

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE
CAMPUS AVANÇADO LAJES

MARIA RAQUEL ALVES XAVIER
MYCKAEL SCHUMACHER FERNANDES DA SILVA

SISTEMA DE GERENCIAMENTO DA POUSADA CABUGI

LAJES – RN

2019

MARIA RAQUEL ALVES XAVIER

MYCKAEL SCHUMACHER FERNANDES DA SILVA

SISTEMA DE GERENCIAMENTO DA POUSADA CABUGI

Relatório de Prática Profissional apresentado ao Curso Técnico de Nível Médio em Informática, na modalidade Integrado, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Campus Avançado Lajes, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Orientador: Prof. Me. Dannilo Martins Cunha.

LAJES – RN

2019

MARIA RAQUEL ALVES XAVIER
MYCKAEL SCHUMACHER FERNANDES DA SILVA

SISTEMA DE GERENCIAMENTO DA POUSADA CABUGI

Relatório de Prática Profissional apresentado ao Curso Técnico de Nível Médio em Informática, na modalidade Integrado, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Campus Avançado Lajes, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Trabalho apresentado e aprovado em ___/___/___, pela seguinte Banca Examinadora:

Dannilo Martins Cunha

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Dannylo Johnathan Bernadino Egidio

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Diogo Eugênio da Silva Cortez

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

AGRADECIMENTOS

Primeiramente agradecemos à Deus por ter nos guiado ao longo do desenvolvimento desse trabalho e nos sustentado nos momentos de dificuldades. Ao nosso orientador, o professor Dannilo Martins Cunha por todo o conhecimento repassado, para que concluíssemos com êxito esse projeto, agradecemos pela paciência e pela dedicação a nós, seus orientandos. Agradecemos também aos demais professores que estiveram conosco em toda a nossa trajetória dentro do IFRN.

Á aqueles que garantiram o nosso conforto e permanência na instituição, os nossos amigos servidores técnicos e terceirizados. Aos motoristas que estiveram diariamente em nosso trajeto Pedra Preta/Lajes que sempre nos conduziram com cuidado e atenção.

Agradecemos ainda, aos nossos amigos que nos apoiaram, incentivaram e acreditaram em nós, eles são peças fundamentais em nossa caminhada.

E por fim, somos gratos aos nossos familiares que estão conosco em todos os momentos, nos apoiando, incentivando, acreditando em nosso potencial e não medindo esforços para contribuir na construção dos nossos sonhos.

RESUMO

Na atualidade o setor hoteleiro encontra-se em grande desenvolvimento em relação à utilização das tecnologias, por esse motivo o uso da automação em pousadas está sendo bastante viável para os empresários, uma vez que, a utilização das tecnologias está permitindo conquistar produtividade e eficiência nas empresas, além da diminuição de gastos e a satisfação dos clientes. Este trabalho apresenta um *software*, que tem como objetivo gerenciar alguns dos serviços oferecidos pela Pousada Cabugi, empresa de pequeno porte localizada na cidade de Lajes/RN. O sistema garantirá ao gerente da pousada fazer *check-in* e o *check-out* do cliente de forma mais rápida e fácil, como por exemplo cadastrar clientes, realizar reservas, efetuar pagamento e disponibilizar quarto. Será apresentado também alguns dos princípios da engenharia de software que serviram de base para a produção do sistema. E por fim, apresentaremos tanto os resultados do *software* como os possíveis aprimoramentos futuros.

Palavras-chave: *software*, sistema, pousada, engenharia, tecnologia.

ABSTRACT

Currently, the hotel sector is in great development due to the use of technologies. For that reason the use of the automation in inns is quite feasible for the entrepreneurs, since the use of the technology is achieving success and efficiency in the companies, as well as reducing payments and customer satisfaction. This work presents a software that aims at managing some of the services offered by Pousada Cabugi, a small business located in the city of Lajes / RN. The system will guarantee that the inn manager makes the check-in and check-out of the client faster and in an easier manner, for example, by registering clients, making reservations, making payment and making room available. It will also present some of the principles of software engineering that form the basis for the production of the system. In conclusion, we will present the software results as possible future enhancements.

Keywords: software, system, inn, engineering, technology.

LISTA DE ILUSTRAÇÕES

Figura 1.Arquitetura MVC	Erro! Indicador não definido.
Figura 2.Subdivisões da linguagem SQL	Erro! Indicador não definido.
Figura 3.Diagrama de caso de uso do sistema.....	Erro! Indicador não definido.
Figura 4.Diagrama de classe do sistema.....	Erro! Indicador não definido.
Figura 5.Tela inicial do sistema.....	Erro! Indicador não definido.
Figura 6.Tela de cadastro de cliente	Erro! Indicador não definido.
Figura 7.Tela de atualização de telefone	Erro! Indicador não definido.
Figura 8. Tela de atualização de endereço.....	Erro! Indicador não definido.
Figura 9.Tela de consulta de cliente	Erro! Indicador não definido.
Figura 10.Tela exclusão de cliente	Erro! Indicador não definido.
Figura 11.Tela de reserva de quarto	Erro! Indicador não definido.
Figura 12.Tela de cancelamento de reserva.....	Erro! Indicador não definido.
Figura 13.Tela de consulta da reserva e efetuação de pagamento	Erro! Indicador não definido.
Figura 14.Modelo relacional.....	Erro! Indicador não definido.
Figura 15.Cadastro do cliente.....	Erro! Indicador não definido.
Figura 16.Resposta do cadastro	Erro! Indicador não definido.
Figura 17.Cadastro de reserva	Erro! Indicador não definido.
Figura 18.Resposta do cadastro de reserva.....	Erro! Indicador não definido.

LISTA DE QUADROS

Quadro 1.Requisitos funcionais..... **Erro! Indicador não definido.**

Quadro 2.Requisitos não funcionais..... **Erro! Indicador não definido.**

LISTA DE SIGLAS

SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
SGBD	Sistema Gerenciador de Banco de Dados
MVC	Model, View, Controller
HTML5	HyperText Markup Language
PHP	Personal Home Page
C/C++	see plus plus
OO	Orientado a Objeto
UML	Unified Modeling Language
API	Application Programming Interface
SQL	Structured Query Language
DQL	Data Query Language
DML	Data Manipulation Language
DDL	Data Definition Language
DCL	Data Control Language
DTL	Transaction Control Language
GNU	Gnu's Not Unix
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
CPF	Cadastro de Pessoas Físicas
WEB	World Wide Web

SUMÁRIO

1. INTRODUÇÃO.....	9
1.1. OBJETIVO GERAL.....	10
1.2. OBJETIVOS ESPECÍFICOS	10
1.3. METODOLOGIA.....	10
1.4. ESTRUTURA DO TRABALHO	11
2. FUNDAMENTAÇÃO TEÓRICA	11
2.1. SOFTWARE.....	11
2.2. PARADIGMAS DA PROGRAMAÇÃO	12
2.3. LINGUAGEM DE PROGRAMAÇÃO	12
2.4. BANCO DE DADOS	13
2.5. ARQUITETURA MVC.....	13
2.6. ENGENHARIA DE SOFTWARE	14
2.6.1 DIAGRAMA DE CASO DE USO.....	15
2.6.2 DIAGRAMA DE CLASSE.....	15
3. FERRAMENTAS UTILIZADAS.....	16
3.1. JAVA.....	16
3.2. NETBEANS.....	16
3.3. SQL.....	16
3.4. POSTGRESQL	17
3.5. ASTAH COMMUNITY	17
4. DESENVOLVIMENTO.....	18
4.1. REQUISITOS	18
4.1.1 REQUISITOS FUNCIONAIS.....	18
4.1.2 REQUISITOS NÃO FUNCIONAIS	19
4.1.3 DIAGRAMA DE CASO DE USO DO SISTEMA.....	20
4.1.4 DIAGRAMA DE CLASSE DO SISTEMA	20
4.2. SOFTWARE DESENVOLVIDO.....	21
4.3. BANCO DE DADOS CONSTRUÍDO.....	25
4.3.1 MODELO RELACIONAL.....	25

4.4. TESTES DO PROGRAMA.....	26
5. CONCLUSÃO.....	29
5.1. TRABALHOS FUTUROS	29
6. REFERÊNCIAS BIBLIOGRÁFICAS	30

1. INTRODUÇÃO

Conforme a cartilha do Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE), o ramo da hotelaria (hotéis, pousadas, chalés) tem crescido bastante nos últimos anos no Brasil, desta forma é necessário que esses estabelecimentos possuam uma boa organização dos serviços ofertados, para garantir seu destaque de qualidade no mercado.

A tecnologia vem sendo uma forte aliada na modernização e avanço das empresas. De acordo com Cachoeira:

“O mercado hoteleiro talvez tenha sido um dos que mais sofreu influência do avanço tecnológico. Isso acontece não apenas porque a chegada da internet mudou completamente o jogo, mas também porque todo hotel que se preze deve ter um sistema eficiente de gestão da informação para que possa atender bem seus hóspedes.” (2017)

Para SANTOS (2017) “O uso da tecnologia para agilizar e otimizar os serviços hoteleiros já é algo inerente à prática”. Com isso, os empresários têm buscado ferramentas tecnológicas que possibilitem a realização de tarefas de maneira mais simples, ágil e eficiente nas suas empresas.

Os *softwares* de gestão hoteleira vêm se destacando em meio a empresas de grande e pequeno porte. Tal recurso tecnológico deixa de ser apenas uma ferramenta útil e se torna um requisito básico para as empresas. SANTOS (2017) afirma que, “Um software é uma maneira segura de organizar o planejamento e o controle predial do hotel ou da rede, ajudando a reduzir gastos e custos operacionais”.

Porém, existem ainda pousadas que não aderiram a esse requisito que o mercado sugere, um exemplo é a pousada Cabugi, empresa no qual foi feito o nosso estudo de caso. Tal negócio pode ser classificado como uma empresa de pequeno porte, porém atende a uma grande demanda de pessoas. A pousada está localizada em uma cidade do interior, a pousada ainda não possui uma ferramenta tecnológica que a auxilie no gerenciamento da empresa. Há ainda um trabalho manual excessivo para cadastro de quartos e reservas, *check-in* e *check-out* de clientes e gerenciamento financeiro.

Diante de tal situação, é proposto um software que sirva de subsídio para um melhor gerenciamento de alguns serviços, diminuição do trabalho manual excessivo e para que a pousada possa também se destacar na rede hoteleira, pois conforme um levantamento feito no primeiro semestre de 2017 pela Avaya, empresa global especializada em fornecer soluções tecnológicas, a implementação de novas tecnologias são fundamentais para o setor hoteleiro.

1.1. Objetivo Geral

Este projeto tem como objetivo principal desenvolver um *software* para o gerenciamento dos serviços oferecidos pela Pousada Cabugi, para que desta forma o trabalho manual excessivo possa diminuir.

1.2. Objetivos Específicos

- a. Conhecer o funcionamento dos serviços da pousada;
- b. Identificar os problemas presentes no estabelecimento;
- c. Compreender como os problemas podem ser resolvidos;
- d. Realizar levantamento das ferramentas que melhores auxiliarão na produção do sistema;
- e. Implementar o software.

1.3. Metodologia

Etapa 1: Pré-Produção ou coleta de requisitos

Inicialmente, para o desenvolvimento do projeto e construção do software, fomos em busca dos dados fundamentais para a compreensão da situação do estabelecimento. Realizamos uma visita a referida pousada e fizemos uma entrevista com o responsável pela administração do estabelecimento. Por meio de uma conversa com o dono da empresa foi possível a coleta das informações necessárias para a identificação dos problemas e possíveis complicações. A partir disso, a proposta é a implantação de um *software* que permita solucionar ou amenizar os problemas encontrados.

Etapa 2: Análise dos dados e início da proposta de intervenção

Após a análise dos dados e identificação dos problemas, iniciamos a fase de intervenção. No qual utilizamos os programas PostgreSQL e Netbeans, para construir um produto de *software* que supra a deficiência da pousada.

O MySQL servirá para organizar as informações do banco de dados, como também possibilitará a função de armazenar os cadastros e o controle de fluxo. Enquanto o NetBeans

servirá para a programação, elaboração da interface gráfica do produto e da lógica de negócio da pousada.

Etapa 3: Pós-produção ou Revisão e entrega dos Resultados

Com todas as fases anteriores concluídas, faremos a revisão da parte escrita e entregaremos o produto ao cliente. Por fim, iremos registrar e apresentar os resultados obtidos através deste trabalho de conclusão de curso, dando-o como concluído.

1.4.Estrutura do Trabalho

Este trabalho de conclusão de curso foi organizado em sete capítulos. O primeiro capítulo contém a introdução e objetivos relacionados ao trabalho, como também a metodologia utilizada. No segundo capítulo é destacado alguns conceitos que serviram de embasamento para o desenvolvimento do sistema. O terceiro capítulo apresenta as ferramentas que auxiliaram na implementação do *software*. Já o quarto capítulo expõe os resultados que foram criados ao decorrer do desenvolvimento do projeto. No quinto é evidenciado por parte dos estudantes o desejo por aprimorar o sistema no futuro. O sexto capítulo é apresentado as conclusões acerca do projeto. E por fim, no último capítulo é feito o levantamento das referências bibliográficas.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é abordado alguns tópicos relacionados aos aspectos teóricos do trabalho, tais como *software*, linguagem de programação, diagrama de classe, diagrama de caso de uso e entre outros. Os tópicos tratados neste capítulo foram de suma importância para o desenvolvimento do projeto.

2.1. Software

O *Software* é todo o conjunto de programas (componentes lógicos) que compõem um computador, possuindo uma forte ligação com o mesmo, pois ele é o componente responsável por fornecer instruções à máquina (*Hardware*). Conforme Marcelo Marçula e Pio Armando Benini Filho.

É a parte lógica do sistema de computação armazenada eletronicamente. É composto por um ou mais programas (conjuntos de instruções) que capacitam o hardware a realizar tarefas específicas. Pelos dados manipulados por eles, pela documentação de especificação (projeto) dos programas e pela documentação de operação de programas. (2014, pág. 154)

2.2. Paradigmas da Programação

Geralmente quando pensamos em paradigma, logo nos vem à mente um padrão de pensamento que guia um conjunto de atividades relacionadas. Nesse sentido pode-se entender paradigma de programação como um modelo de resolução de problemas, cuja a utilização de determinados gêneros de programas e linguagens, possibilitam solucionar problemas de diversas maneiras (NOONAN e TUCKER, 2010, pág. 3).

Noonan e Tucker declaram que atualmente existem quatro paradigmas fundamentais que progrediram ao longo dos anos, sendo eles: programação imperativa, programação funcional, programação lógica e programação orientada a objeto (NOONAN e TUCKER, 2010, pág. 3).

Falbo e Barcellos (2011, pág 8) citam que para o paradigma orientado a objetos “[...] o mundo é visto como sendo composto por objetos, onde um objeto é uma entidade que combina estrutura de dados e comportamento funcional.”

A orientação a objetos oferece conceitos que são de suma importância para a modelagem de sistemas. (Falbo e Barcellos, 2011, pág 8)

De acordo com Falbo e Barcellos:

Em um sistema construído segundo o paradigma orientado a objetos (OO), componentes são partes encapsuladas de dados e funções, que podem herdar atributos e comportamento de outros componentes da mesma natureza e cujos componentes comunicam-se entre si por meio de mensagens (2011, pág 8 apud YOURDON, 1994).

Tendo em vista tais atributos deste paradigma, o *software* desenvolvido teve como base os conceitos do paradigma orientado a objetos. Esse modelo serviu de alicerce para a construção do projeto.

2.3. Linguagem de Programação

Linguagens de programação são pontes que estabelecem a comunicação entre seres humanos e computadores. Através delas, pode-se descrever procedimentos e executar funções

para alcançar um objetivo, que geralmente é a solução de algum problema. Marcelo Marçula e Pio Armando Benini Filho definem linguagem, como:

Um conjunto de palavras (vocabulário) e um conjunto de regras gramaticais (para relacionar essas palavras) usadas para instruir o sistema de computação a realizar tarefas específicas e com isso, criar programas. Cada linguagem tem o seu conjunto de palavras-chave e sintaxes. (2014, pág. 169)

Existem diversas linguagens com finalidades diferentes. O *software* proposto foi desenvolvido na linguagem de programação Java, que é uma linguagem de alto nível (mais próxima da linguagem humana) e multiplataforma, ou seja, funciona nos principais sistemas operacionais.

2.4. Banco de Dados

De acordo com Silberschatz, Korth e Sudarshan:

Um *Sistema Gerenciador de Banco de Dados* (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados. O conjunto de dados comumente chamado banco de dados, contém informações sobre uma empresa em particular. (1999, p.1)

O SGBD proporciona aos usuários um ambiente conveniente e eficiente na recuperação e armazenamento das informações contidas no sistema de banco de dados (SILBERSCHATZ, KORTH e SUDARSHAN, 1999).

O sistema de banco de dados é de suma importância nos *softwares* das empresas, uma vez que, tais estabelecimentos prezam pela segurança dos dados pessoais dos seus clientes. Silberschatz, Korth e Sudarshan (1999, p.1) afirmam ainda que “[...] um sistema de banco de dados deve garantir a segurança das informações armazenadas contra eventuais problemas com o sistema, além de impedir tentativas de acesso não autorizadas”.

A utilização do SGBD neste projeto foi um dos requisitos adicionados para que o sistema permitisse uma manipulação dos dados do cliente de forma mais rápida e segura. Além disso, tornar o sistema com mais vantagens, como Marçula e Benini (2014 p.179) citam: “Os bancos de dados armazenam dados substituindo grandes volumes de papéis”.

2.5. Arquitetura MVC

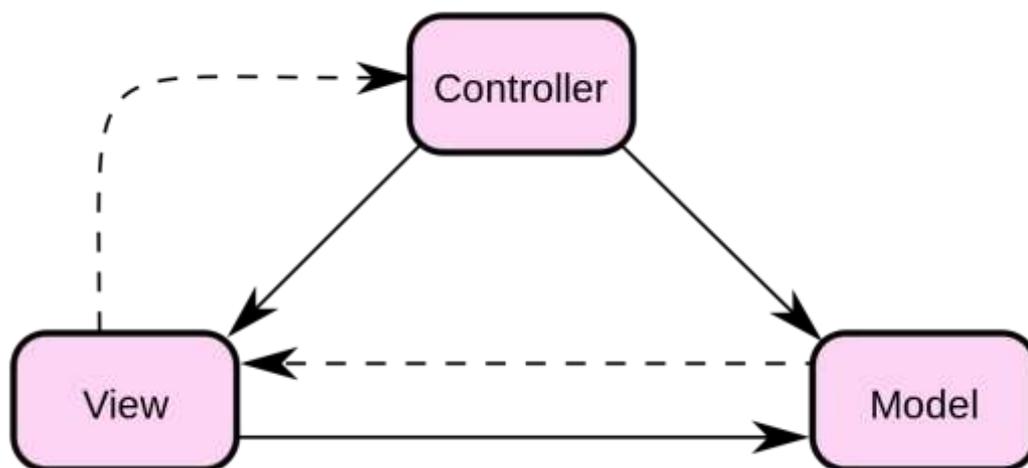
A arquitetura MVC é um modelo de arquitetura de *software* utilizada na engenharia de software. Essa arquitetura tem a função de separar as informações da interação do usuário. A MVC é composta por três camadas, que são separadas de acordo com a aplicação. Suas camadas são: Modelo (*Model*), Visão (*View*) e Controle (*Controller*). (MEDEIROS, 2013)

A camada Modelo (Model) é a camada base da manipulação de dados, é responsável pela leitura e escrita de dados. A camada Visão (View) é considerada simples por fornecer apenas a exibição dos dados a partir de meios como html ou xml. E o Controle (Controller) é responsável por administrar o recebimento de todas as requisições do usuário. Ele tem métodos denominados de *actions* que controlam qual model utilizar e qual view será mostrado ao usuário.

Segundo Fróes (2014) “O Controller pode se comunicar com a View e também com o Model. O Model nunca se comunica com a View, mas a View observa o Model que avisa quando as solicitações foram atendidas para que a View possa mostrá-las.”

Para melhor entendimento segue a figura 1:

Figura 1: Arquitetura MVC



(Fonte: Wikipédia)

2.6. Engenharia de Software

CARVALHO e CHIOSSI (2001) definem a engenharia de software como uma disciplina que é formada por metodologias, métodos e ferramentas que servem como apoio para a resolução dos problemas, desde a percepção do problema até o momento que o sistema desenvolvido deixa de ser operacional.

Nesse sentido, a aplicação da engenharia de software no desenvolvimento de um sistema é essencial, pois é justamente essa engenharia que garante um *software* com alta qualidade. Além disso a engenharia de software molda como deve ser projetado um sistema de qualidade com baixo custo e que seja confiável.

De acordo com cientista Alemão BAUER (2010), “Engenharia de Software é a criação e a utilização de sólidos princípios de engenharia a fim de obter *software* de maneira econômica, que seja confiável e que trabalhe em máquinas reais”.

2.6.1. Diagrama de Caso de Uso (UML)

De acordo com BOOCH, RUMBAUGH e JACOBSON (2006), “Um diagrama de casos de uso mostra um conjunto de casos de uso e atores e seus relacionamentos, sendo utilizado para ilustrar uma visão estática das maneiras possíveis de se usar o sistema”.

Nesse sentido a criação de casos de uso, é de suma importância para se ter um melhor entendimento de como o *Software* desempenhará as funções projetadas pelo desenvolvedor. Além, disso o desenvolvimento do sistema seguindo as especificações dos casos de uso garantirá um *Software* com desempenho satisfatório.

O Diagrama de Caso de Uso especificando o sistema da Pousada Cabugi é apresentado no tópico 4.1.3 (Figura 3)

2.6.2. Diagrama de Classe

Para Tybel:

Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes. (2016)

O diagrama de classes da UML é uma ferramenta de suma importância para auxiliar os profissionais no desenvolvimento de sistemas orientados a objetos. Com o diagrama de classes o desenvolvedor poderá ter um melhor entendimento acerca do que deve ser feito e como deve ser feito na produção do seu projeto (Ventura, 2018). Veja no tópico 4.1.4 a Figura 4 o diagrama de classe do sistema.

3. FERRAMENTAS UTILIZADAS

3.1. Java

Segundo Indrusiak (1996) “Java é uma linguagem computacional completa, adequada para o desenvolvimento de aplicações baseadas na rede Internet, redes fechadas ou ainda programas *stand-alone*”.

Essa linguagem foi criada nos moldes do paradigma da orientação a objetos, proporcionando ao programador o uso de conceitos de herança, polimorfismo e encapsulamento. (Mendes, pág 18, 2009)

A linguagem Java é considerada uma linguagem simples, porém poderosa em ambientes complexos como a internet. Além disso, é uma linguagem eficiente e segura, e seus códigos podem ser tanto interpretados como compilados. O java permite ao programador recursos suficientes para a construção de uma gama de aplicativos. (Indrusiak, págs 2,3,4, 1996)

3.2. NetBeans

O NetBeans IDE é um ambiente de desenvolvimento (programação) integrado, gratuito e de código aberto para desenvolvedores de softwares, sendo apto a utilização das linguagens Java, JavaScript, HTML5, PHP, C/C++, Groovy, Ruby, entre outras. Este *software* está disponível para Windows, Linux e Mac (Wikipédia 2018).

De acordo com a documentação oficial da plataforma NetBeans:

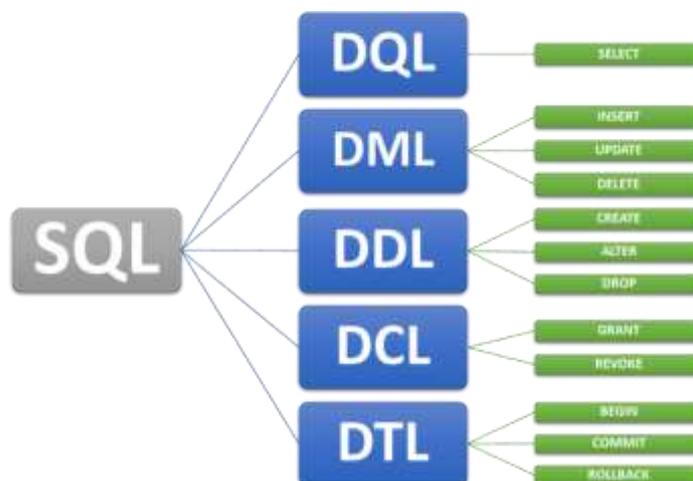
A Plataforma Netbeans é uma ampla framework Java em que você pode basear grandes aplicações desktop. O NetBeans IDE é, em si, uma das centenas de aplicações baseadas na Plataforma NetBeans. A Plataforma NetBeans contém APIs que simplificam o tratamento de janelas, ações, arquivos e muitos outros itens típicos em aplicações.

3.3. SQL

A linguagem SQL (*Structured Query Language*) é responsável pela execução de comandos em banco de dados relacionais. Através da SQL podemos criar tabelas, colunas, índices entre outros, bem como pode-se realizar consultas aos dados armazenados no banco. É através da SQL que ocorre a comunicação com o banco de dados (DEV MEDIA, 2019).

A linguagem SQL é ainda organizada em subconjuntos, cada um com propósitos bem definidos como mostra a figura 2:

Figura 2: Subdivisões da linguagem SQL



Fonte: (DEVMEDIA)

Dentre essas subdivisões o sistema desenvolvido se enquadra na DQL (*Data Query Language*), DML (*Data Manipulation Language*) e DDL (*Data Definition Language*).

3.4. PostgreSQL

O PostgreSQL é um sistema que gerencia bancos de dados com licença de *software* livre da GNU. Segundo Guilherme (pág. 77, 2016) “Banco SQL, é utilizado para armazenar as informações, organizar e categorizar de acordo com seu tipo e contexto”. A interface do PostgreSQL é bastante conhecida por possuir código aberto, isso facilita seu uso e principalmente seu acesso. O PostgreSQL usa a linguagem SQL (*Structured Query Language*) como ferramenta base para a criação dos códigos. Além disso, sua manipulação é simples e tem a capacidade de rodar em diversos sistemas operacionais, propiciando assim, seu crescimento nas últimas décadas.

3.5. Astah Community

De acordo com Brondani, Arend, Souza e Pires:

Astah Community é uma ferramenta gratuita voltada para a modelagem de diagramas UML (Unified Modeling Language). Além do Astah Community, existem outras três versões: Astah UML, Astah Professional e Astah Share que disponibilizam outras funcionalidades além da modelagem UML, porém, sua licença é comercial. A ferramenta Astah Community é conhecida por sua praticidade e simplicidade em elaborar diagramas, como por exemplo: diagramas de classe, caso de uso, sequência, atividade, comunicação, máquina de estado, componentes, implantação, estrutura decomposição, objetos e pacotes. (2013, pág 1)

O Astah Community foi de suma importância para a elaboração do Diagrama de Caso de Uso (Figura 3) e o Diagrama de Classe deste projeto (Figura 4).

4. DESENVOLVIMENTO

4.1. Requisitos

De acordo com Falbo e Barcellos (2011, pág 5 apud SOMMERVILLE, 2007):

As descrições das funções que um sistema deve incorporar e das restrições que devem ser satisfeitas são os requisitos para o sistema. Isto é, os requisitos de um sistema definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar. Em outras palavras, os requisitos definem os serviços que o sistema deve fornecer e dispõem sobre as restrições à operação do mesmo.

Os requisitos podem ser divididos em Funcionais e Não Funcionais. Os Requisitos Funcionais podem indicar o que o sistema deve fazer e como o sistema deve agir em determinadas situações. Já os Requisitos Não Funcionais descrevem as restrições do sistema, com tempo, uso de recursos entre outros (Falbo e Barcellos 2011, pág 5, 6 apud SOMMERVILLE, 2007).

4.1.1. Requisitos Funcionais

Quadro 1: Requisitos Funcionais

Cod.	Nome	Descrição	Prioridade
RF00	Cadastrar Cliente	O dono do estabelecimento poderá efetuar o cadastro do hóspede que ainda não está cadastrado no sistema.	Alta
RF01	Consultar Cliente	Essa função permitirá que o dono do estabelecimento hoteleiro consiga verificar se o cliente já possui cadastro ou não, na pousada, através do seu CPF.	Média
RF02	Editar Cliente	O gestor poderá editar os dados do cliente e efetuar modificações caso haja necessidade.	Média

RF03	Excluir Cliente	O empresário poderá remover o cliente do sistema.	Média
RF04	Reservar Quarto	O funcionário poderá reservar um quarto disponível e conseqüentemente a quantidade de dias que o hóspede deseja permanecer na pousada.	Alta
RF05	Consultar Quarto	O administrador poderá verificar os quartos disponíveis.	Média
RF06	Cancelar Reserva	O responsável por gerenciar as demandas da pousada poderá cancelar a reserva de um quarto que foi solicitado pelo cliente.	Baixo
RF07	Efetuar Pagamento	O gestor poderá confirmar o pagamento do cliente no sistema e conseqüentemente liberar o quarto.	Alta

Fonte: Elaboração Próxima (2019)

4.1.2. Requisitos Não Funcionais

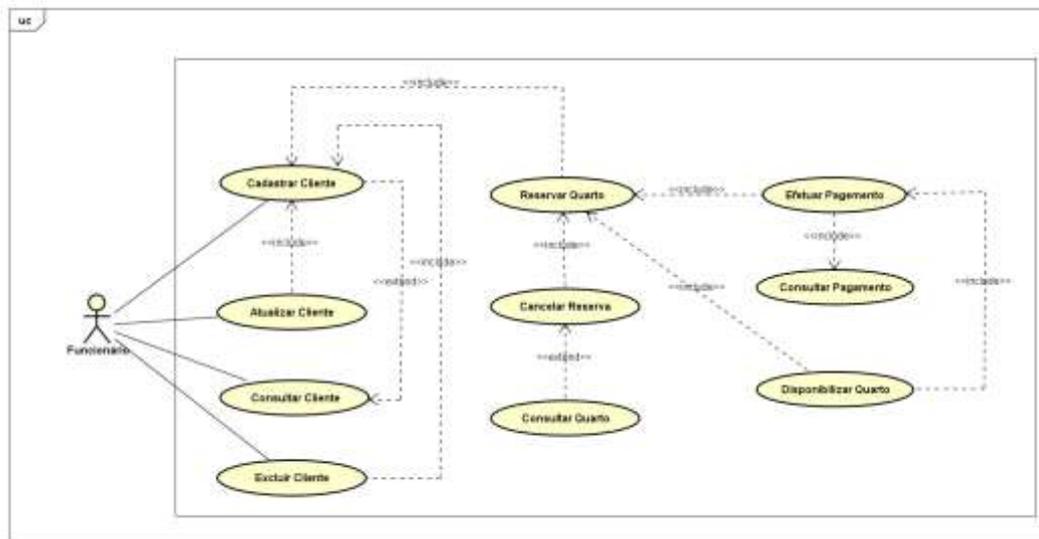
Quadro 2: Requisitos Não Funcionais

Cod.	Nome	Descrição	Prioridade
RNF00	Desempenho	O dono do estabelecimento poderá ter um retorno do sistema com duração de no máximo 5 segundos.	Média
RNF01	Confiabilidade	O <i>software</i> não deve apresentar erros, mas caso presente, os desenvolvedores devem ser notificados para a verificação	Média
RNF02	Usabilidade	O sistema deve ser de fácil uso.	Média

Fonte: Elaboração Próxima (2019)

4.1.3. Diagrama de Caso de Uso do Sistema

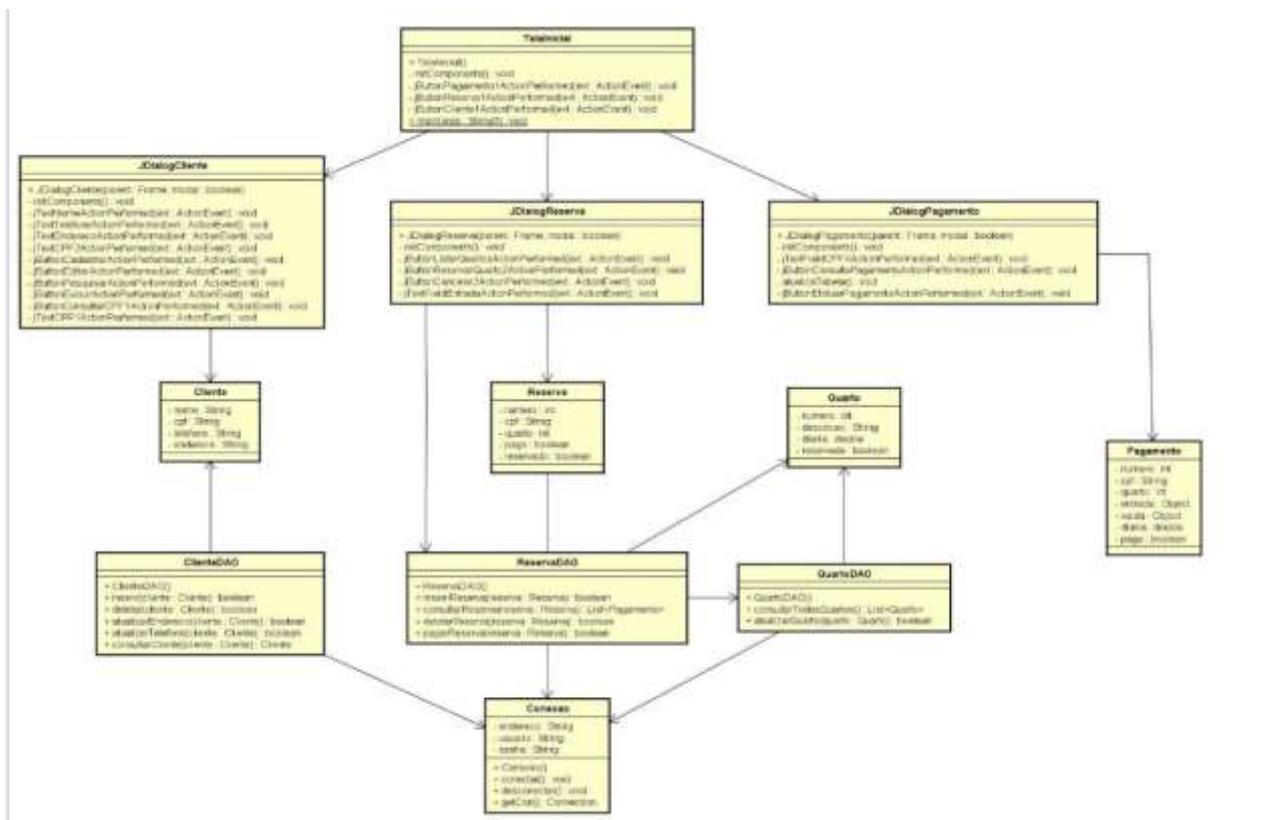
Figura 3: Diagrama de Caso de uso do sistema



Fonte: Elaboração Próxima (2019)

4.1.4. Diagrama de Classe do Sistema

Figura 4: Diagrama de Classe do Sistema



Fonte: Elaboração Próxima (2019)

4.2. Software desenvolvido

Na tela inicial (**Figura 5**) do *software* o usuário poderá acessar as principais funções do sistema. Ao clicar no botão **Cliente** ele é direcionado para uma tela de quatro abas que são responsáveis pelos requisitos relacionados diretamente ao cliente (**Figura 6, Figura 7, Figura 7.1, Figura 8, Figura 9**). Ao clicar no botão **Reserva**, o sistema exibirá uma tela com duas abas relacionadas às reservas de quartos (**Figura 10, Figura 11**). Já ao clicar no botão **Pagamento** o usuário será encaminhado para “Tela de consulta de reserva e efetuação do pagamento” (**Figura 12**).

Figura 5: Tela Inicial



Fonte: Elaboração Própria (2019)

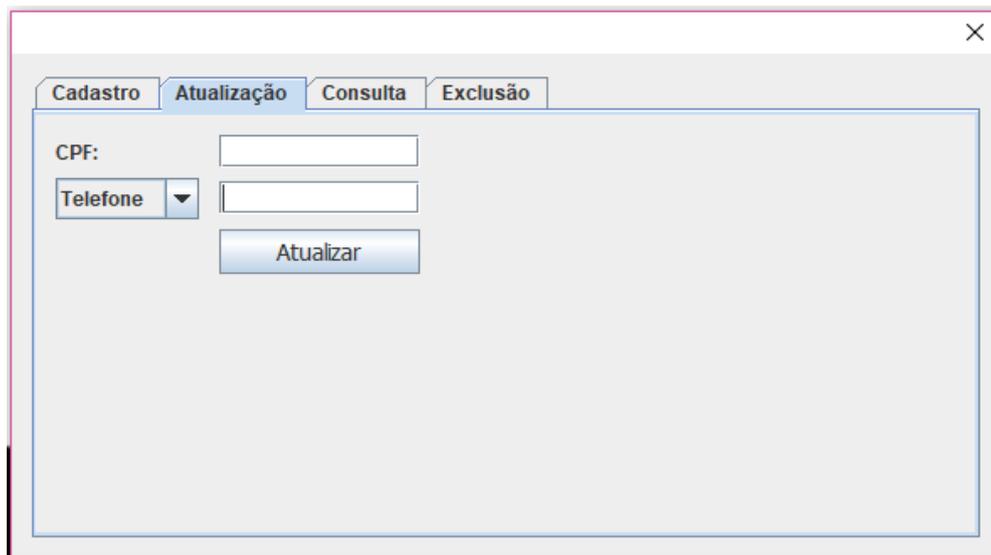
Na aba Cadastro (Figura 6) o usuário poderá realizar o cadastro de novos clientes, informando os dados básicos do cliente.

Figura 6: Tela de Cadastro de Cliente

Fonte: Elaboração Própria (2019)

A aba Atualização (Figura 7, figura 8) é responsável por atualizar tanto o telefone como o endereço do cliente cadastrado. A atualização poderá ser feita através do CPF do cliente.

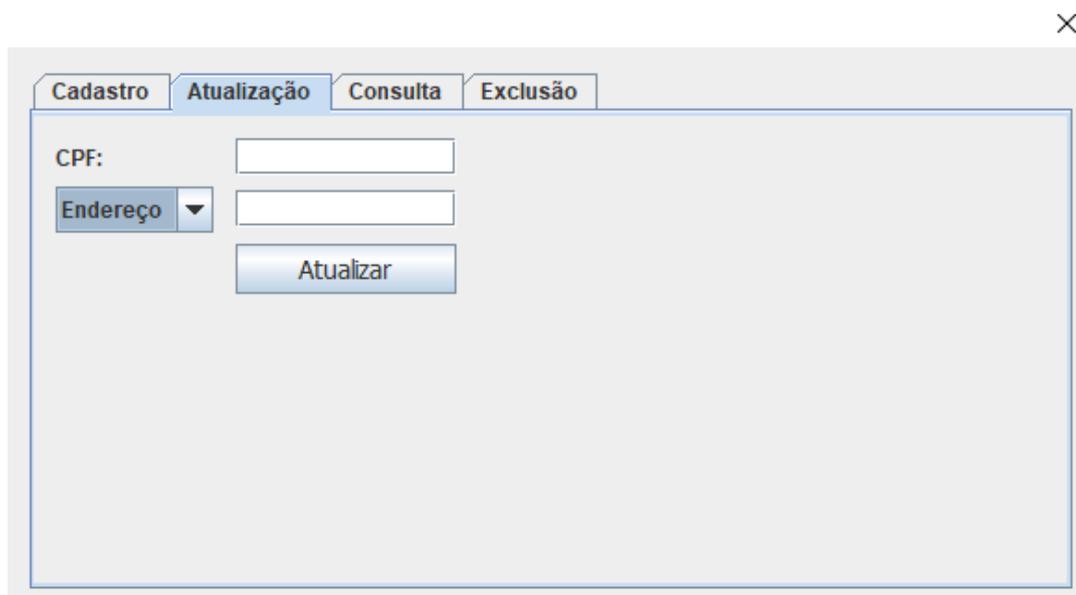
Figura 7: Tela de atualização do telefone



A imagem mostra uma janela de software com uma barra de abas no topo contendo 'Cadastro', 'Atualização', 'Consulta' e 'Exclusão'. A aba 'Atualização' está selecionada. Abaixo das abas, há um formulário com o rótulo 'CPF:' e um campo de entrada de texto. Logo abaixo, há um menu suspenso com o rótulo 'Telefone' e uma seta para baixo, seguido por outro campo de entrada de texto. Abaixo desses campos, há um botão azul com o texto 'Atualizar'.

Fonte: Elaboração Própria (2019)

Figura 8: Tela de atualização do endereço



A imagem mostra uma janela de software com uma barra de abas no topo contendo 'Cadastro', 'Atualização', 'Consulta' e 'Exclusão'. A aba 'Atualização' está selecionada. Abaixo das abas, há um formulário com o rótulo 'CPF:' e um campo de entrada de texto. Logo abaixo, há um menu suspenso com o rótulo 'Endereço' e uma seta para baixo, seguido por outro campo de entrada de texto. Abaixo desses campos, há um botão azul com o texto 'Atualizar'.

Fonte: Elaboração Própria (2019)

Nesta aba o usuário (Figura 9) poderá consultar se um cliente específico está cadastrado no sistema. A consulta poderá ser feita através do CPF.

Figura 9. Tela de Consulta de Clientes

Consultar Cliente:

CPF:

Nome	CPF	Endereço	Telefone
------	-----	----------	----------

Fonte: Elaboração Própria (2019)

Já na última aba (Figura 10) o usuário tem a opção de excluir clientes cadastrados.

Figura 10: Tela de Exclusão de Clientes

CPF:

Nome	CPF	Endereço	Telefone
------	-----	----------	----------

Fonte: Elaboração Própria (2019)

Na aba Cadastrar (Figura 11) é responsável tanto por listar todos os quartos sejam eles reservados ou não, como por realizar a reserva do cliente já cadastrado. Para a realização da reserva é necessário informar o CPF cadastrado e a data de entrada e saída.

Figura 11. Tela de reserva de quarto

Nº do quarto	Descrição	Diária	Reservado
--------------	-----------	--------	-----------

CPF: Entrada: Saída:

Listar
Reservar

Fonte: Elaboração Própria (2019)

Já na aba Excluir (Figura 12) poderá ser feito o cancelamento da reserva. É necessário informar o número do quarto no qual foi reservado.

Figura 12: Tela de cancelamento de reserva

Excluir Reserva:

Número do Quarto:

Cancelar

Fonte: Elaboração Própria (2019)

A última tela (Figura 13) é responsável pela consulta da reserva do cliente, realização do pagamento. Ao clicar no botão Efetuar Pagamento o quarto no qual o cliente estava será liberado para uma nova reserva.

Figura 13. Tela de consulta da reserva e efetuação do pagamento

The screenshot shows a web interface for consulting reservations and making payments. The window title is "Consultar". It contains a search form with a label "CPF:" followed by an input field and a "Consultar" button. Below the form is a table with the following columns: "Nº da Reser...", "CPF do Clie...", "Nº do Quarto", "Entrada", "Saída", "Diária", and "Pago?". At the bottom right of the window, there is a button labeled "Efetuar Pagamento".

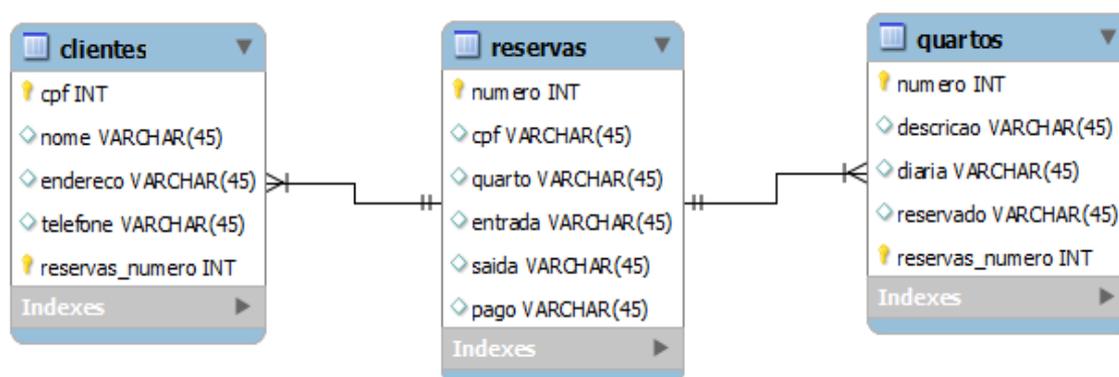
Fonte: Elaboração Própria (2019)

4.3. Banco de Dados Construído

4.3.1. Modelo Relacional

O modelo relacional representa os dados contidos em um banco de dados, essa representação é feita através de uma coleção de tabelas (relações), na qual cada tabela terá um nome, que será único, e um conjunto de atributos com seus respectivos nomes e domínios. (Baptista)

Figura 14: Modelo Relacional



Fonte: Elaboração Própria (2019)

A tabela **clientes** possui os atributos cpf (chave primária), nome (nome do cliente), endereço (endereço do cliente) e telefone (telefone do cliente). Essa tabela se relaciona com a tabela reservas, pois um cliente pode fazer uma ou mais reservas.

Na tabela **reservas** temos como atributos numero (chave primária), cpf (cpf do cliente), quarto (numero do quarto), entrada (entrada do cliente), saida (saida do cliente) e pago (débito do cliente). A tabela reservas se relaciona tanto com a tabela cliente como com a tabela quartos.

Já a tabela **quartos** tem os atributos numero (chave primária), descricao (descrição do quarto), diaria (valor da diária) e reservado (disponibilidade do quarto). A tabela possui um relacionamento com a tabela reservas, uma vez que um quarto pode ser reservado várias vezes.

4.4. Testes do programa

Foram feitos vários testes no programa, a fim de detectar erros para os solucionar. O passo foi iniciar processo de cadastrar um cliente, como pode-se observar na figura (Figura 15) abaixo.

Figura 15. Cadastro de Cliente



Cadastro

Atualização

Consulta

Exclusão

Cadastrar Cliente:

Nome: Myckael Schumacher Fernandes da Silva

CPF: 12360620436

Endereço: Rua Francisco Otaviano Filho, nº 136, centro, Pedra Preta - RN

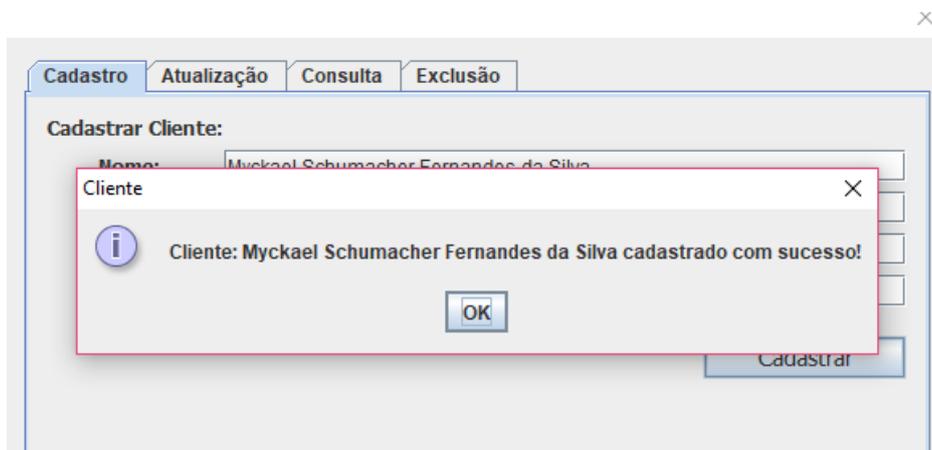
Telefone: (84) 991593349

Cadastrar

Fonte: Elaboração Própria (2019)

Nessa outra figura (Figura 16) o cadastro do cliente é exibido com uma mensagem de sucesso.

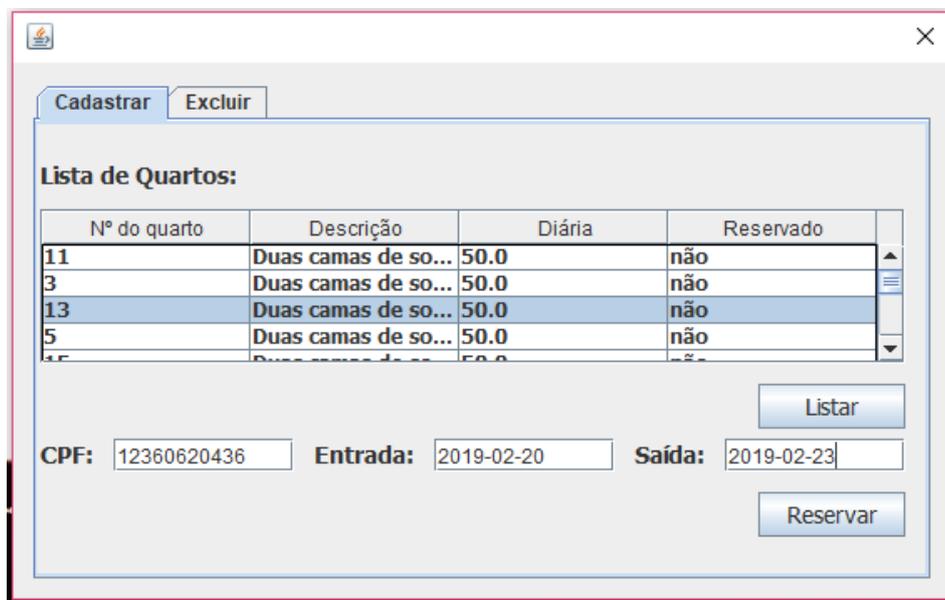
Figura 16. Resposta do cadastro



Fonte: Elaboração Própria (2019)

Na tela Cadastro de Reservas (Figura 17) foi feita a listagem dos quartos disponíveis e escolha do quarto desejado. Antes de finalizar a reserva, o cliente informou o CPF e a data de entrada e saída.

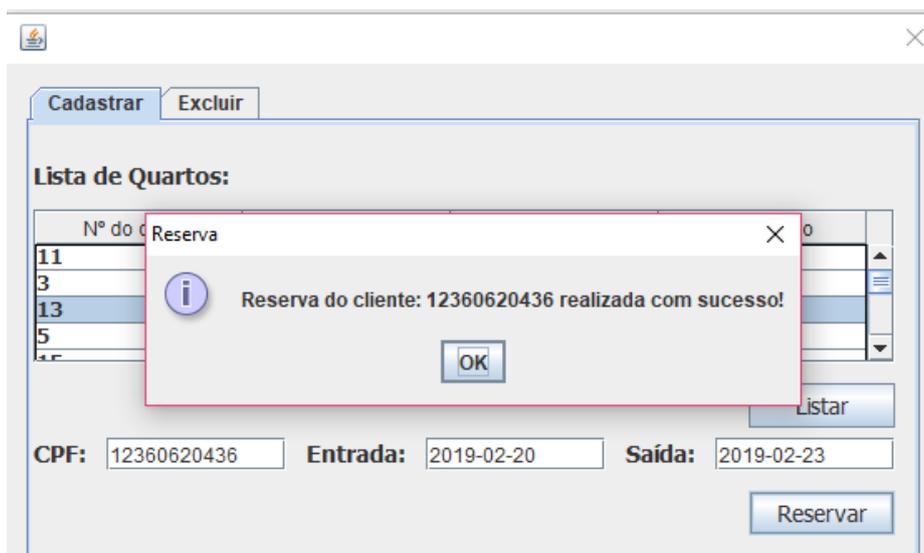
Figura 17: Cadastro de Reservas



Fonte: Elaboração Própria (2019)

Já nessa tela (Figura 18) é apresentado uma mensagem de sucesso na finalização do pedido de reserva.

Figura 18: Resposta do cadastro da reserva



Fonte: Elaboração Própria (2019)

Tendo realizado os principais testes do programa, foi possível notar o bom funcionamento do sistema, como também a facilidade em seu uso. O testes não apresentarem erros e atendeu aos requisitos

5. CONCLUSÃO

Durante o desenvolvimento desse trabalho foi notório a importância dos avanços tecnológicos para a nossa sociedade. Destacamos nesse projeto a importância da tecnologia para o sistema hoteleiro.

Sendo assim, propomos um *software*, que solucionasse alguns dos problemas em relação ao gerenciamento dos serviços ofertados na Pousada Cabugi. A fim de tornar os serviços automatizados e conseqüentemente mais simples. Além disso, fazer com que a pousada se destacasse também no mercado hoteleiro da região.

O *software* foi elaborado, desenvolvido e implementado baseado nos casos de uso de maior importância para a Pousada, sendo eles: Cadastro de Clientes, Reserva de Quartos, Realização de Pagamento e Disponibilização de Quartos.

Esse projeto foi de total importância para o nosso enriquecimento profissional. Além disso, foi possível colocar em prática os conhecimentos obtidos ao longo do curso, que serviram de base para a produção do sistema, e tivemos a oportunidade de conhecer melhor sobre a área de desenvolvimento de sistemas, proporcionando assim a solução dos problemas encontrados nos serviços da pousada.

Portanto, é certo afirmar que a experiência obtida no decorrer do desenvolvimento do sistema, teve grande impacto em nossas vidas, tanto pessoal como profissional, uma vez que nos permitiu engrandecer o nosso campo de conhecimento e contribuir com o desenvolvimento da nossa sociedade.

5.1. Trabalhos Futuros

Futuramente o *software* poderá ser aperfeiçoado para uma melhor utilização. Algumas evoluções que serão feitas são: aperfeiçoamento da interface gráfica, desenvolvimento em relação à segurança de dados, aprimoramento do requisito **Efetuar Pagamento**, modificação do padrão utilizado nas datas de **entrada** e **saída** ao realizar a reserva, acrescentar a emissão de relatórios diários, semanais ou mensais e a criação de sistema web para a divulgação da empresa.

6. REFERÊNCIAS BIBLIOGRÁFICAS

SOFTWARE - Definição. In: MARÇULA, Marcelo; BENINI FILHO, Pio Armando. **Informática Conceitos e Aplicações**. 4 rev.. ed. São Paulo: Érica, 2014. cap. 13, p. 154.

LINGUAGENS de Programação. In: MARÇULA, Marcelo; BENINI FILHO, Pio Armando. **Informática Conceitos e Aplicações**. 4 rev.. ed. São Paulo: Érica, 2014. cap. 15, p. 169.

ORGANIZAÇÃO e Armazenamento de Dados. In: MARÇULA, Marcelo; BENINI FILHO, Pio Armando. **Informática Conceitos e Aplicações**. 4 rev.. ed. São Paulo: Érica, 2014. cap. 16, p. 175.

GUILHERME, Igor Roberto. **Sistema para submissão de artigos**. Disponível em: <<http://159.203.166.88/index.php/interfacetecnologica/article/view/103/101>> Acesso em: 15 de março de 2018.

ORACLE (Ed.). **Trilha do Aprendizado da Plataforma NetBeans: O que é a Plataforma NetBeans?**. 2018. Disponível em: <https://netbeans.org/kb/trails/platform_pt_BR.html>. Acesso em: 10 abr. 2018.

PACIEVITCH, Yuri. **MySQL**. Disponível em: <<https://www.infoescola.com/informatica/mysql/>>. Acesso em: 10 abr. 2018.

ASTAH. **Sobre o Astah**. Disponível em: <<http://astah.net/about-us>>. Acesso em: 23 fev. 2019.

CACHOEIRA et al. **Software de Gestão Hoteleira: como ele pode ajudar o seu negócio?** 2017. Disponível em: <https://blog.hmax.com.br/software-de-gestao-hoteleira-como-ele-pode-ajudar-o-seu-negocio/?utm_source=blog&utm_campaign=rc_blogpost>. Acesso em: 20 jan. 2019.

DEVMEDIA. **Guia Completo de SQL**. Disponível em: <<https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>>. Acesso em: 14 fev. 2019.

PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER. Bebedouro – Sp: Centro Universitário Unifafibe, v. 1, 2011. Disponível em: <<https://www.inf.ufes.br/~monalessa/PaginaMonalessa-NEMO/ES/NotasDeAula-EngSoftware-EngComp-Parte-II.pdf>>. Acesso em: 15 fev. 2019.

BAPTISTELLA, Adriano José et al. **Abordando a arquitetura MVC, e Design Patterns: Observer, Composite, Strategy.** Disponível em: <<http://www.linhadecodigo.com.br/artigo/2367/abordando-a-arquitetura-mvc-e-design-patterns-observer-composite-strategy.aspx>>. Acesso em: 07 fev. 2019.

VENTURA, Plínio et al. Entendendo o Diagrama de Classes da UML: Modelo de Classes com UML. 2018. Disponível em: <<https://www.ateomomento.com.br/uml-diagrama-de-classes/>>. Acesso em: 23 jan. 2019.

INDRUSIAK, Leandro Soares et al. **Linguagem Java.** Porto Alegre: Grupo Javars Jug Rio Grande do Sul, 1996. Disponível em: <<http://www.cin.ufpe.br/~arfs/introjjava.pdf>>. Acesso em: 28 jan. 2019.

FRÓES, Gabriel et al. **Entendendo a arquitetura MVC de forma simples:** Entendendo de forma simplificada o conceito por trás do MVC. 2014. Disponível em: <<https://www.codigofonte.com.br/artigos/aprenda-a-arquitetura-mvc>>. Acesso em: 20 jan. 2019.

TUCKER, Allen B.; NOONAN, Robert E.. **Linguagens de Programação: Princípios e Paradigmas.** 2. ed. São Paulo: Mcgraw-hill Interamericana do Brasil - Ltda., 2009. Disponível em <https://pt.scribd.com/document/339519406/Linguagens-de-Programacao-Principios-e-Paradigmas-2%C2%AA-Ed>>. Acesso em: 21 fev. 2019.

CACHOEIRA et al. **Software de Gestão Hoteleira: como ele pode ajudar o seu negócio?** 2017. Disponível em: <https://blog.hmax.com.br/software-de-gestao-hoteleira-como-ele-pode-ajudar-o-seu-negocio/?utm_source=blog&utm_campaign=rc_blogpost>. Acesso em: 12 nov. 2018.

SEBRAE et al. **GOSTAR DE HOSPEDAR NÃO É O BASTANTE PARA EMPREENDER EM HOTELARIA: PERFIL DE NEGÓCIOS DE HOTELARIA (HOTÉIS E Pousadas).** Recife: Sebrae, 2014. 45 p. Disponível em:

<http://www.sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/PE/Anexos/Perfil%20de%20negocios_hotelaria_.pdf>. Acesso em: 12 nov. 2018.

REZENDE, Denis Alcides et al. **Engenharia de Software e Sistemas de Informação**. 3. ed. Rio de Janeiro: Brasport Livros e Multimídia Ltda., 2005. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=rtBvl_L-1mcC&oi=fnd&pg=PT23&dq=Engenharia+de+Software+defini%C3%A7%C3%A3o&ots=9Afj_J2q3t&sig=jd-apHE8VJN6nwyw08oTJj6wS9Y#v=onepage&q&f=false>. Acesso em: 20 jan. 2019.

BRONDANI, Camila Hübner; AREND, Cesar Frantz; SOUZA, Darciele Aparecida Zilio; PIRESDE, José Carlos Puiati. **TUTORIAL: Guia Prático de utilização da ferramenta Astah Community 6.1**. Santa Maria: Pet Informática Ufsm, 2013. Disponível em: <<https://pt.scribd.com/doc/139768773/Astah-Community#user-util-view-profile>>. Acesso em: 25 jan. 2019.

FALBO, Ricardo de Almeida; BARCELLOS, Monalessa Perini. **Engenharia de Software**. Espírito Santo: Ufes - Universidade Federal do Espírito Santo, 2011. Disponível em: <<https://www.inf.ufes.br/~monalessa/PaginaMonalessa-NEMO/ES/NotasDeAula-EngSoftware-EngComp-Parte-II.pdf>>. Acesso em: 25 jan. 2019.

BAPTISTA, Cláudio de Souza et al. Banco de Dados: Modelo Relacional. In: BAPTISTA, Cláudio de Souza et al. **Banco de Dados**. Paraíba: Ufpb/cct, [1989]. Cap. 2. p. 1-44. Disponível em: <<http://www.dsc.ufcg.edu.br/~baptista/cursos/BDadosI/Capitulo22.pdf>>. Acesso em: 05 fev. 2019.

MANSSOUR, Isabel Harb et al. **Paradigmas de Linguagem I: Paradigma Orientado a Objetos**. [s.l.]: [s.n.], [1996]. Disponível em: <<https://www.inf.pucrs.br/~gustavo/disciplinas/pli/material/paradigmas-aula12.pdf>>. Acesso em: 15 fev. 2019.