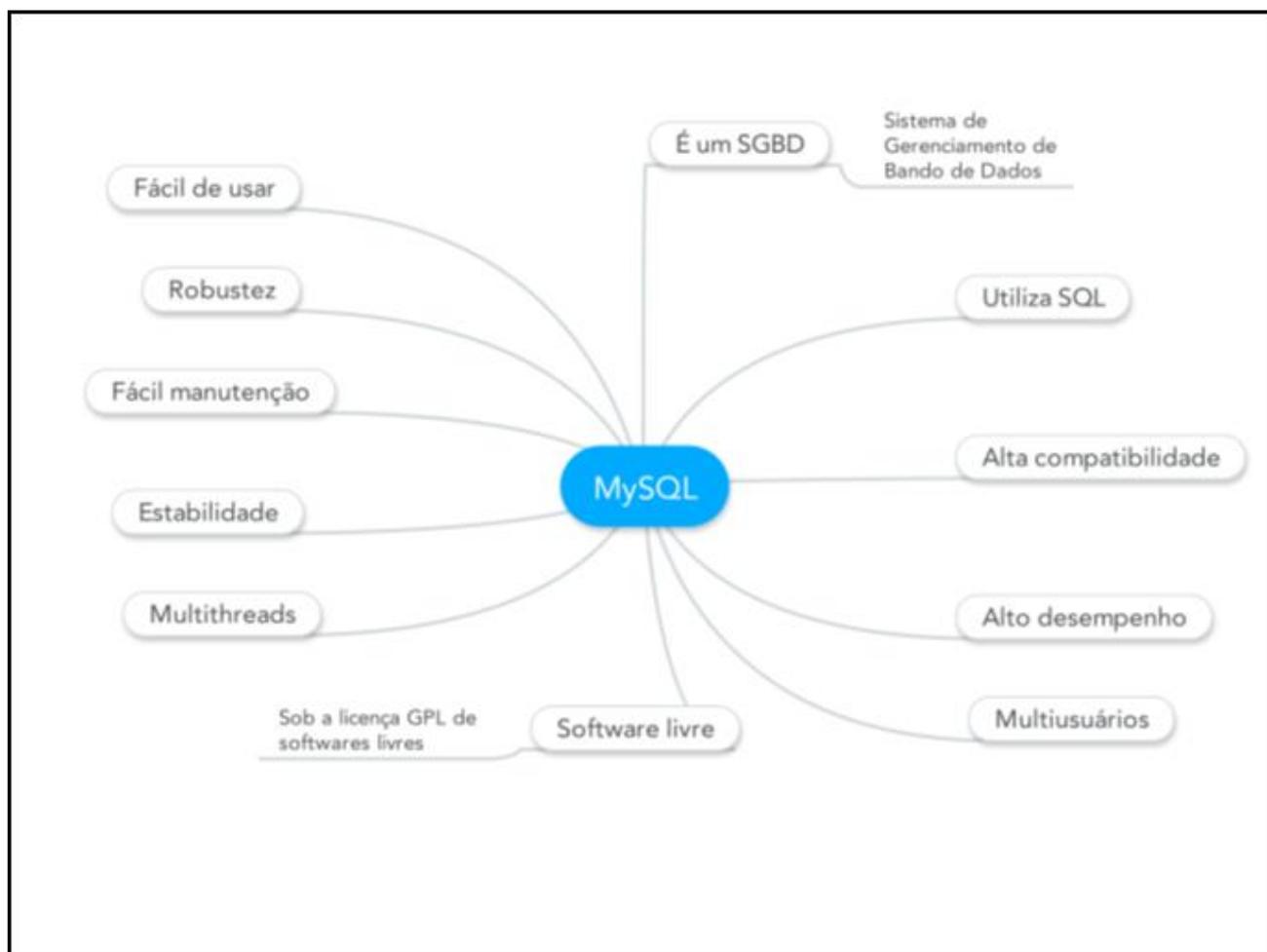


Figura 1 - MySQL – SGBD

2.1.7. Primefaces

PrimeFaces é uma bibliotecas de componentes visuais para desenvolvimento Web com Java. A figura 2 ilustra o *showcase* do *primefaces*. A suíte de componentes inclui diversos campos de entrada, botões, tabelas de dados, árvores, gráficos, diálogos, entre outros. (ALGAWORKS SOFTWARES, 2015).

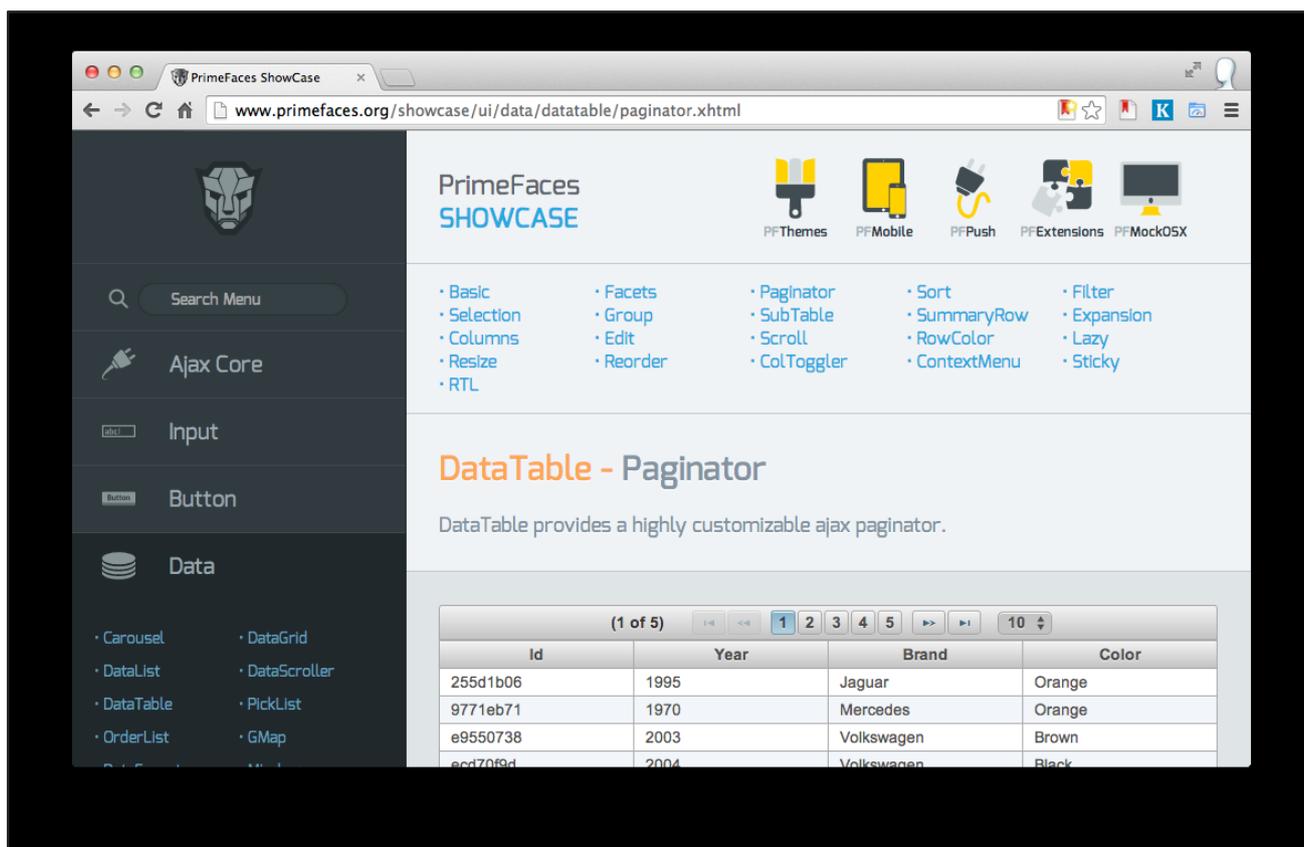


Figura 2 - PrimeFaces

O *PrimeFaces* é um popular *framework* para projetos JSF que pode ser usado para desenvolver rapidamente aplicações sofisticadas para empresas ou no desenvolvimento de sites padrão. Tecnicamente, o *PrimeFaces* é uma biblioteca de componentes de interface gráfica para as aplicações web baseadas em JSF. Com ele é possível aumentar a produtividade do desenvolvedor e a experiência do usuário com a aplicação, pois torna menos árduo criar uma aplicação que seja exibida corretamente na maioria dos dispositivos, sem contar que é muito flexível e personalizável, com uma grande opção de componentes para os mais diversos fins.

Além deste, temos outros *frameworks* para a mesma finalidade, dentre eles o *IceFaces* e o *RichFaces*, porém com menos popularidade, sem contar que o *PrimeFaces* está em alta comparado aos seus concorrentes que estão cada vez mais perdendo espaço. É possível comprovar isso com uma simples busca no Google *Trends*, como mostra a Figura 3.



Figura 3 - Popularidade dos frameworks através do Google Trends

3. PLANEJAMENTO DO PROJETO

3.1. WBS (WORK BREAKDOWN STRUCTURE /ESTRUTURA ANALÍTICA DE TRABALHO - EAP)

A WBS ou estrutura analítica de trabalho é uma subdivisão hierárquica do trabalho do projeto em partes menores, mais facilmente gerenciáveis. Seu objetivo primário é organizar o que deve ser feito para produzir as entregas do projeto.

A figura 4 ilustra a estrutura analítica de trabalho.

A EAP garante ao gerente de projetos a visibilidade das principais entregas, facilitando o controle de tempo e de custo. Ela faz parte do processo de gerenciamento de escopo do projeto, descrito no Guia PMBOK® (*Project Management Body of Knowledge*), uma das principais referências em gestão de projetos do mundo (EUAX Consulting – 2022).

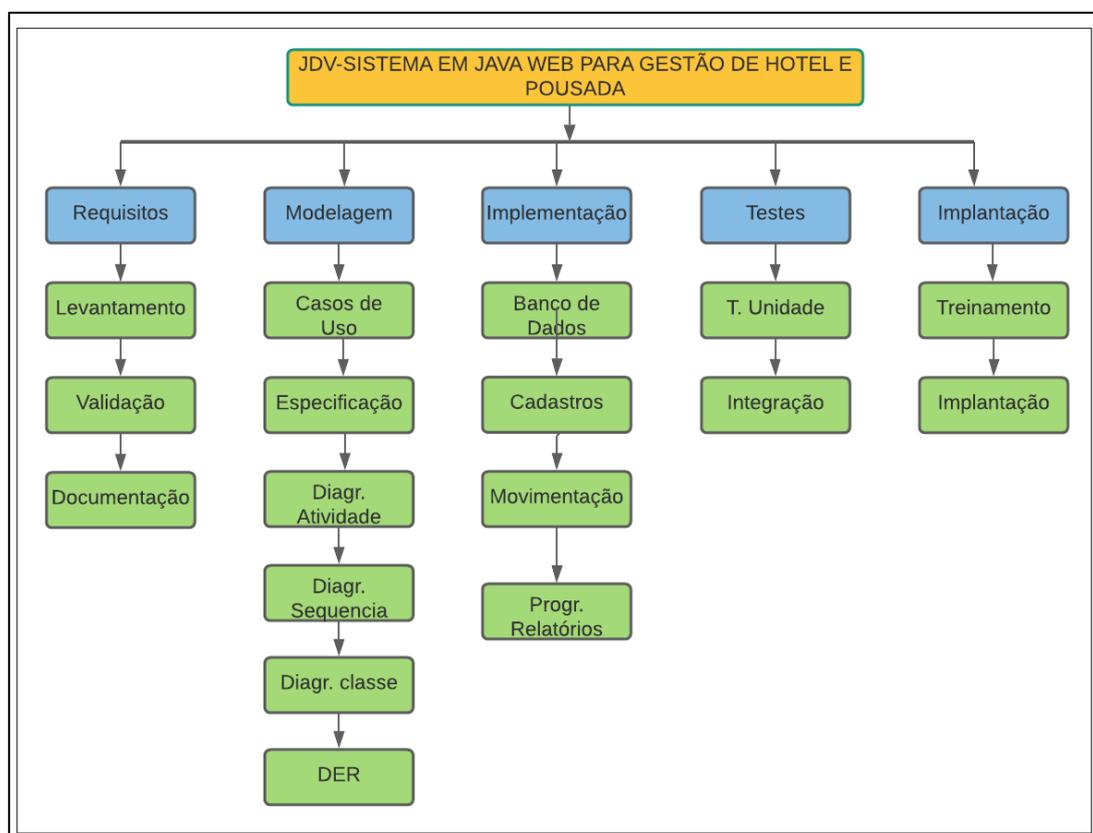


Figura 4 – WBS

3.2. SEQUENCIAMENTO DE ATIVIDADES

O processo de sequenciar as atividades, exibida na figura 5, consiste em identificar a sequência que as atividades serão realizadas e faz parte do Gerenciamento do tempo do projeto, um dos processos que compõe o Guia PMBOK, um guia do conhecimento em gerenciamento de projetos.

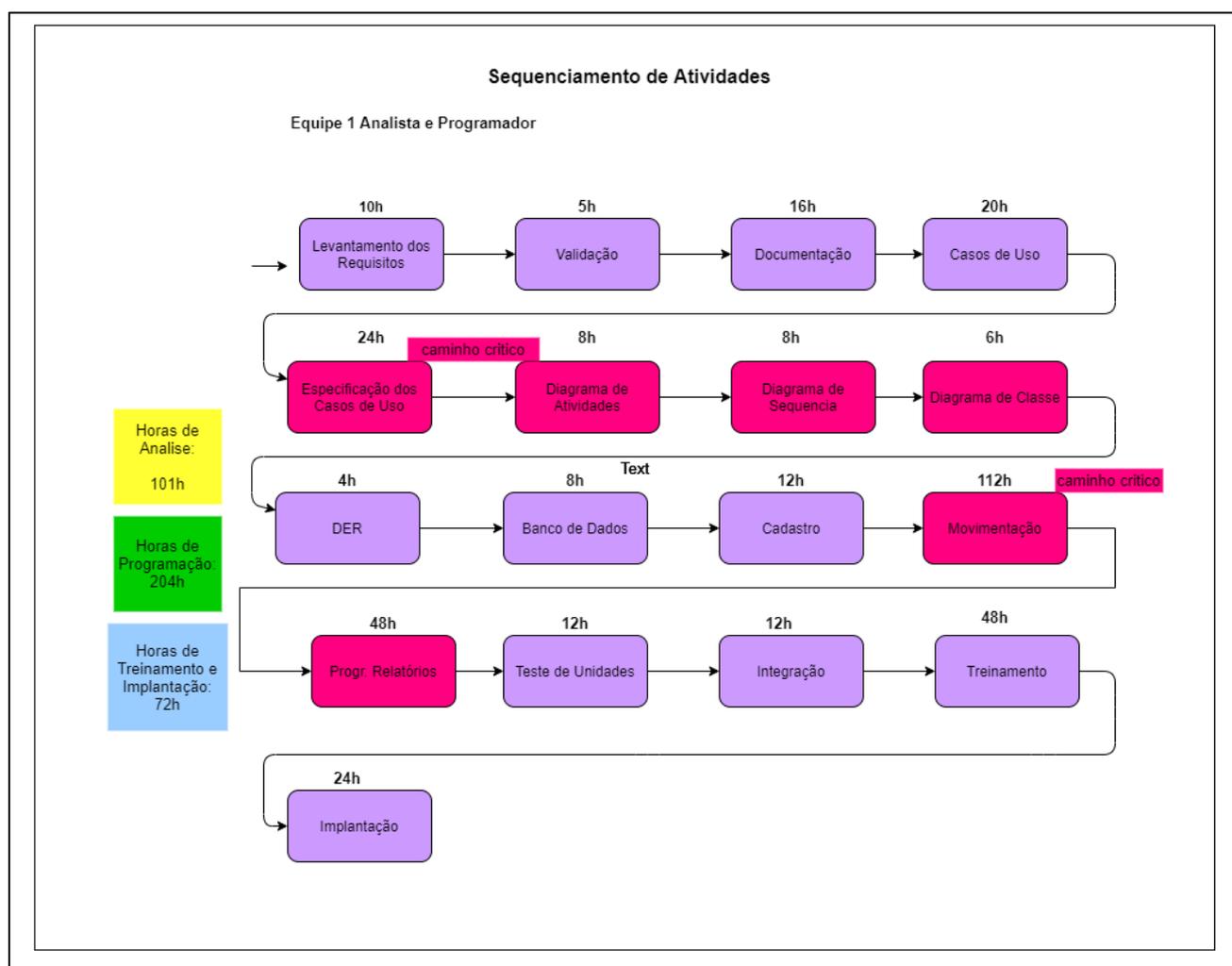


Figura 5 - Sequenciamento de Atividades

4. ANÁLISE E ESPECIFICAÇÃO DO TRABALHO

Para realizar a análise e especificação do software para gestão de hotel e pousada é necessário seguir algumas etapas. Identificar as necessidades e requisitos do cliente, definir o escopo do software, identificar os atores e casos de uso, construir o diagrama de classes e diagrama de entidade relacionamentos e outros que necessitar.

4.1. MAPA MENTAL

Um mapa mental é uma técnica de organização de ideias, que utiliza uma representação gráfica e visual para facilitar a compreensão e a memorização de conceitos, tarefas ou informações.

O mapa mental é uma ferramenta útil para organizar ideias, planejar projetos, estudar, criar apresentações, entre outras finalidades, e pode ser utilizado tanto individualmente quanto em grupo. É uma técnica flexível e adaptável, que permite a personalização de acordo com as necessidades e preferências de cada usuário. A figura 6 exibe o mapa mental do sistema.

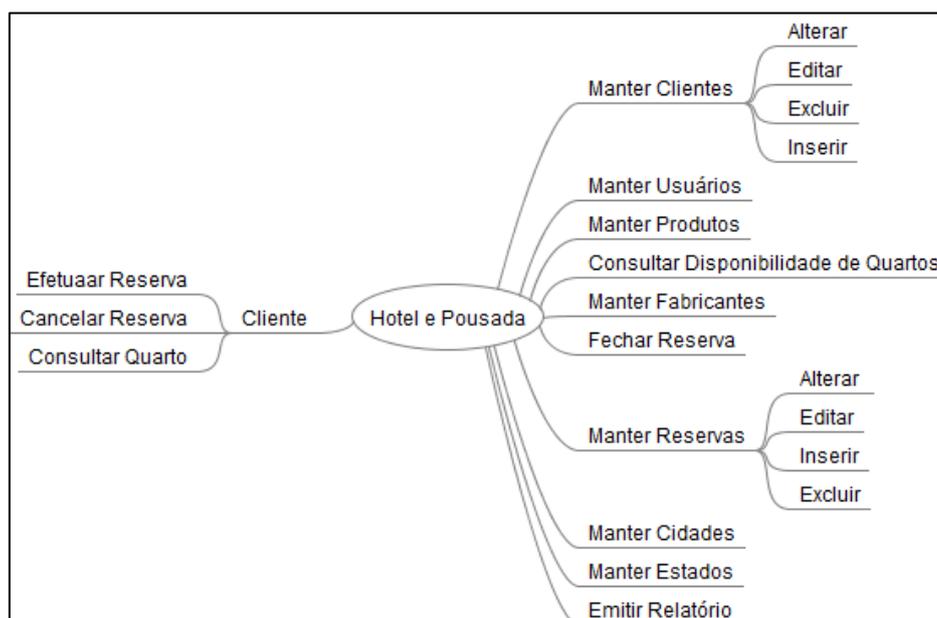


Figura 6 - Mapa Mental

4.2. MAPEAMENTO DE REQUISITOS

Por ser uma das partes mais importantes no desenvolvimento do sistema, é necessário realizar entrevistas com o cliente, usuários e pessoas que irão fazer uso do software, para o levantamento de requisitos do sistema.

Após realizar entrevista com o cliente usuários foram obtidos os seguintes requisitos do sistema:

- ✓ Manter hóspedes;
- ✓ Realizar *check-in*;
- ✓ Realizar *check-out*;
- ✓ Efetuar reserva;
- ✓ Cancelar reserva;
- ✓ Consultar hóspede;
- ✓ Manter apartamentos;
- ✓ Consultar reserva;
- ✓ Manter produtos;
- ✓ Manter usuários;
- ✓ Manter apartamento;
- ✓ Emitir relatórios de cliente;
- ✓ Emitir relatório de consumo;
- ✓ Emitir relatórios de funcionários.

4.3. DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO

O Diagrama de Casos de Uso é o diagrama utilizado normalmente nas fases de Levantamento e Análise de Requisitos do sistema. Esse diagrama apresenta uma linguagem simples e de fácil compreensão. Para que os usuários possam ter uma ideia inicial de como será o sistema, apresenta-se o diagrama de caso de Uso geral, conforme ilustrado pela Figura 8 (GUEDES, GILLEANES T. A. 2006). Com base nas informações fornecidas até o momento, é possível desenhar um diagrama de casos de uso para um sistema de gerenciamento de hotéis e pousadas. A Figura 7 é uma representação do sistema:

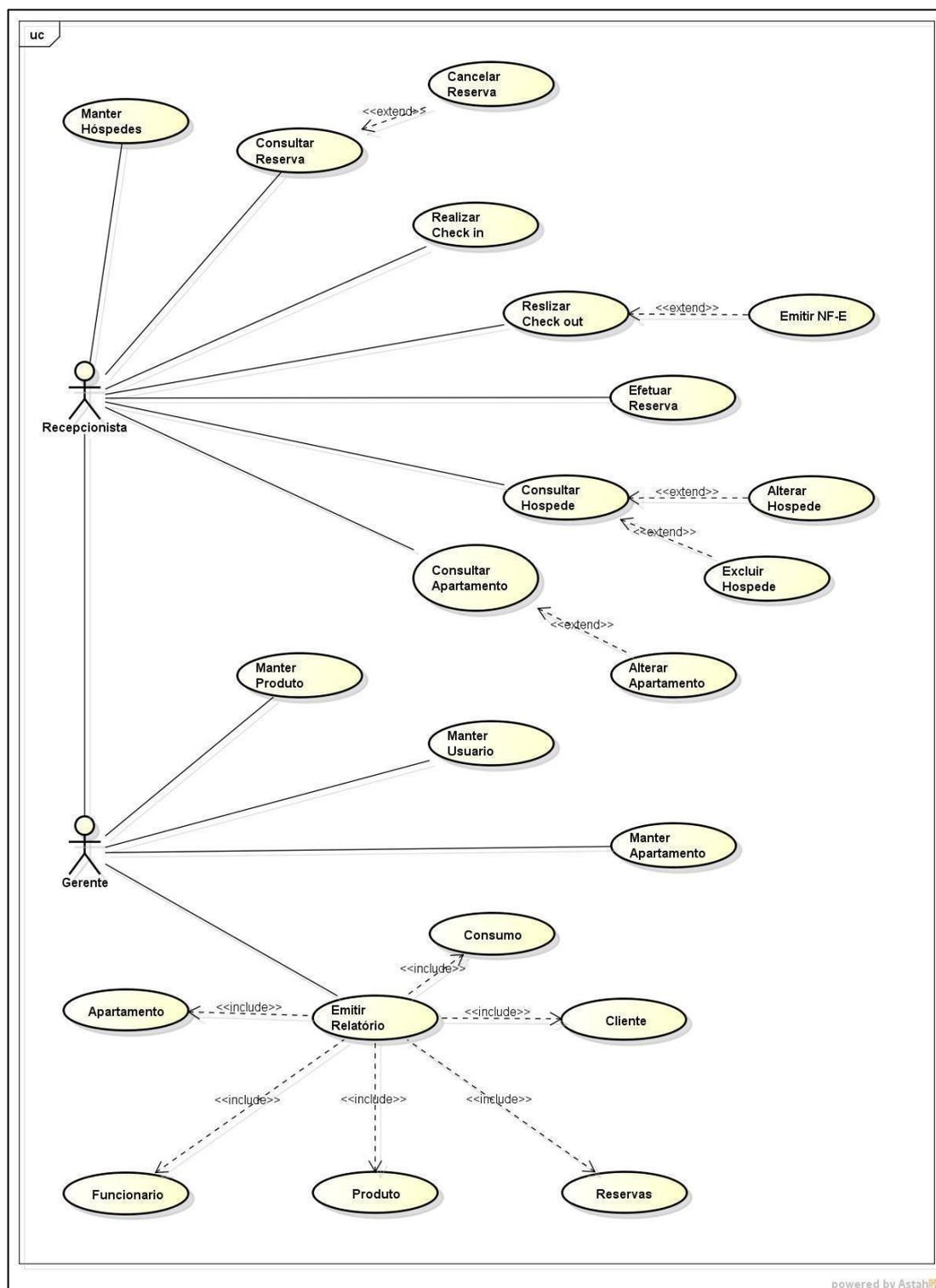
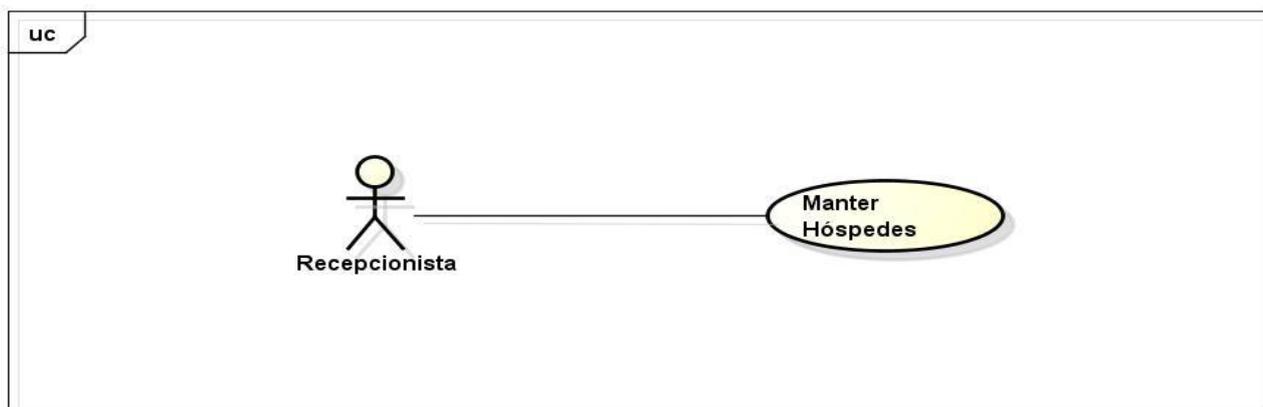


Figura 7 - Caso de Uso Geral

O diagrama da Figura 7 representa os atores envolvidos no sistema e os casos de uso que eles executam. Os atores são o recepcionista e o administrador. Para cada caso de uso, é possível especificar os fluxos básicos e alternativos.



powered by Astah

Figura 8 - Caso de Uso Manter Hóspedes

Caso de Uso Manter Hospedes
<p>1. Finalidade:</p> <p>Realizar o cadastro do hóspede;</p>
<p>2. Atores:</p> <p>Recepcionista;</p>
<p>3. Pré – Condições:</p> <p>O recepcionista deve estar autenticado no sistema;</p>
<p>4. Evento Inicial:</p> <p>O recepcionista seleciona a opção de cadastrar hóspede na página de entrada do sistema.</p>
<p>5. Fluxo Principal:</p> <p>a) O recepcionista seleciona a opção de cadastro;</p> <p>b) O sistema solicita os dados necessários para o cadastro do cliente (nome, endereço, RG, CPF, telefone, e-mail), (A1);</p>

c) O recepcionista verifica se digitou os dados corretamente e confirma o cadastro, (E1);

d) O caso de uso é encerrado;

6. Fluxo Alternativo:

A1. Cancelar operação.

a) O recepcionista cancela a opção de cadastro.

b) O sistema retorna ao passo 5.a);

7. Fluxo de Exceção:

E1. Hóspede já cadastrado.

a) Ocorre redundância de dados, ou seja, o hóspede já está cadastrado no sistema;

b) O sistema retorna ao passo 5.b);

8. Pós-condições:

O hóspede possui o cadastro no sistema, e poderá realizar reserva;

9. Casos de Teste:

Opção de cancelar o cadastro do hóspede.

Tabela 1 – Caso de Uso Manter Hospedes

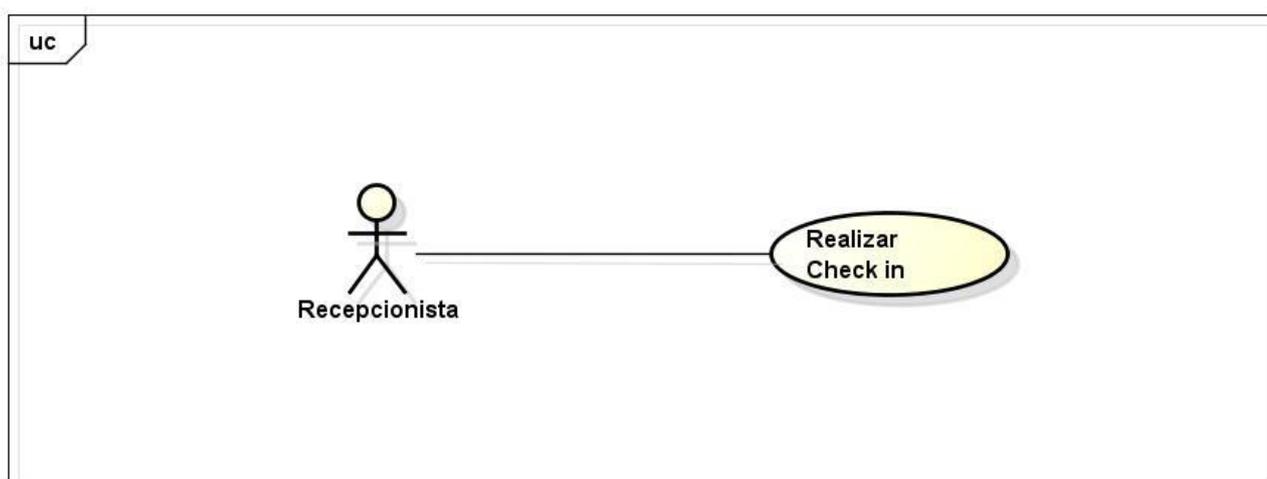


Figura 9 - Caso de Uso Realizar Check-in

Caso de Uso: Realizar Check in**1. Finalidade:**

Realizar check in de um novo hóspede;

2. Atores:

Recepcionista.

3. Pré – Condições:

O recepcionista deve realizar sua autenticação no sistema.

4. Evento Inicial:

O recepcionista seleciona a opção de “Realizar Check in”, na página inicial do sistema;

5. Fluxo Principal:

- a) O sistema solicita os dados para consulta do hóspede;
- b) O recepcionista informa os dados e confirma (A1);
- c) O recepcionista verifica os dados do cliente e solicita a opção de “Selecionar Apartamento” (E1);
- d) O recepcionista seleciona o apartamento, e solicita o preenchimento das informações referentes à locação;
- e) O recepcionista insere a data de início da estadia e do seu fim;
- f) O recepcionista insere a opção de pagamento (Diária Livre, ou promocional);
- g) O recepcionista solicita a opção de “Imprimir Contrato”;
- h) Os dados de check-in são armazenados no sistema;
- i) O recepcionista indica o apartamento;
- j) O caso de uso é encerrado;

6. Fluxo Alternativo:

A1. Cancelar operação

- a) O recepcionista cancela a operação de consulta;
- b) O sistema retorna ao passo 5.a);

7. Fluxo de Exceção:

E1. Hóspede não cadastrado.

- a) O sistema realiza a consulta e verifica que o cliente não está cadastrado.
- b) O sistema oferece a opção de “Novo Cadastro” do hóspede;
- c) O sistema retorna ao passo 5.a);

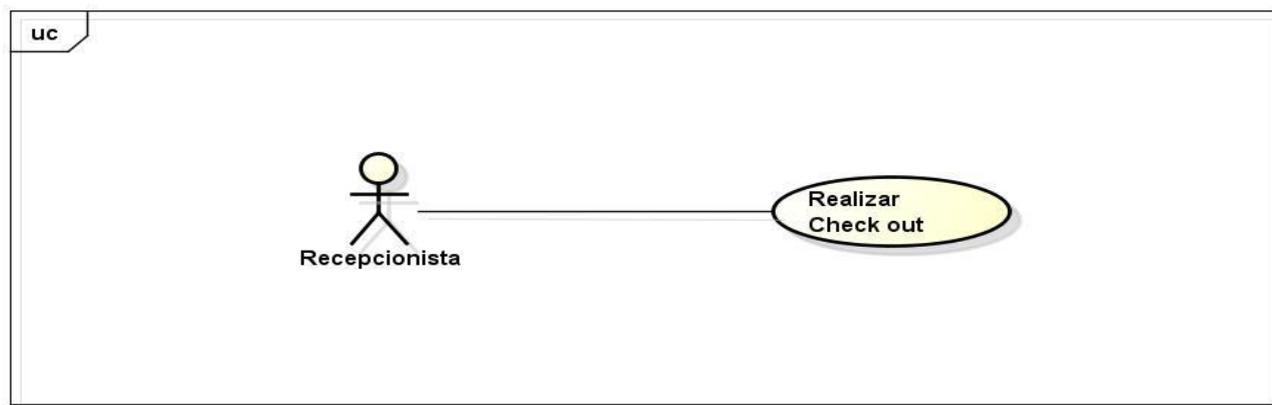
8. Pós-condições:

Apartamento é liberado ao hóspede;

9. Casos de Teste:

Verificar disponibilidade de cada apartamento (Status);

Tabela 2 - Caso de Uso Realizar Check-in



powered by Astah

Figura 10 - Caso de Uso Realizar *Check-out***Caso de Uso: Realizar Check-out****1. Finalidade:**Realizar *check-out* do hóspede;**2. Atores:**

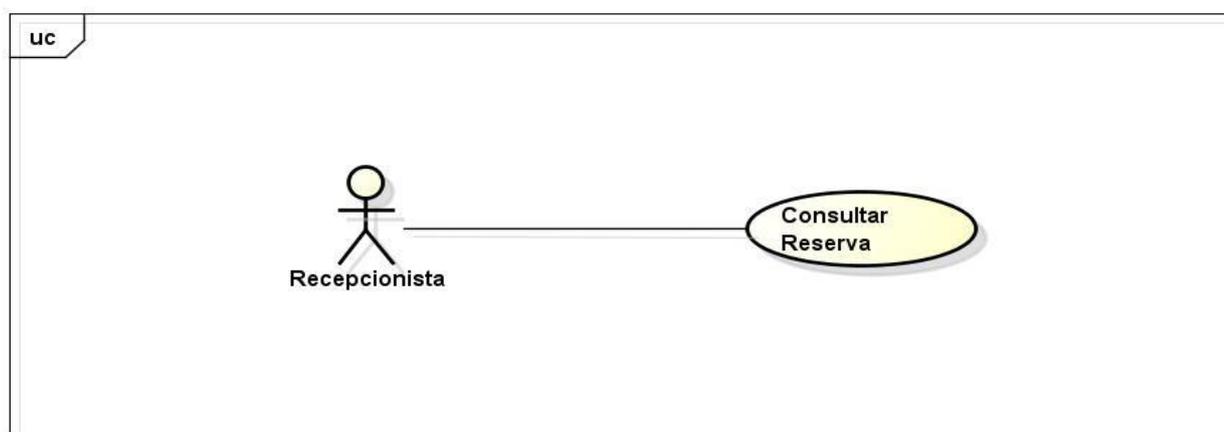
Recepcionista.

3. Pré-Condições:

O recepcionista deve realizar sua autenticação no sistema.

<p>4. Evento Inicial:</p> <p>O recepcionista seleciona a opção de “Realizar <i>Check out</i>”, na página inicial do sistema;</p>
<p>5. Fluxo Principal:</p> <p>a) O sistema solicita os dados para consulta do hóspede;</p> <p>b) O recepcionista informa os dados e confirma (A1);</p> <p>c) O recepcionista verifica os dados do cliente e solicita a opção de “Finalizar Estadia” (E1);</p> <p>d) O recepcionista insere a opção de pagamento (Diária Livre, ou promocional);</p> <p>e) O recepcionista solicita a opção de “Imprimir NF-E”;</p> <p>f) Os dados de check out são armazenados no sistema;</p> <p>g) O caso de uso é encerrado;</p>
<p>6. Fluxo Alternativo:</p> <p>A1. Cancelar operação</p> <p>a) O recepcionista cancela a operação de check out;</p> <p>b) O sistema retorna ao passo 5.a);</p>
<p>7. Fluxo de Exceção:</p> <p>E1. Hóspede passou do limite de dias.</p> <p>a) O sistema realiza o cálculo e adiciona a soma final.</p> <p>b) O sistema retorna ao passo 5.a);</p>
<p>Pré-condições:</p> <p>Apartamento é bloqueado para limpeza;</p>
<p>Casos de Teste:</p> <p>Verificar o apartamento (Status);</p>

Tabela 3 - Caso de Uso Realizar Check-out



powered by Astah

Figura 11 - Caso de Uso Consultar Reserva

Caso de Uso: Consultar Reserva
<p>1. Finalidade:</p> <p>Habilitar a consulta de uma determinada reserva cadastrado no sistema;</p>
<p>2. Atores:</p> <p>Recepcionista.</p>
<p>3. Pré-Condições:</p> <p>O recepcionista deve realizar sua autenticação no sistema</p>
<p>4. Evento Inicial:</p> <p>O recepcionista seleciona a opção de “Reserva”, na página inicial do sistema.</p>
<p>5. Fluxo Principal:</p> <ul style="list-style-type: none">a) O recepcionista seleciona a opção de “Consultar”;b) O recepcionista insere o número do apartamento ou a data para consulta (A1);c) O sistema retorna uma lista de reservas (E1);d) O recepcionista seleciona uma reserva para obter informações mais detalhadas;e) O sistema apresenta todas as informações da reserva ao recepcionista (A2);f) O caso de uso é encerrado;
<p>6. Fluxo Alternativo:</p> <p>A1. Cancelar Operação</p> <ul style="list-style-type: none">a) O recepcionista cancela a operação de “Consultar”;b) O sistema retorna ao passo 5.a). <p>A2. Alterar Reserva</p> <ul style="list-style-type: none">a) O recepcionista seleciona a opção de “Alterar Reserva”;

- b) O sistema exibe os campos de dados preenchidos com as informações mais atualizadas da reserva;
- c) O recepcionista altera as informações necessárias e/ou adiciona novas observações a reserva;
- d) O recepcionista confirma a alteração através da opção “Salvar Dados”;
- e) O caso de uso é encerrado;

7. Fluxo de Exceção:

E1. Nenhum resultado encontrado

- a) O sistema retorna sem resultados de busca;
- b) O sistema informa a mensagem de nenhum resultado encontrado;
- c) O sistema retorna ao passo 5.a);

8. Pós-condições:

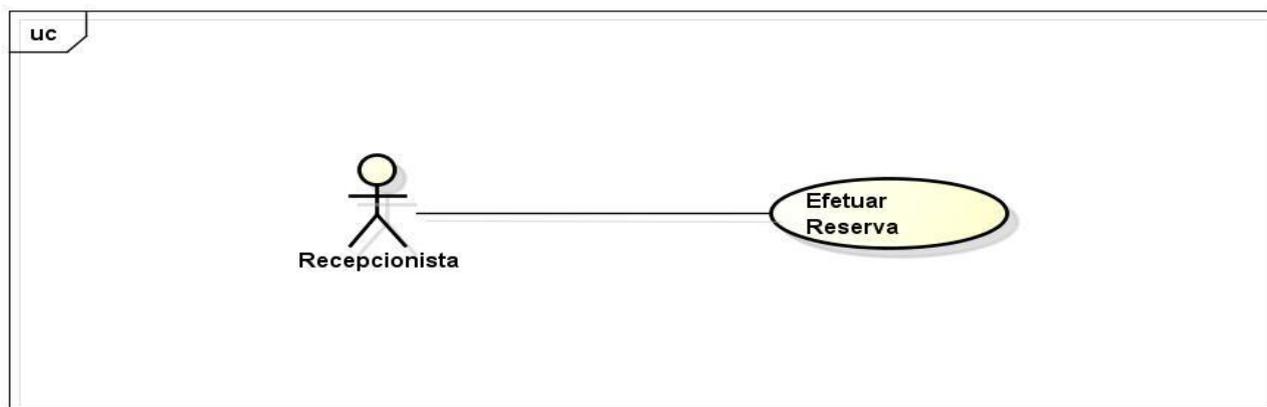
Informações detalhadas de determinada reserva cadastrado no sistema.

Atualização de dados da reserva;

9. Casos de Teste:

Definir data ou número do apartamento como parâmetros de consulta.

Tabela 4 - Caso de Uso Consultar Reserva



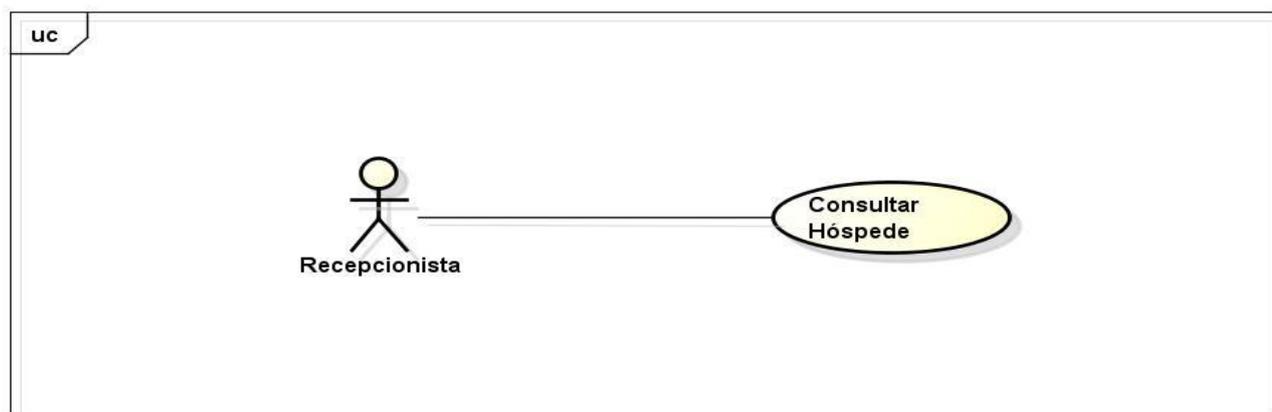
powered by Astah

Figura 12 - Caso de Uso Efetuar Reserva

Caso de Uso: Efetuar Reserva	
1. Finalidade:	Agendar uma nova reserva de apartamento;
2. Atores:	Recepcionista;
3. Pré – Condições:	O recepcionista deve realizar sua autenticação no sistema;
4. Evento Inicial:	O recepcionista seleciona a opção de “Realizar Reserva”, na página inicial;
5. Fluxo Principal:	<ul style="list-style-type: none"> a) O Sistema solicita os dados para consulta do hóspede; b) O recepcionista informa os dados e confirma (A1);

<p>c) O recepcionista verifica os dados do hóspede e solicita a opção de “Selecionar Apartamento” (E1);</p> <p>d) O recepcionista seleciona o apartamento, e solicita o preenchimento das informações referentes à reserva;</p> <p>e) O recepcionista insere a data de início da reserva e do seu fim;</p> <p>f) O recepcionista insere a opção de pagamento (Diária Livre, ou promocional);</p> <p>g) O recepcionista solicita a opção de “Imprimir Reserva”;</p> <p>h) Os dados da reserva do apartamento são armazenados;</p> <p>i) O recepcionista informa os dados da reserva ao hóspede;</p> <p>j) O caso de uso é encerrado;</p>
<p>6. Fluxo Alternativo:</p> <p>A1. Cancelar Operação</p> <p>a) O recepcionista cancela a operação de consulta;</p> <p>b) O Sistema retorna ao passo 5.a);</p>
<p>7. Fluxo de Exceção:</p> <p>E1. Hóspede não cadastrado.</p> <p>a) O sistema realiza a consulta e verifica que o hóspede não está cadastrado.</p> <p>b) O Sistema oferece a opção de “Novo Cadastro” ao recepcionista;</p> <p>c) O Sistema retorna ao passo 5.a);</p>
<p>8. Pós-condições:</p> <p>Apartamento é reservado ao hóspede;</p>
<p>9. Casos de Teste:</p> <p>Verificar disponibilidade de cada apartamento para determinada data (Status);</p>

Tabela 5 - Caso de Uso Efetuar Reserva



powered by Astah

Figura 13 - Caso de Uso Consultar Hóspede

Caso de Uso: Consultar Hóspede	
1.Finalidade:	Habilitar a consulta de um determinado hóspede no sistema;
2.Atores:	Recepcionista;
3.Pré – Condições:	O recepcionista deve realizar sua autenticação no sistema;
4.Evento Inicial:	O recepcionista seleciona a opção de “Hospede”, na Página inicial do Sistema;
5. Fluxo Principal:	a) O recepcionista seleciona a opção de “Consultar”; b) O recepcionista insere o CPF ou nome do hóspede para consulta (A1); c) O sistema retorna uma lista de hóspedes para o recepcionista (E1); d) O recepcionista seleciona o hóspede para obter informações mais detalhadas (A2), (A3); e) O sistema apresenta todas as informações do hóspede cadastrado; f) O caso de uso é encerrado;

6. Fluxo Alternativo: A1. Cancelar Operação a) O recepcionista cancela a operação de “Consultar” b) O sistema retorna ao passo 5.a). A2. Alterar Hóspede a) O recepcionista seleciona a opção de “Alterar Hóspede”; b) O sistema exibe os campos de dados preenchidos com os dados mais recentes do hóspede; c) O recepcionista altera as informações do hóspede necessárias; d) O recepcionista confirma a alteração através da opção “Salvar Dados”; e) O caso de uso é encerrado; A3. Excluir Hóspede a) O recepcionista opta pela opção de “Excluir Cadastro”; b) O Sistema solicita a confirmação da exclusão (A4); c) O recepcionista confirma a exclusão; d) O Sistema retorna uma mensagem de Exclusão; e) O Caso de Uso é encerrado; A4. Cancelar Operação de Exclusão; a) O Recepcionista cancela a operação de “Excluir Hóspede”; b) O Caso de Uso é encerrado;

7. Fluxo de Exceção:

E1. Nenhum resultado encontrado

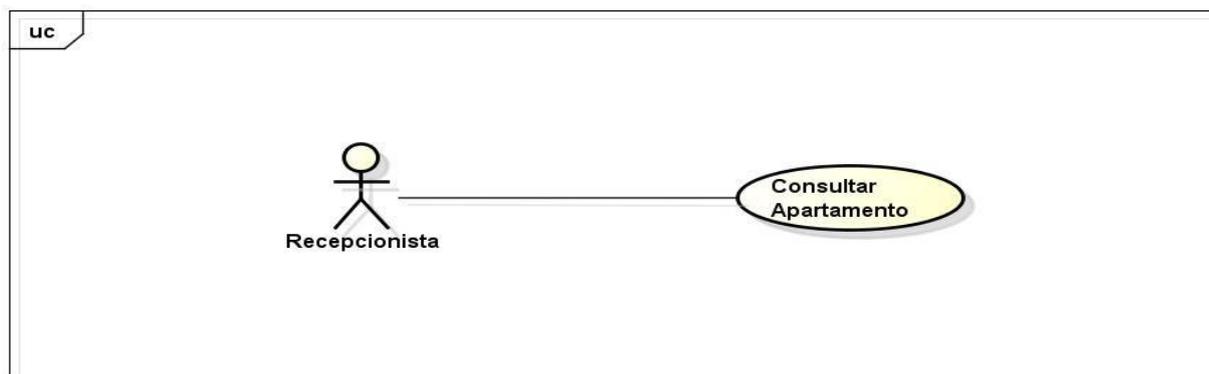
- a) O Sistema retorna sem resultados de busca;
- b) O Sistema informa a mensagem de nenhum resultado encontrado;
- c) O Sistema retorna ao passo 5.b);

8. Pós-condições:

Informações detalhadas de determinado Hóspede cadastrado no sistema;

9. Casos de Teste: Definir nome e CPF do hóspede como campos de pesquisa;

Tabela 6 - Caso de Uso Consultar Hospede



powered by Astah

Figura 14 – Caso de Uso Consultar Apartamento

Caso de Uso: Consultar Apartamento
<p>1. Finalidade:</p> <p>Habilitar a consulta de um determinado apartamento cadastrado no sistema;</p>
<p>2. Atores:</p> <p>Recepcionista;</p>
<p>3. Pré – Condições:</p> <p>O Recepcionista deve realizar sua autenticação no sistema.</p>
<p>4. Evento Inicial:</p> <p>O Recepcionista seleciona a opção de “Apartamento”;</p>
<p>5. Fluxo Principal:</p> <p>a) O Recepcionista seleciona a opção de “Consultar”;</p> <p>b) O Recepcionista insere número do apartamento para consulta (A1);</p>

- c) O sistema retorna uma lista de apartamentos (E1);
- d) O Recepcionista seleciona um apartamento para obter informações mais detalhadas;
- e) O sistema apresenta todas as informações do apartamento ao recepcionista (A2);
- f) O caso de uso é encerrado;

6. Fluxo Alternativo:

A1. Cancelar Operação

- a) O recepcionista cancela a operação de “Consultar”;
- b) O sistema retorna ao passo 5.a).

A2. Alterar Apartamento

- a) O recepcionista seleciona a opção de “Alterar Apartamento”;
- b) O sistema exibe os campos de dados preenchidos com as informações mais atualizadas do apartamento;
- c) O recepcionista altera as informações necessárias e/ou adiciona novas observações ao apartamento;
- d) O recepcionista confirma a alteração através da opção “Salvar Dados”;
- e) O Caso de uso é encerrado;

7. Fluxo de Exceção:

E1. Nenhum resultado encontrado

- a) O Sistema retorna sem resultados de busca;
- b) O Sistema informa a mensagem de nenhum resultado encontrado;
- c) O sistema retorna ao passo 5.a);

8. Pós-condições:

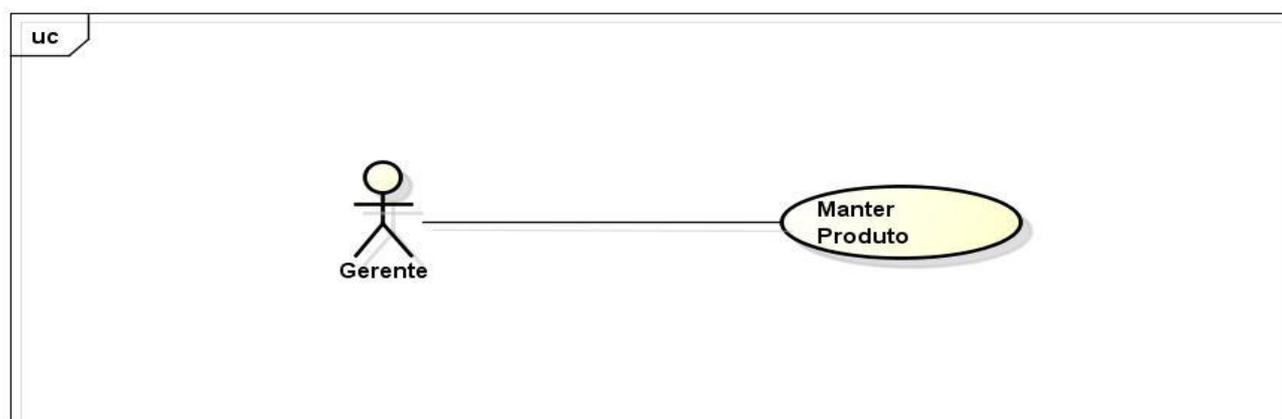
Informações detalhadas de determinado apartamento cadastrado no sistema.

Atualização de dados do apartamento;

9. Casos de Teste:

Definir número do apartamento como parâmetro de consulta.

Tabela 7 - Caso de Uso Consultar Apartamento



powered by Astah

Figura 15 - Caso de Uso Manter Produto

Caso de Uso: Manter Produto**1. Finalidade:**

Permite ao gerente: incluir, alterar e excluir produtos;

2. Atores:

Gerente;

3. Pré – Condições:

O gerente deve estar autenticado no sistema;

4. Evento Inicial:

O gerente seleciona a opção de cadastrar produto.

5. Fluxo Principal:

- a) O gerente seleciona uma opção: 1. Cadastrar; 2. Alterar; 3. Excluir;
- b) Se o gerente selecionar a opção 1; (A1);
- c) Se o gerente selecionar a opção 2; (A2);
- d) Se o gerente selecionar a opção 3; (A3);
- e) O gerente verifica se os dados estão corretos e confirma;
- f) O caso de uso é encerrado;

6. Fluxo Alternativo:**A1. Cadastrar.**

- a) O sistema solicita os dados do novo produto;
- b) O gerente informa os dados. (E1);
- c) O sistema retorna ao passo 5.a);

A2. Alterar.

- a) O sistema informa os dados cadastrados;

b) O gerente seleciona os dados a serem alterados;

c) O gerente altera os dados.

d) O sistema retorna ao passo 5.b);

A3. Excluir.

a) O sistema informa os dados cadastrados;

b) O gerente seleciona os dados a serem excluídos.;

c) O gerente exclui os dados;

d) O sistema retorna ao passo 5.c);

7. Fluxo de Exceção:

E1. Produto já cadastrado;

a) Ocorre redundância de dados, ou seja, o produto já está cadastrado no sistema;

b) O sistema retorna ao passo 5.b);

Tabela 8 - Caso de Uso Manter Produto

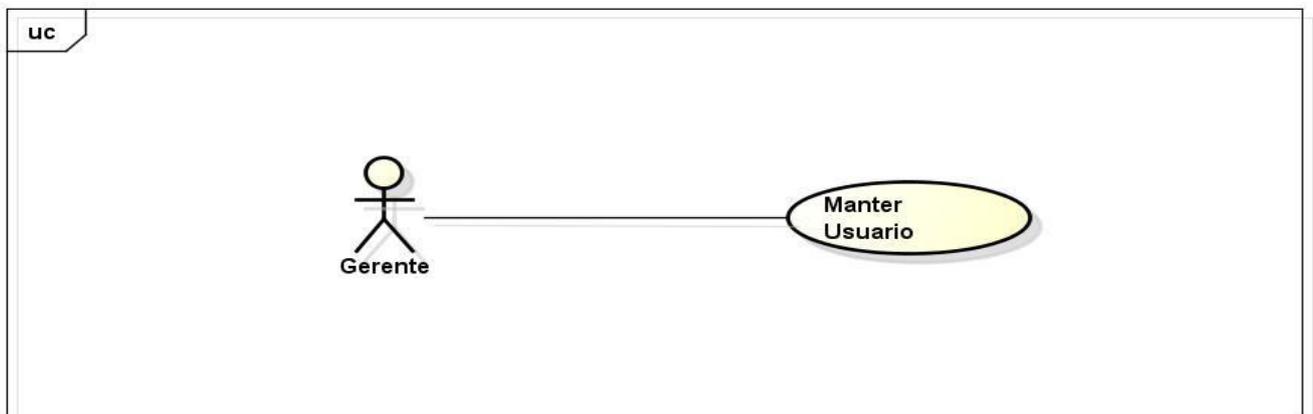


Figura 16 - Caso de Uso Manter Usuário

Caso de Uso: Manter Usuário**1. Finalidade:**

Permite ao gerente: incluir, alterar e excluir usuários;

2. Atores:

Gerente;

3. Pré – Condições:

O gerente deve estar autenticado no sistema;

4. Evento Inicial:

O gerente seleciona a opção de cadastrar usuário.

5. Fluxo Principal:

- a) O gerente seleciona uma opção: 1. Cadastrar; 2. Alterar; 3. Excluir;
- b) Se o gerente selecionar a opção 1; (A1);
- c) Se o gerente selecionar a opção 2; (A2);
- d) Se o gerente selecionar a opção 3; (A3);
- e) O gerente verifica se os dados estão corretos e confirma;
- f) O caso de uso é encerrado;

6. Fluxo Alternativo:**A1. Cadastrar;**

- a) O sistema solicita os dados do novo usuário.
- b) O gerente informa os dados. (E1)
- c) O sistema retorna ao passo 5.a);

A2. Alterar;

- a) O sistema informa os dados cadastrados.

b) O gerente seleciona os dados a serem alterados.

c) O gerente altera os dados.

d) O sistema retorna ao passo 5.b);

A3. Excluir;

a) O sistema informa os dados cadastrados.

b) O gerente seleciona os dados a serem excluídos.

c) O gerente exclui os dados.

d) O sistema retorna ao passo 5.c);

7. Fluxo de Exceção:

E1. Usuário já cadastrado;

a) Ocorre redundância de dados, ou seja, o usuário já está cadastrado no sistema;

b) O sistema retorna ao passo 5.b);

Tabela 9 - Caso de Uso Manter Usuário

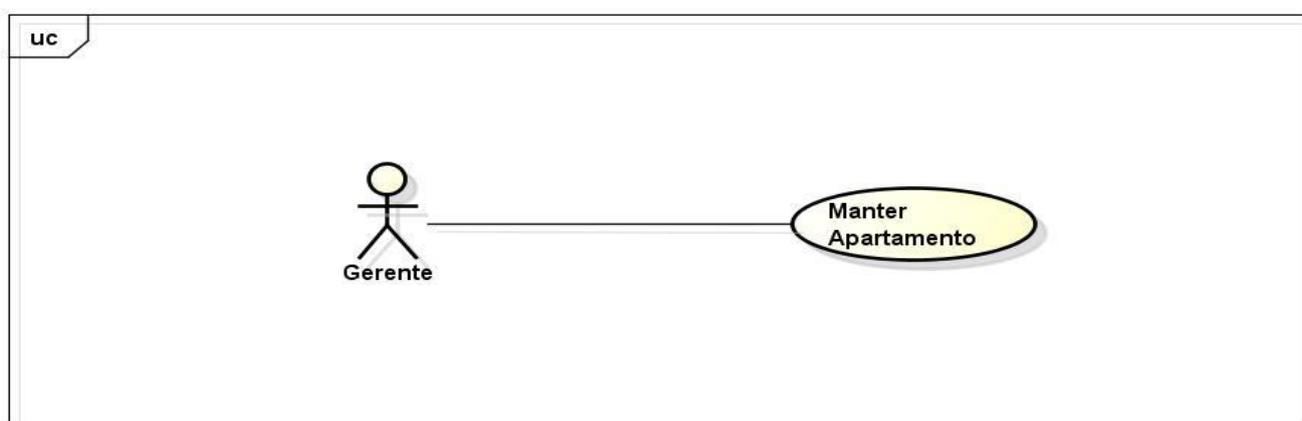


Figura 17 - Caso de Uso Manter Apartamento

Caso de Uso: Manter Apartamento**1. Finalidade:**

Permite ao gerente: incluir, alterar e excluir apartamentos;

2. Atores:

Gerente;

3. Pré – Condições:

O gerente deve estar autenticado no sistema;

4. Evento Inicial:

O gerente seleciona a opção de cadastrar apartamento;

5. Fluxo Principal:

- a) O gerente seleciona uma opção: 1. Cadastrar; 2. Alterar; 3. Excluir;
- b) Se o gerente selecionar a opção 1; (A1);
- c) Se o gerente selecionar a opção 2; (A2);
- d) Se o gerente selecionar a opção 3; (A3);
- e) O gerente verifica se os dados estão corretos e confirma;
- f) O caso de uso é encerrado;

6. Fluxo Alternativo:**A1. Cadastrar.**

- a) O sistema solicita os dados do novo apartamento.
- b) O gerente informa os dados. (E1)
- c) O sistema retorna ao passo 5.a);

A2. Alterar.

- a) O sistema informa os dados cadastrados.

b) O gerente seleciona os dados a serem alterados.

c) O gerente altera os dados.

d) O sistema retorna ao passo 5.b);

A3. Excluir.

a) O sistema informa os dados cadastrados.

b) O gerente seleciona os dados a serem excluídos.

c) O gerente exclui os dados.

d) O sistema retorna ao passo 5.c);

7. Fluxo de Exceção:

E1. Apartamento já cadastrado.

a) Ocorre redundância de dados, ou seja, o apartamento já está cadastrado no sistema;

b) O sistema retorna ao passo 5.b);

Tabela 10 - Caso de Uso Manter Apartamento

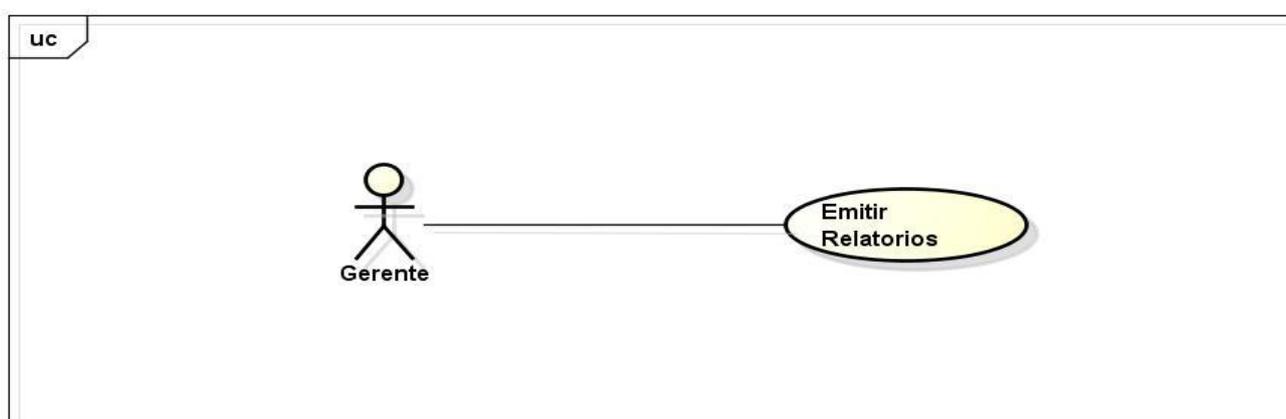


Figura 18 - Caso de Uso Emitir Relatórios

Caso de Uso: Emitir Relatório
<p>1. Finalidade:</p> <p>Permite ao gerente imprimir relatórios de consumo, cliente, reservas, produtos, funcionários e apartamentos;</p>
<p>2. Atores:</p> <p>Gerente;</p>
<p>3. Pré – Condições:</p> <p>O gerente deve estar autenticado no sistema;</p>
<p>4. Evento Inicial:</p> <p>O gerente seleciona a opção de emitir relatório;</p>
<p>5. Fluxo Principal:</p> <p>a) O gerente seleciona uma opção: 1. Consumo; 2. Cliente; 3. Reservas; 4. Produtos; 5. Funcionários; 6. Apartamentos;</p> <p>b) Se o gerente selecionar a opção 1; (A1);</p> <p>c) Se o gerente selecionar a opção 2; (A2);</p> <p>d) Se o gerente selecionar a opção 3; (A3);</p> <p>e) Se o gerente selecionar a opção 4; (A4);</p> <p>f) Se o gerente selecionar a opção 5; (A5);</p> <p>g) Se o gerente selecionar a opção 6; (A6);</p> <p>h) O gerente verifica se os dados estão corretos e confirma;</p> <p>i) O caso de uso é encerrado;</p>
<p>6. Fluxo Alternativo:</p>

A1. Consumo.

- a) O sistema habilita a opção de impressão.
- b) O gerente clica no botão de imprimir.
- c) O sistema retorna ao passo 5.a);

A2. Cliente.

- a) O sistema habilita a opção de impressão.
- b) O gerente clica no botão de imprimir.

A3. Reservas.

- a) O sistema habilita a opção de impressão.
- b) O gerente clica no botão de imprimir.
- c) O sistema retorna ao passo 5.a);

A4. Produtos.

- a) O sistema habilita a opção de impressão.
- b) O gerente clica no botão de imprimir.
- c) O sistema retorna ao passo 5.a);

A5. Funcionários.

- a) O sistema habilita a opção de impressão.
- b) O gerente clica no botão de imprimir.
- c) O sistema retorna ao passo 5.a);

A6. Apartamentos.

- a) O sistema habilita a opção de impressão.
- b) O gerente clica no botão de imprimir.
- c) O sistema retorna ao passo 5.a);

Tabela 11 - Caso de Uso Emitir Relatórios

4.4. DIAGRAMA DE CLASSES

Na UML, o diagrama de classes é uma das ferramentas mais utilizadas e importante. Com o objetivo de definir as estruturas das classes do sistema, este serve de apoio para a maioria dos demais diagramas. É com o diagrama de classes que mostramos os atributos e métodos que cada classe possui e ainda como as classes se relacionam e trocam informações entre si (Guedes, Gilleanes 2006).

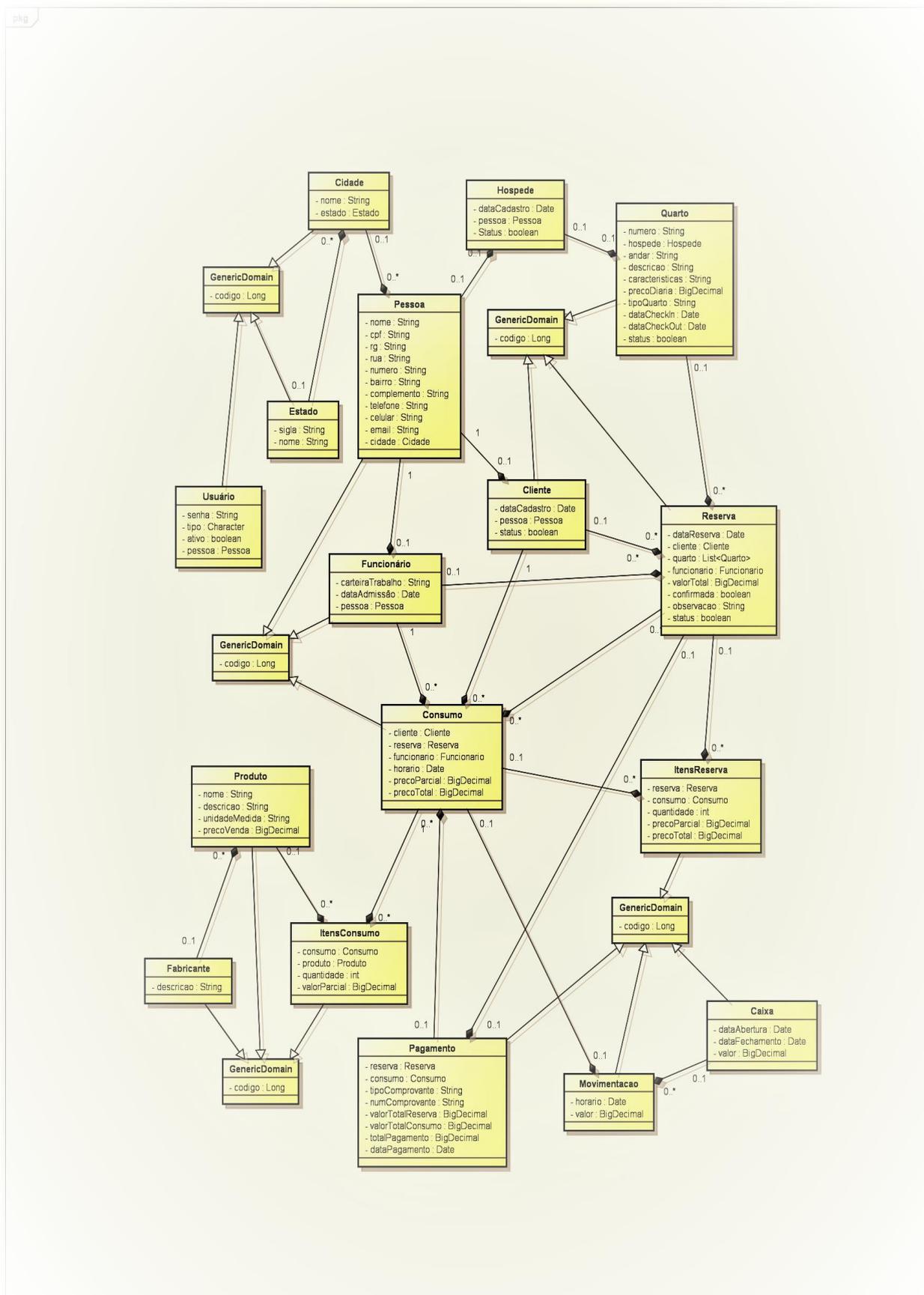


Figura 19 - Diagrama de Classe

No diagrama representado pela Figura 19, estão todas as classes do sistema: Pessoa, Usuário, Cliente, Funcionário, Quarto, Hóspede, Reserva, Pagamento, Consumo, Produto.

A classe quarto representa os quartos disponíveis nos estabelecimentos, com informações como número, tipo, preço diário, disponibilidade.

A classe Cliente representa os clientes, com informações como nome, CPF, telefone, e-mail, entre outros e uma lista de suas reservas.

Temos também a classe Reserva, que associa um hóspede a um quarto em um determinado período de tempo, e guarda informações como as datas de *check-in* e *check-out*, o valor total da reserva, e outras informações pertinentes.

Esse é um exemplo de como o diagrama de classe para o sistema de gestão de hotel e pousada em Java pode ser modelado. Dependendo dos requisitos específicos do sistema, é possível que outras classes sejam necessárias.

4.5. DIAGRAMA DE ENTIDADE-RELACIONAMENTO (DER)

Os conceitos de modelagem do modelo Entidade-Relacionamento (ER), que é um modelo de dados conceitual de alto nível, além de muito popular. Esse modelo e suas variações são normalmente empregados para o projeto conceitual de aplicações de um banco de dados, e muitas ferramentas de projeto de um banco de dados aplicam seus conceitos. Descreveremos os conceitos da estruturação de dados básica e as restrições do modelo ER.

As metodologias de modelagem de objetos como UML (*Universal Modeling Language* — Linguagem de Modelagem Universal) estão se tornando cada vez mais populares no projeto e engenharia de software. Essas metodologias vão além do projeto de um banco de dados, especificando o projeto detalhado dos módulos de software e suas interações, utilizando vários tipos de diagramas. Uma parte importante dessas metodologias — os diagramas de classe — é similar, sob muitos aspectos, aos diagramas ER. As operações em objetos são definidas nos diagramas de classe que, além disso, especificam a estrutura do esquema do

banco de dados. As operações podem ser usadas para especificar os requisitos funcionais durante o projeto do banco de dados (Elmasri, Ramez; Navathe, Shamkant B-2005)

No diagrama representado pela Figura 20 temos todas as entidades do projeto e seus relacionamentos.

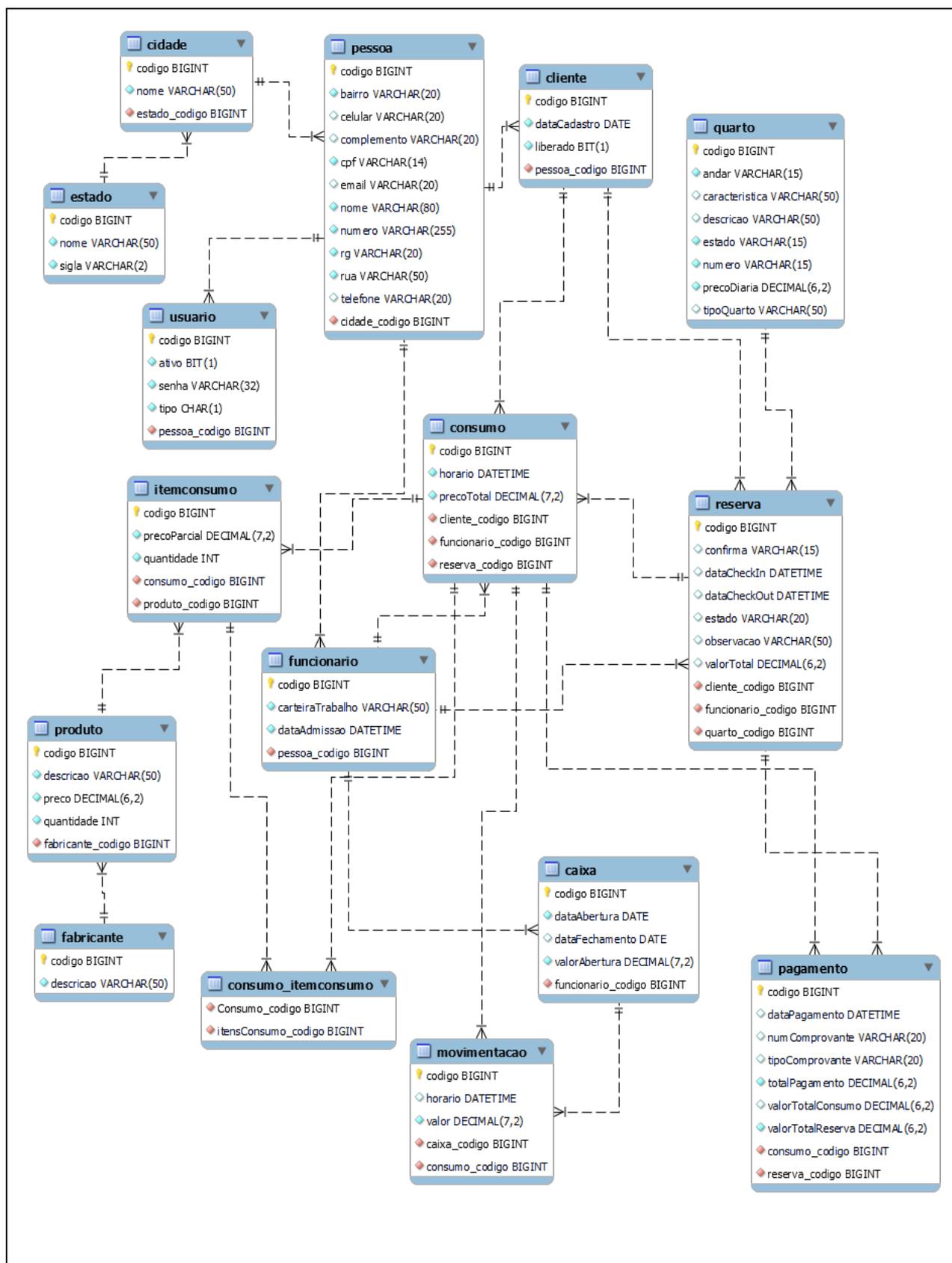


Figura 20 - Diagrama de Entidade Relacionamento (DER)

4.6. DIAGRAMAS DE ATIVIDADES

O Diagrama de Atividade descreve passo a passo os caminhos que serão percorridos para a conclusão da atividade. O Diagrama de Atividade representa o fluxo de controle de uma atividade, podendo ser até de um método de alto grau de complexidade. (Guedes, Gilleanes T. A. 2006).

A Figura 21 ilustra o diagrama de atividades Realizar *CheckIn*.

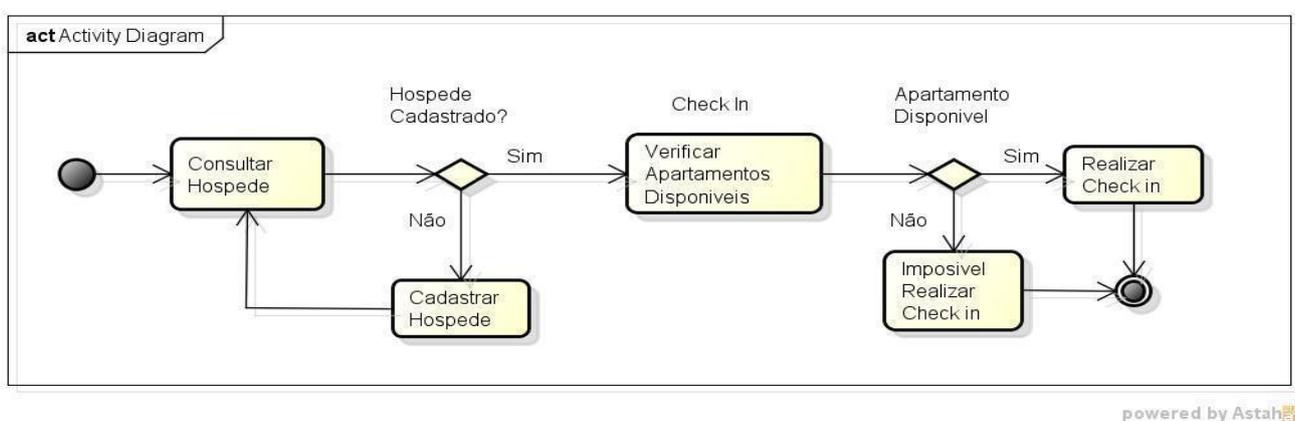


Figura 21- Diagrama de Atividade – Realizar Check-in

A figura 22 ilustra o Diagrama de Atividades Manter Apartamento

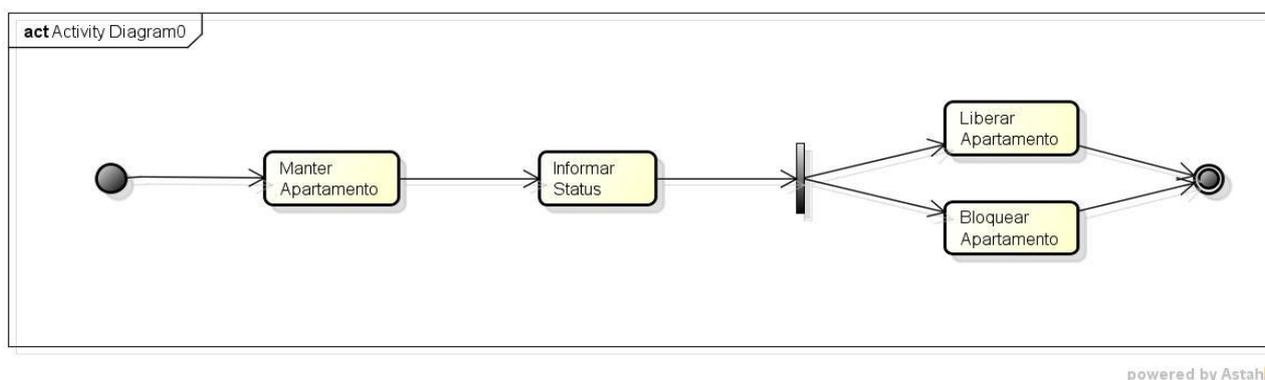


Figura 22 - Diagrama de Atividade – Manter Apartamento

A Figura 23 ilustra o Diagrama de Atividades Alterar Reserva

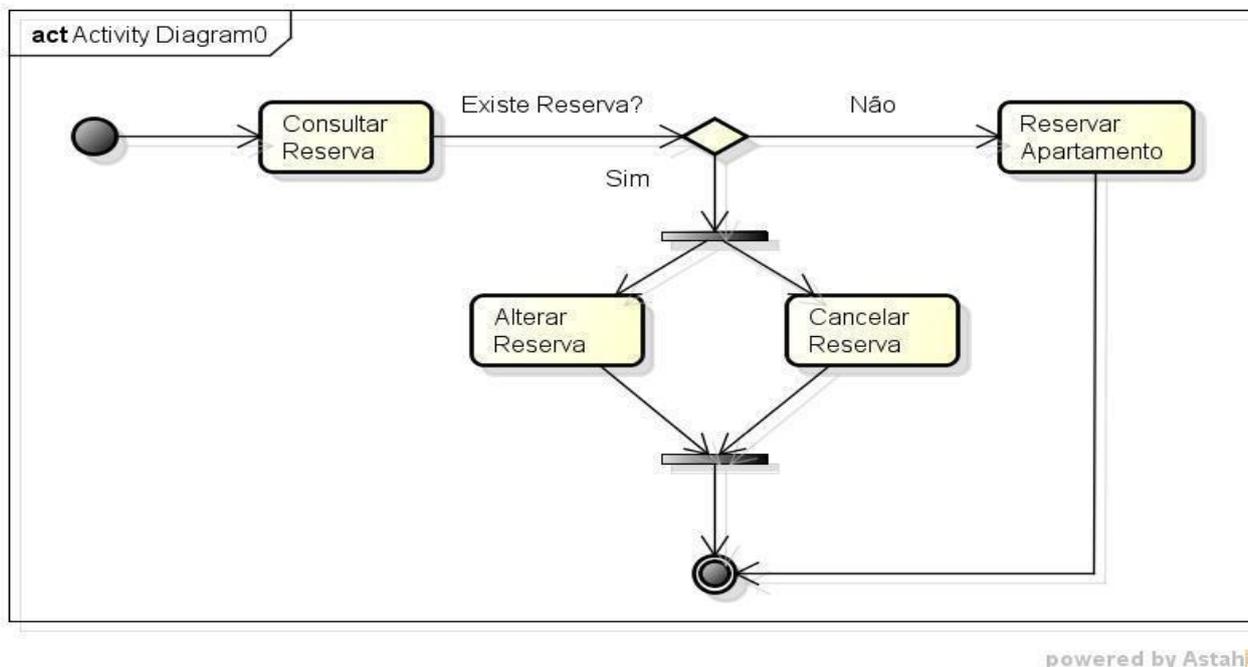


Figura 23 - Diagrama de Atividade – Alterar Reserva

Realizar Check-in, Manter Apartamento, Alterar Reserva, essas atividades representam algumas das principais funcionalidades do sistema de gestão de hotel e pousada em Java. É claro que esse diagrama pode ser expandido e detalhado conforme necessário, de acordo com as necessidades específicas do projeto.

4.7. DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência é um diagrama comportamental que se preocupa com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso definido pelo diagrama de

mesmo nome e apoia-se no diagrama de classes para determinar os objetos das classes envolvidas em um processo. Um diagrama de seqüência costuma identificar o evento gerador do processo modelado, bem como o ator responsável por esse evento, e determina como o processo deve se desenrolar e ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos (Guedes, Gilleanes T. A. 2009).

Na figura 24 é apresentado o diagrama de seqüência cadastrar funcionário ele exhibe os passos que o administrador deve seguir para cadastrar um funcionário.

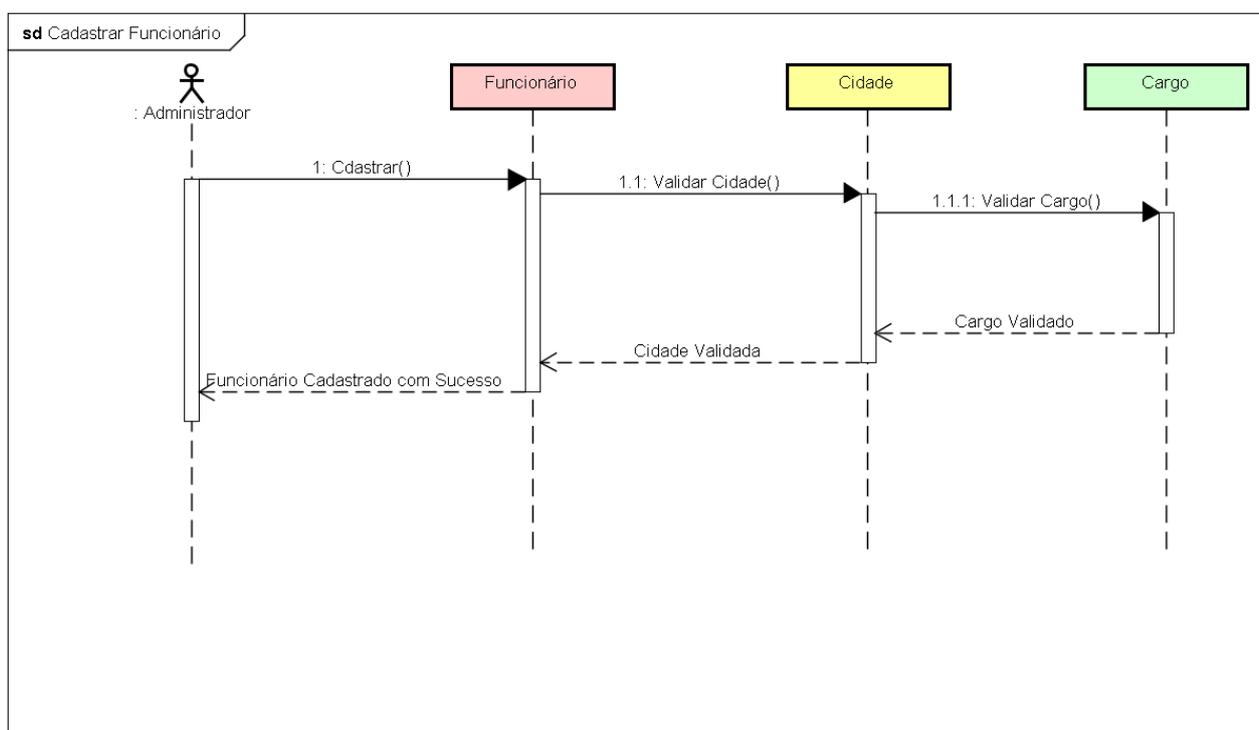


Figura 24 - Diagrama de Seqüência Cadastrar Funcionário

Na figura 25 é apresentado o diagrama de seqüência cadastrar clientes ele exhibe os passos que o administrador deve seguir para cadastrar um cliente. O administrador tem permissão para cadastrar um cliente. Primeiro ele verifica se o cliente já está cadastrado, caso não esteja ele efetua o cadastro.

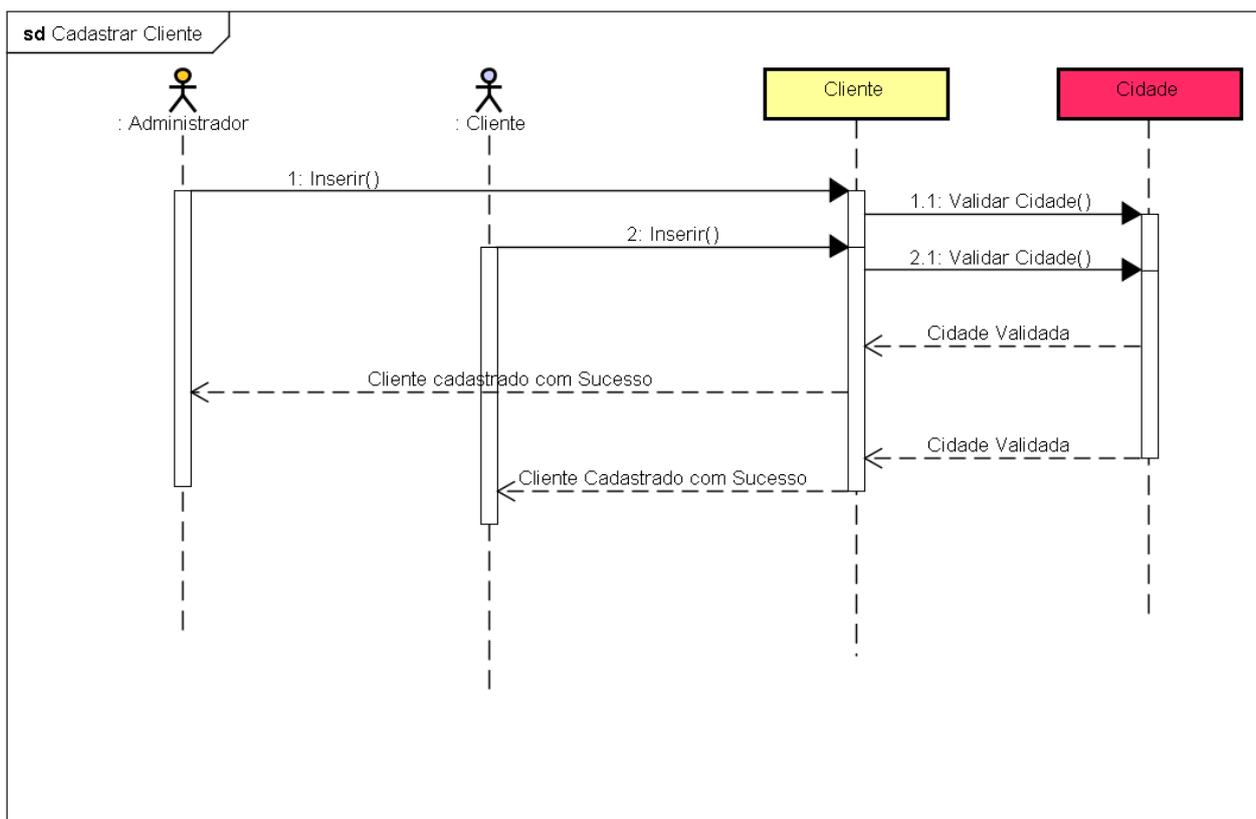


Figura 25 - Diagrama de Sequência Cadastrar Cliente.

Na figura 26 é apresentado o diagrama de sequência cadastrar fornecedor ele exibe os passos que o administrador deve seguir para cadastrar um fornecedor. O administrador tem permissão para cadastrar um fornecedor. Primeiro ele verifica se o fornecedor já está cadastrado, caso não esteja ele efetua o cadastro.

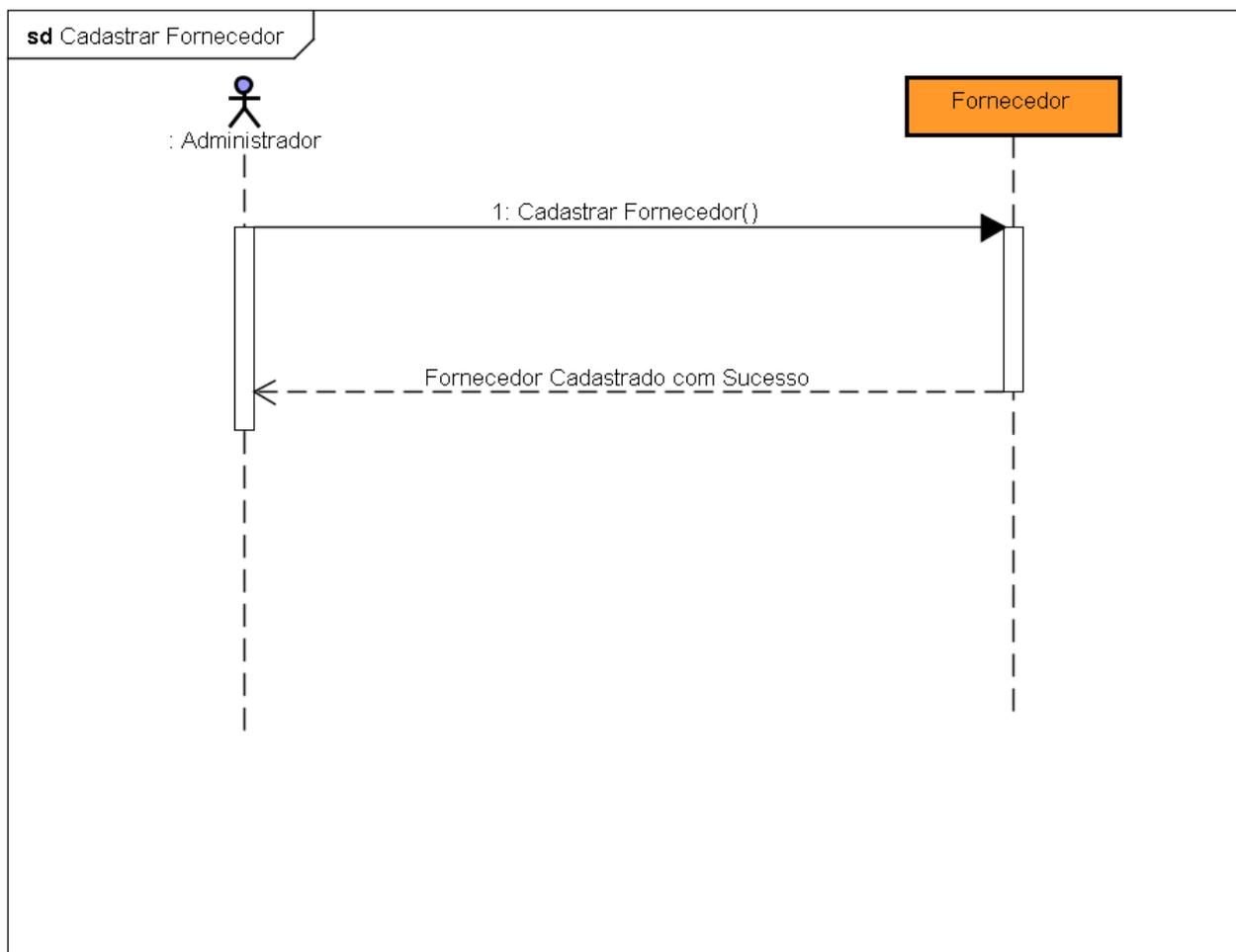


Figura 26 - Diagrama de Sequência Cadastrar Fornecedor.

Na figura 27 é apresentado o diagrama de sequência cadastrar produto ele exibe os passos que o administrador deve seguir para cadastrar um produto. O administrador tem permissão para cadastrar um fornecedor. Primeiro ele verifica se o produto já está cadastrado, caso não esteja ele efetua o cadastro.

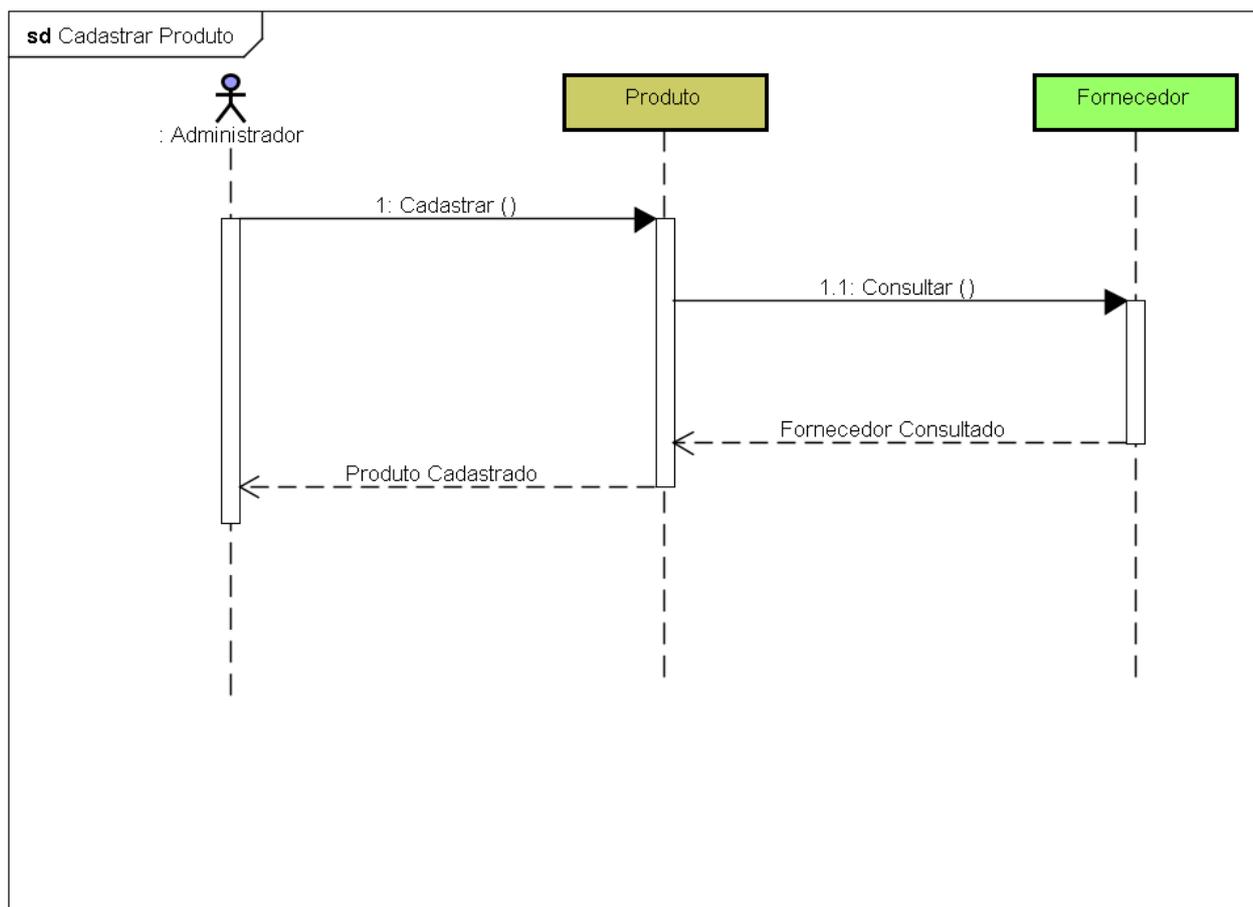


Figura 27 - Diagrama de Sequência Cadastrar Produto.

Na figura 28 é apresentado o diagrama de sequência cadastrar uma cidade ele exibe os passos que o administrador deve seguir para cadastrar uma cidade. O administrador tem permissão para cadastrar uma cidade. Primeiro ele verifica se a cidade já está cadastrada, caso não esteja ele efetua o cadastro.

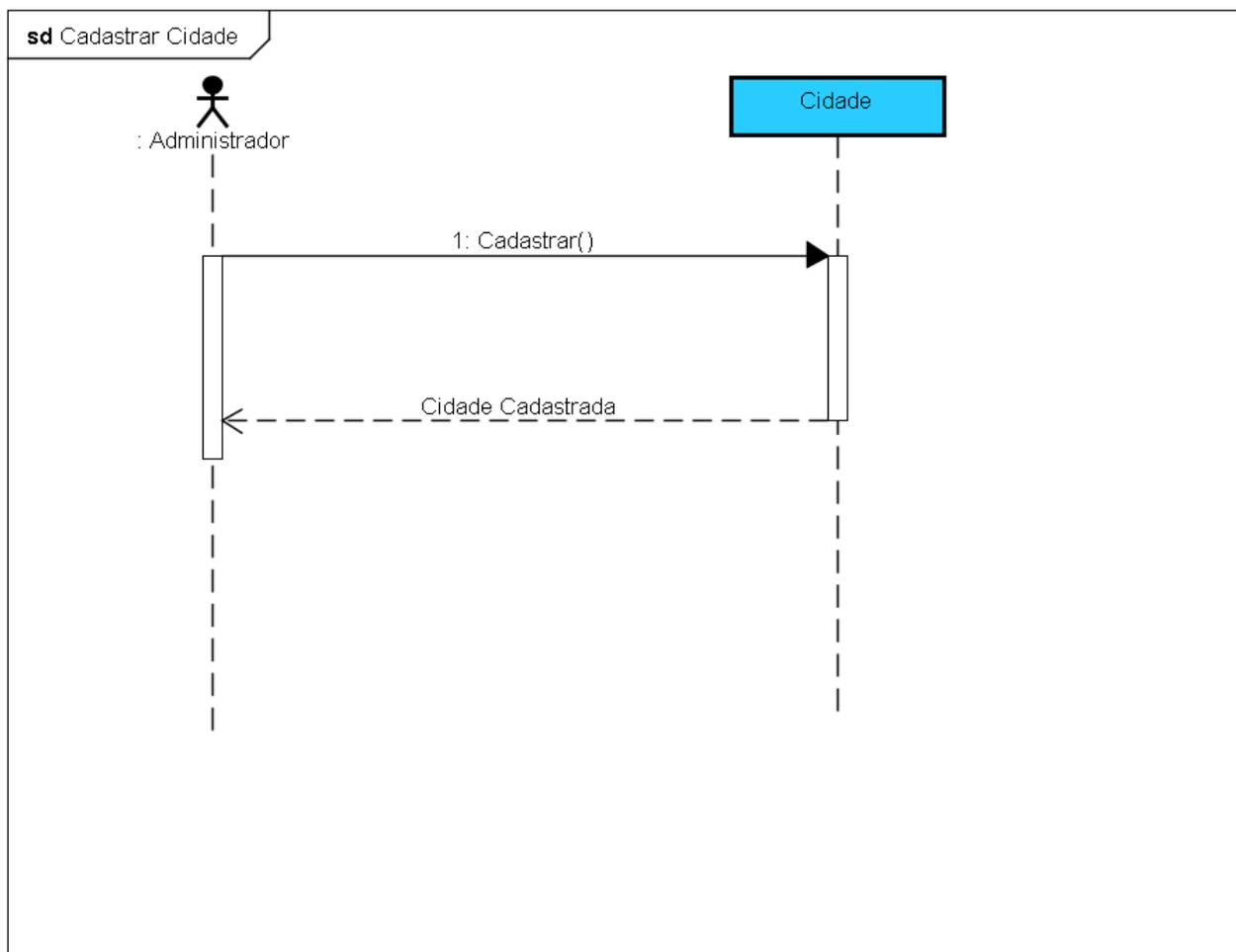


Figura 28 - Diagrama de Sequência Cadastrar Cidade.

Na figura 29 é apresentado o diagrama de sequência cadastrar o pagamento de determinada reserva, ele exhibe os passos que o administrador deve seguir para cadastrar um pagamento. O administrador tem permissão para cadastrar um pagamento. Primeiro ele verifica se o pagamento já está cadastrado, caso não esteja ele efetua o cadastro.

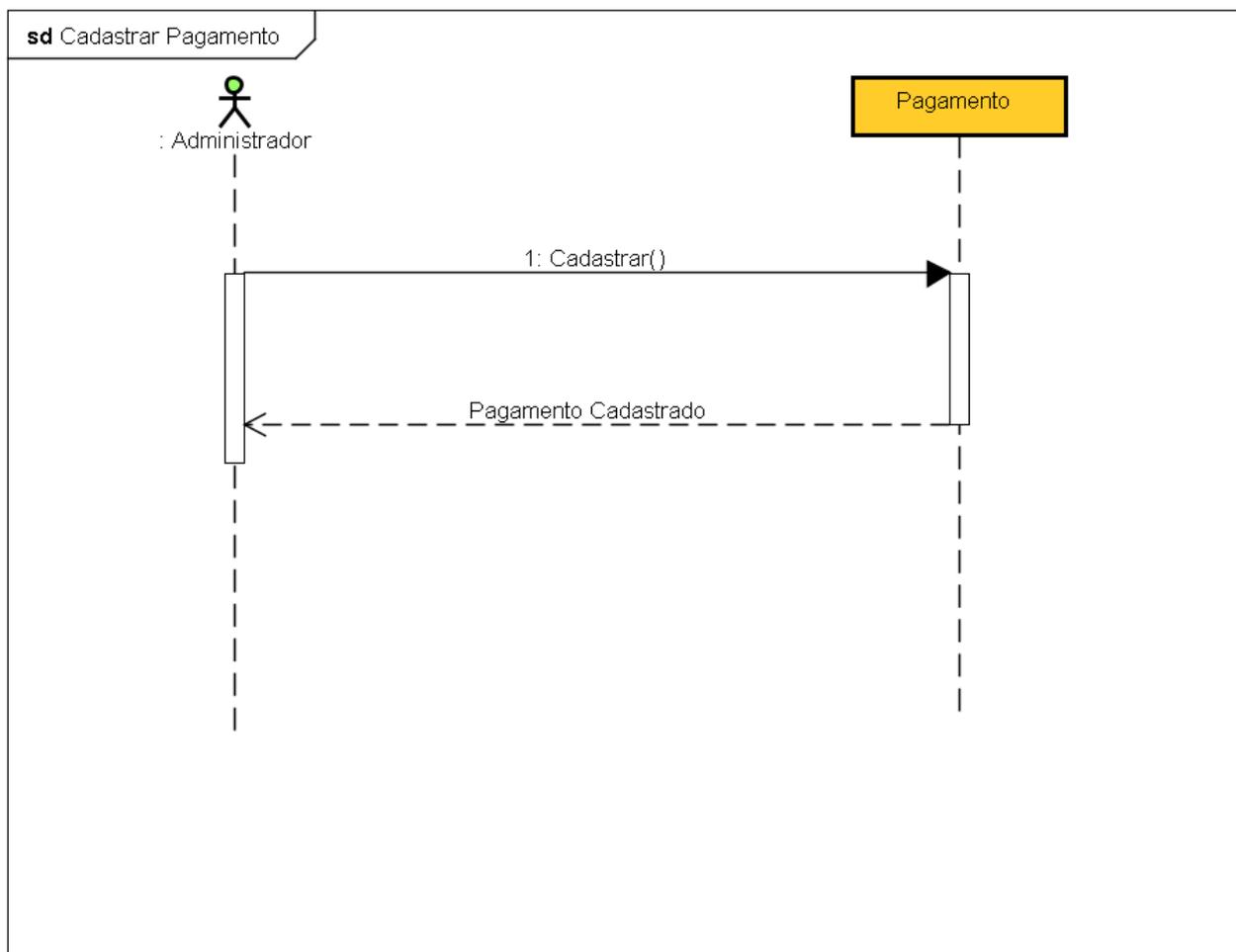


Figura 29 - Diagrama de Sequência Cadastrar Pagamento.

Na figura 30 é apresentado o diagrama de sequência consultar quartos, ele exibe os passos que o administrador deve seguir para consultar o status do quarto, ou seja o quarto pode estar ocupado, disponível ou em manutenção. Este passo é muito importante no ato de efetuar a reserva.

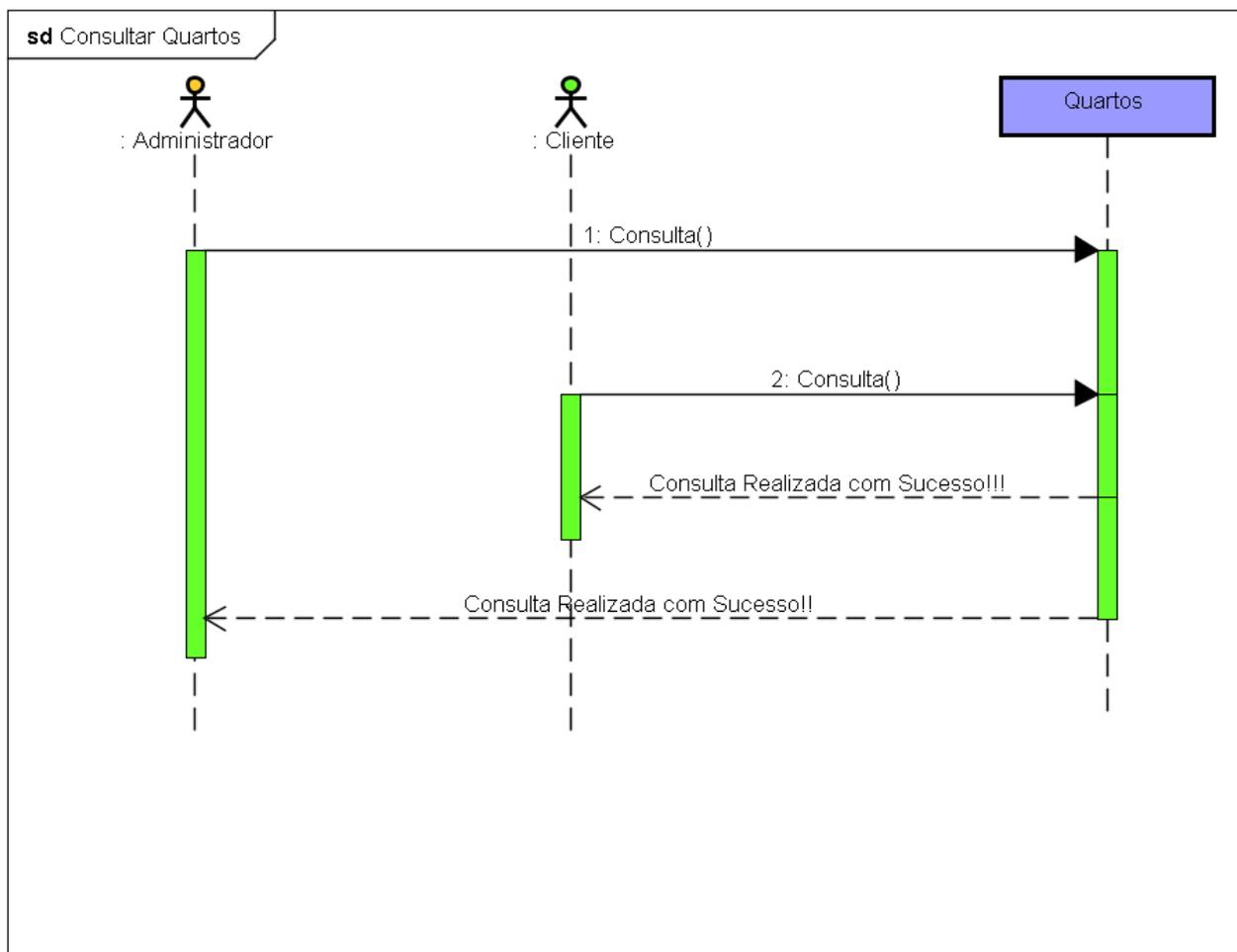


Figura 30 - Diagrama de Sequência Consultar Quartos.

5. IMPLEMENTAÇÃO DO SISTEMA

Um ambiente de desenvolvimento integrado (IDE) é uma aplicação de software que ajuda os programadores a desenvolver código de software de maneira eficiente. Ele aumenta a produtividade do desenvolvedor, combinando recursos como edição, compilação, teste e empacotamento de software em uma aplicação fácil de usar. Assim como escritores usam editores de texto e contadores usam planilhas, desenvolvedores de software usam IDEs para facilitar o trabalho (AWS).

Para a implementação do sistema foi utilizado o ambiente de desenvolvimento Eclipse IDE e a linguagem de programação Java com alguns dos seus frameworks.

5.1. ORGANIZAÇÃO DOS PACOTES E CLASSES DO SISTEMA

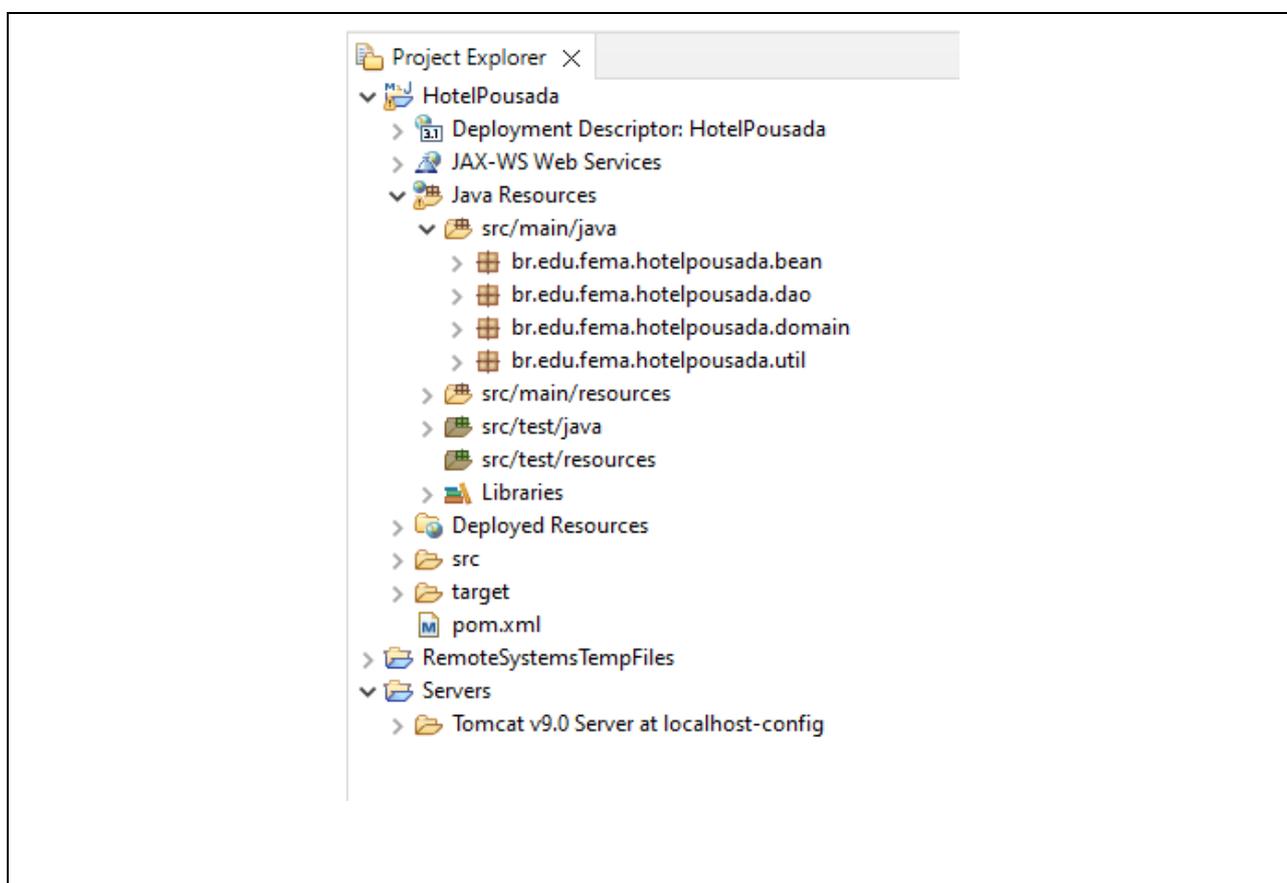


Figura 31 – Organização dos Pacotes e Classes

Pacote: `br.edu.fema.hotelpousada.bean`: este pacote contém as classes *JavaBean* que é uma classe extremamente simples, ela é apenas uma classe que possui apenas atributos e seus respectivos *getters* e *setters*, em geral não possui nenhuma lógica adicional.

Pacote: `br.edu.fema.hotelpousada.dao`: Este pacote está definido o Crud, ou seja, todas classes que serão cadastradas, consultadas, atualizadas e excluídas.

Pacote: `br.edu.fema.hotelpousada.domain` as classes de domínio, tem como função organizar os dados e garantir que as regras do negócio sejam atendidas. As classes de domínio podem ser consideradas como o alicerce da aplicação.

Pacote: `br.edu.fema.hotelpousada.util.util`: Neste pacote estão adicionadas todas as classes que farão conexão com o banco de dados através do *Hibernate*.

5.2. ORGANIZAÇÃO DAS PÁGINAS DO SISTEMA

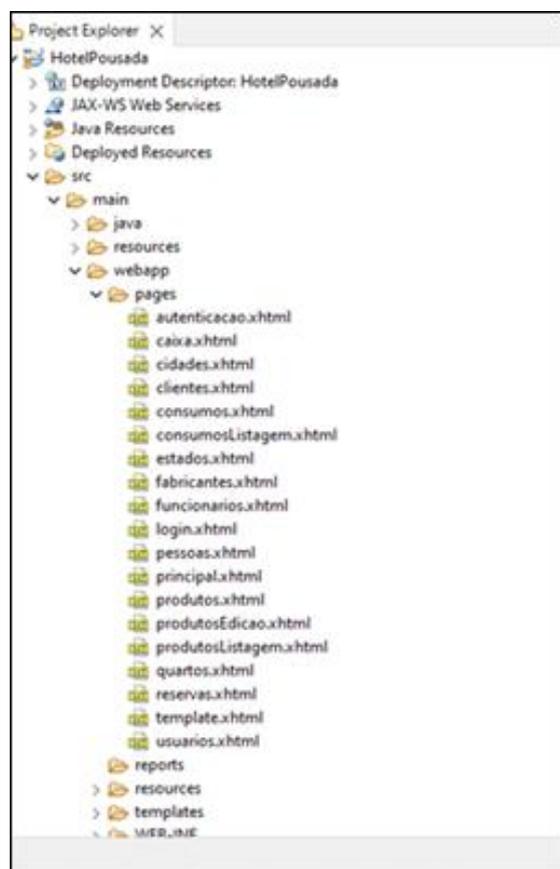


Figura 32 - Organização das Páginas

Páginas WEB foram armazenadas no diretórios webapp/pages para facilitar o trabalho para manter.

5.3. ORGANIZAÇÃO DAS TELAS DO SISTEMA

5.3.1. Tela Inicial

A figura 33 é a tela inicial do sistema. O usuário para ter acesso ao sistema deverá efetuar o seu login, fornecendo o nome de usuário e a senha. O nome e a senha serão armazenados no banco de dados, a senha será criptografada usando o framework Apache Shiro ele é uma estrutura de segurança Java poderosa, e fácil de usar que executa autenticação, autorização, criptografia e gerenciamento de sessão.

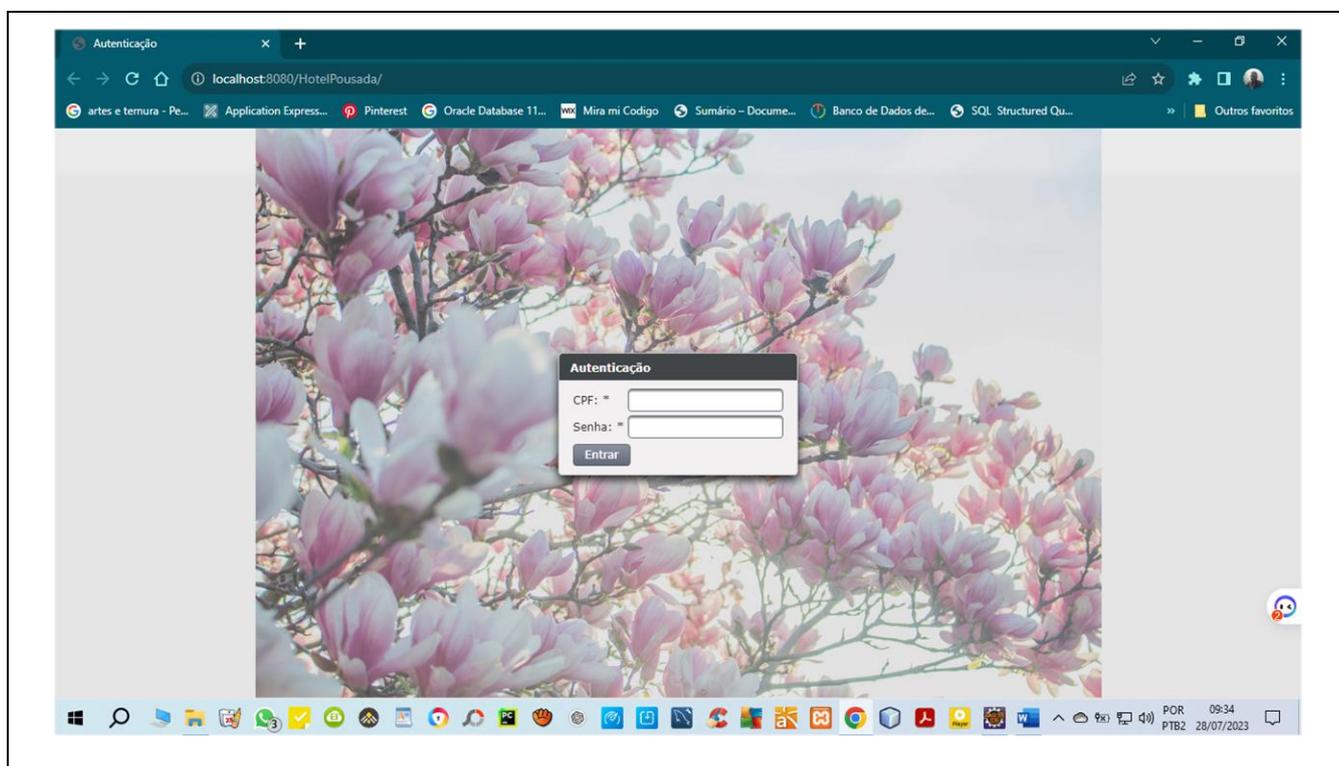


Figura 33 - Tela Inicial do Sistema

5.3.2. Tela do Sistema

O usuário após ter efetuado o login e ser autenticado ele terá acesso à tela do sistema que onde ele poderá acessar as telas as quais ele tem permissão. O sistema concede permissões aos vários tipos de usuários. Existem três tipos de usuários: administrador, gerente e funcionário. No canto inferior esquerdo é exibido o nome do usuário logado. A figura 34 exibe a tela do sistema.

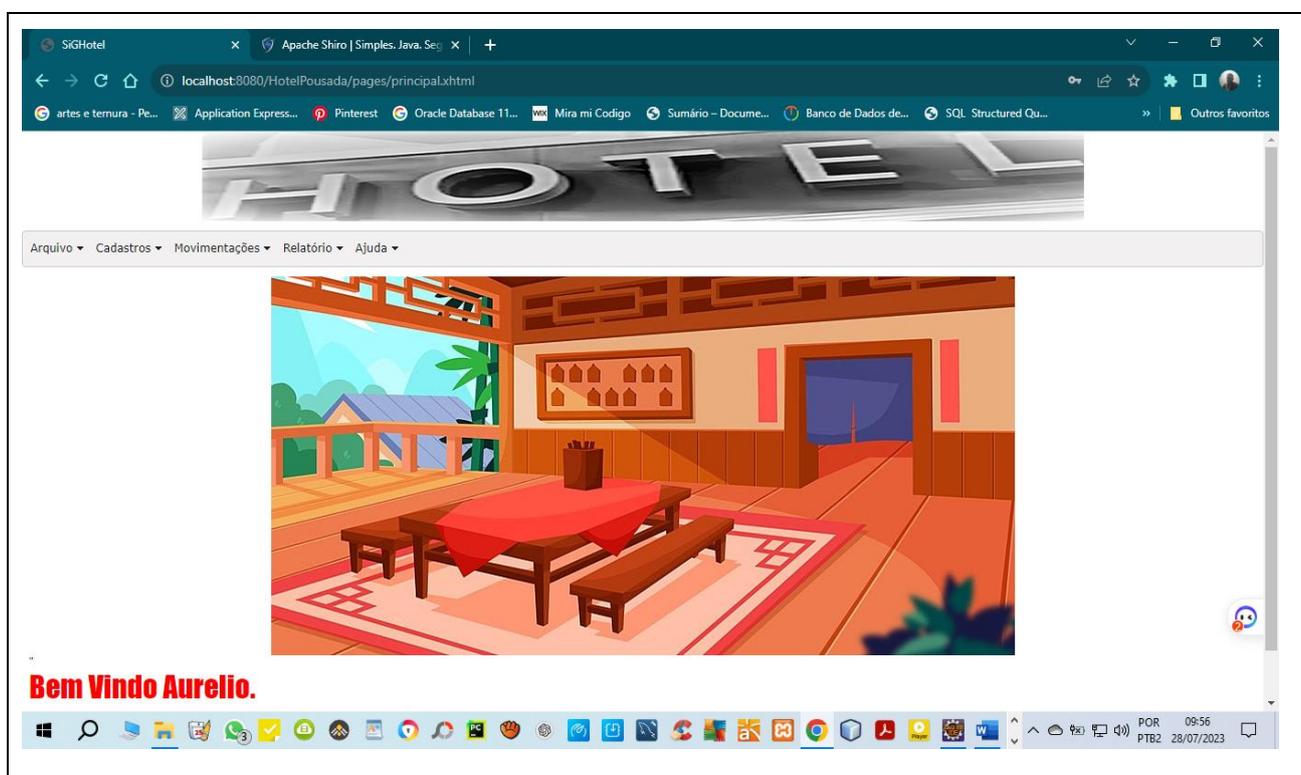


Figura 34 - Tela do Sistema

5.3.3. Tela de Cadastro de Clientes

Caso o usuário escolha a opção de cadastro de cliente ele será conduzido à tela conforme a figura 35 para efetuar o cadastro. Nesta tela ele tem a possibilidade de editar, excluir e cadastrar um novo cliente. As demais telas de cadastro são similares e não serão exibidas.

Arquivo ▾ Cadastros ▾ Movimentações ▾ Relatório ▾ Ajuda ▾

Clientes - Listagem

Novo

Nome ↕	CPF ↕	Data de Cadastro ↕	Liberado ↕	Opções
Andressa	333.333.333-33	4 de jul. de 2023	<input checked="" type="checkbox"/>	
Bruna	111.111.111-11	8 de jul. de 2023	<input type="checkbox"/>	
Raul	999.999.999-99	18 de jul. de 2023	<input checked="" type="checkbox"/>	

TCC - 2023 - FEMA-ASSIS

Windows taskbar: 10:09, 28/07/2023

Figura 35 - Tela de Cadastro de Usuários

5.3.4. Tela de Consumos

Esta tela da figura 36 é onde é feito o consumo dos quartos, nela é especificado os consumos e os itens de consumos de cada quarto.

Arquivo ▾ Cadastros ▾ Movimentações ▾ Relatório ▾ Ajuda ▾

Produtos

Descrição ↕	Quantidade ↕	Preço ↕	Fabricante ↕	Opções
Conti Cola	25	3,90	Conti	+
Guarana	15	3,90	Antartica	+
Guarana Conti	4	3,90	Conti	+

⏪ ⏩ 1 2 3 ⏪ ⏩

Cesta de Consumo

Descrição	Preço Un.	Qtde.	Preço Parcial	Opções
Bhoeme	4,90	1	4,90	-
Cerveja	2,90	1	2,90	-
Guarana	3,90	1	3,90	-

⏪ ⏩ 1 ⏪ ⏩

Finalização

Total: R\$ 11,70

Finalizar

TCC - 2023 - FEMA-ASSIS

Português (Brasil)
Teclado Português (Brasil ABNT2)

Para alternar os métodos de entrada, pressione a tecla Windows+Espaço.

POR 10:22
PTB2 28/07/2023

Figura 36 - Tela de Consumo por Quarto

A tela da Figura 37 exibe os consumos e seus respectivos itens.

The screenshot displays a web browser window with the URL `localhost:8080/HotelPousada/pages/consumosListagem.xhtml`. The application interface features a navigation menu with options: Arquivo, Cadastros, Movimentações, Relatório, and Ajuda. The main content area is titled "Consumos - Listagem" and contains a table with the following data:

Horario	Cliente	Funcionário	Preço Total
27/07/2023 04:06	Jaime David Verderesi - 924.327.518-68	Bruna	11,70
20/07/2023 17:14	Bruna - 111.111.111-11	Andressa	6,80
20/07/2023 17:14	Bruna - 111.111.111-11	Andressa	6,80

Below the table, there are detailed views for each consumption item. The first detailed view shows a client named "Jaime David Verderesi" who consumed 3 units of "cerveja" for a total price of 11,70. The second detailed view shows a client named "Bruna" who consumed 30 units of "Conti Cola" for a total price of 20,00. The interface also includes a pagination control at the bottom of the table and a system tray at the bottom of the browser window showing the date and time as 10:32 on 28/07/2023.

Figura 37 - Tela exibe Consumos e itens

6. CONCLUSÕES E TRABALHOS FUTUROS

Ao desenvolver um sistema em Java para gestão de hotel e pousada, é importante considerar as conclusões e os trabalhos futuros para garantir a evolução e a continuidade do projeto. Algumas possíveis conclusões e trabalhos futuros incluem:

6.1 CONCLUSÕES

O sistema permite uma gestão eficiente e organizada de reservas, hospedagens, pagamentos e outras atividades relacionadas a um hotel ou pousada. A utilização de padrões de projeto, arquitetura bem definida e testes automatizados permite uma maior qualidade, manutenibilidade e escalabilidade do sistema. O *design responsivo* do sistema permite que ele seja acessado por dispositivos móveis, tablets e computadores de mesa, aumentando a sua usabilidade e acessibilidade. A integração com outras ferramentas pode permitir uma maior automatização de processos e melhorias em atividades como reservas online, gestão de marketing e outras atividades.

6.2. TRABALHOS FUTUROS

Implementar novas funcionalidades, como integração com sistemas de pagamentos mais abrangentes, sistemas de gestão de estoque e outras atividades que possam melhorar a eficiência do sistema. Melhorar a interface do usuário, aprimorando o design e a usabilidade do sistema para torná-lo mais amigável e fácil de usar. Integrar o sistema com outras tecnologias, como inteligência artificial e *machine learning*, para melhorar a precisão das previsões de reservas, otimização de tarifas e outras atividades. Garantir a segurança do sistema com a implementação de medidas de segurança adicionais, como autenticação de dois fatores, criptografia mais forte e outras medidas de segurança avançadas.

BIBLIOGRAFIA

[2020?], D. I. À. J. –. J. P. A. [I. DEVMEDIA. Introdução à JPA. **DEVMEDIA**. Disponível em: <<https://www.devmedia.com.br/introducao-a-jpa-java-persistence-api/28173>>. Acesso em: 20 nov. 2022.

[S.l.]: [s.n.].

BAUER, C.; KING, G. **Java Persistence Com Hibernate**. 3ª ed. ed. São Paulo: Manning, 2007.

CONSULTING, E. EUAX Consulting. Disponível em: <<https://www.euax.com.br/2018/12/eap-estrutura-analitica-projeto/#o-que-e-eap>>. Acesso em: 01 20 2023.

DEVMEDIA. Introdução ao PrimeFaces. **www.devmedia.com**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-primefaces/33139>>. Acesso em: 10 nov. 2022.

ELMASRI, R.; NAVATHE, S. B. **SISTEMAS DE BANCO DE DADOS**. 4ª Edição. ed. São Paulo: Pearson Education do Brasil Ltda., 2005.

FARIA, T. **Java EE 7 com JSF, PrimeFaces e CDI**. São Paulo: AlgaWorks Softwares, Treinamentos e Serviços Ltda, 2013.

GOMES, Y. M. P. **Java na Web com JSF, Spring, Hibernate e Netbeans 6**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2008.

GUEDES, G. T. A. **Uml - uma Abordagem Prática**. São Paulo: Novatec, 2006.

PACIEVITCH. infoescola.com. **www.infoescola.com/informatica/mysql/**. Disponível em: <<https://web.archive.org/web/20171218104341/http://www.luis.blog.br:80/o-que-e-banco-de-dados.aspx>>. Acesso em: 12 set. 2012.

QUEIRÓS, R.; PORTELA, F. **Introdução ao Desenvolvimento Moderno Para a Web. Do Front-End ao Back-End. Uma Visão Global!** 1ª ed. ed. São Paulo: FCA, 2018.

SANTOS, J. N.; SILVA, J. A. S. SGBD MySQL. **FCCCAT**. Disponível em:
<https://fit.faccat.br/~jonis/Artigo_mySQL.pdf>. Acesso em: 12 ago. 2023.

TECHNOLOGIES., C. [S.l.]: [s.n.]. Disponível em:
<<https://www.cedrotech.com/blog/rabbitmq-o-que-e-e-como-utilizar/>>. Acesso em: 13 jan.
2023.