

ANNA JULIA LIMA DE SOUSA SP3024016
GUILHERME AKIO MIURA SP3120791
GUILHERME BITTENCOURT SCHMIDT SP313640X
KELLY RADCHELLE ARAUJO DE SOUZA SP3123588
RAFAEL TEIXEIRA FONSECA SP3126919
RICARDO CARRIEL DE OLIVEIRA FILHO SP3136728

Pousada Chalés Água de Coco

São Paulo - SP - Brasil

2025

ANNA JULIA LIMA DE SOUSA SP3024016
GUILHERME AKIO MIURA SP3120791
GUILHERME BITTENCOURT SCHMIDT SP313640X
KELLY RADCHELLE ARAUJO DE SOUZA SP3123588
RAFAEL TEIXEIRA FONSECA SP3126919
RICARDO CARRIEL DE OLIVEIRA FILHO SP3136728

Pousada Chalés Água de Coco

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo Câmpus São Paulo, como requisito parcial para conclusão do curso Tecnologia em Análise e Desenvolvimento de Sistemas \LaTeX .

IFSP - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Tecnologia em Análise e Desenvolvimento de Sistemas

Orientador: Marcelo Tavares de Santana

São Paulo - SP - Brasil

2025

Resumo

Este Projeto de Conclusão de Curso tem como objetivo o desenvolvimento de um sistema web para automatizar os processos administrativos da pousada Chalés Água de Coco, que atualmente realiza a gestão de hóspedes, reservas, acomodações e controle financeiro por meio de planilhas eletrônicas no Excel. A falta de integração e a limitação desse método tornam a operação vulnerável a erros, retrabalho e dificuldade de acesso remoto às informações. Com base em uma parceria estabelecida com a pousada, foi possível realizar um levantamento detalhado dos requisitos e desenvolver uma solução personalizada, capaz de centralizar as informações em uma única plataforma, acessível via internet. O sistema propõe melhorias significativas na organização dos dados, no controle de reservas e na geração de relatórios gerenciais, otimizando a tomada de decisões. Este projeto representa a aplicação prática dos conhecimentos adquiridos ao longo do curso, ao mesmo tempo em que oferece uma ferramenta útil para a modernização da gestão em pequenos empreendimentos do setor de hospitalidade.

Palavras-chave: sistema web, pousada, automação, reservas, gestão de hóspedes, controle financeiro.

Abstract

This Final Paper aims to develop a web-based system to automate the administrative processes of the inn Chalés Água de Coco, which currently manages guests, reservations, accommodations, and financial control through Excel spreadsheets. The lack of integration and limitations of this manual method make operations prone to errors, rework, and hinder remote access to information. Based on a partnership established with the inn, it was possible to conduct a detailed requirements analysis and develop a customized solution capable of centralizing data on a single, internet-accessible platform. The system brings significant improvements in data organization, reservation management, and the generation of management reports, optimizing decision-making. This project represents the practical application of the knowledge acquired, while also delivering a useful tool to modernize management practices in small hospitality businesses.

Keywords: web system, inn, automation, reservations, guest management, financial control.

Lista de ilustrações

Figura 1 – Diagrama de Implantação desenvolvido no Online Visual-Paradigm .	34
Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm .	35
Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo .	41
Figura 4 – Diagrama Entidade-Relacionamento (MER) desenvolvido no brModelo	42
Figura 5 – Comparação dos custos e receitas acumuladas no cenário otimista . .	45
Figura 6 – Comparação dos custos e receitas acumuladas no cenário pessimista .	45
Figura 7 – Comparação dos custos e receitas acumuladas no cenário realista . .	46

Lista de quadros

Quadro 1 – Comparativo de funcionalidades entre Chalés Água De Coco e seus concorrentes	19
Quadro 2 – Função e Responsabilidades da Equipe do Projeto	22
Quadro 3 – Product backlog	24
Quadro 4 – Requisitos Funcionais - Parte 1	26
Quadro 5 – Requisitos Funcionais - Parte 2	27
Quadro 6 – Requisitos Funcionais - Parte 3	28
Quadro 7 – Requisitos Funcionais - Parte 4	29
Quadro 8 – Requisitos Não Funcionais - Parte 1	30
Quadro 9 – Requisitos Não Funcionais - Parte 2	31
Quadro 10 – Regras de Negócio	32

Lista de tabelas

Tabela 1 – Função dos Integrantes da Equipe	23
Tabela 2 – Resumo dos custos de Equipamentos e Serviços	43
Tabela 3 – Resumo dos custos de Infraestrutura	43
Tabela 4 – Quantidade e horas trabalhadas por função	44
Tabela 5 – Custos por função	44
Tabela 6 – Custo Total por Categoria	44

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
abnTeX	ABsurdas Normas para TeX

Sumário

1	INTRODUÇÃO	17
1.1	Objetivo	17
1.2	Justificativa	17
1.3	Análise da Concorrência	18
1.3.1	SimplesHotel	18
1.3.2	HospedaJá	18
1.3.3	Comparativo	19
2	REVISÃO DE LITERATURA	20
2.1		20
3	GESTÃO DO PROJETO	21
3.1	Organização da Equipe	21
3.1.1	Funções e Responsabilidades	21
3.2	Metodologias de gestão e desenvolvimento	21
3.2.1	Time Scrum	22
3.3	Artefatos	23
3.3.1	Product Backlog	23
4	DESENVOLVIMENTO DO PROJETO	26
4.1	Escopo do Projeto	26
4.1.1	Requisitos Funcionais	26
4.1.2	Requisitos Não Funcionais	30
4.1.3	Regras de Negócio	32
4.2	História de Usuário	33
4.2.1	Descrição	33
4.3	Arquitetura	33
4.3.1	Desenho da Arquitetura	33
4.3.2	Diagrama da Arquitetura	34
4.3.2.1	Diagrama de Implantação	34
4.3.2.2	Diagrama de Componentes	35
4.4	Tecnologias	35
4.4.1	Front-End	35
4.4.2	Back-End	36
4.4.3	Banco de Dados	36
4.4.4	Infraestrutura	36

4.5	Ferramentas de Apoio	37
4.5.1	GitHub	37
4.5.2	BRModelo	37
4.5.3	Visual Paradigm Online	37
4.5.4	Latex	38
4.5.5	Google Meet	38
4.6	Manutenibilidade	38
4.7	Segurança, Privacidade e Legislação	39
4.7.1	Conformidade com a LGPD	40
4.7.2	Protocolos de Segurança na Comunicação	40
4.7.2.1	Certificação Digital com SSL/TLS	40
4.7.2.2	Protocolo HTTPS	40
4.7.2.3	Compatibilidade com a Arquitetura Django	41
4.8	Modelagem do Banco de Dados	41
4.8.1	Modelo Entidade-Relacionamento - MER	41
4.8.2	Diagrama Entidade-Relacionamento - DER	41
4.9	Entregas	42
5	VIABILIDADE FINANCEIRA	43
5.1	Custos	43
5.1.1	Custo Estrutural	43
5.1.2	Custo Mão de Obra	44
5.1.3	Custo Total	44
5.2	Cenários	45
5.2.1	Cenário Otimista	45
5.2.2	Cenário Pessimista	45
5.2.3	Cenário Realista	46
6	CONSIDERAÇÕES FINAIS	47
7	CONCLUSÃO	48
	REFERÊNCIAS	49
	APÊNDICES	51
	APÊNDICE A – QUISQUE LIBERO JUSTO	52

APÊNDICE B – NULLAM ELEMENTUM URNA VEL IMPER- DIET SODALES ELIT IPSUM PHARETRA LI- GULA AC PRETIUM ANTE JUSTO A NULLA CURABITUR TRISTIQUE ARCU EU METUS .	53
ANEXOS	55
ANEXO A – MORBI ULTRICES RUTRUM LOREM.	56
ANEXO B – CRAS NON URNA SED FEUGIAT CUM SOCIIS NATOQUE PENATIBUS ET MAGNIS DIS PARTU- RIENT MONTES NASCETUR RIDICULUS MUS .	57
ANEXO C – FUSCE FACILISIS LACINIA DUI	58

1 Introdução

A transformação digital tem impactado significativamente a forma como empresas de diversos segmentos gerenciam suas atividades operacionais e estratégicas. No setor de hospitalidade, especialmente em pequenos empreendimentos como pousadas, a adoção de tecnologias adequadas pode representar um grande avanço em eficiência, organização e qualidade no atendimento ao cliente.

Apesar disso, muitas pousadas ainda utilizam métodos manuais ou ferramentas limitadas, como planilhas eletrônicas, para controlar reservas, hospedagens e finanças. Esse é o caso da pousada Chalés Água de Coco, que realiza a gestão de suas operações exclusivamente por meio do Excel. Tal prática, embora inicialmente funcional, apresenta limitações consideráveis, como risco elevado de erros, dificuldade de atualização em tempo real e ausência de acessibilidade remota.

Diante dessa realidade, identificou-se a necessidade de modernização e automação dos processos da pousada, visando torná-los mais ágeis, seguros e organizados. A parceria firmada com a Chalés Água de Coco permitiu levantar as principais dificuldades enfrentadas na gestão atual, servindo como base para o desenvolvimento de uma solução tecnológica alinhada às reais necessidades do negócio.

1.1 Objetivo

Desenvolver um sistema web para automatizar os processos de gestão da pousada Chalés Água de Coco, substituindo o controle manual realizado via planilhas do Excel. O sistema permitirá o gerenciamento eficiente de hóspedes, reservas, acomodações e informações financeiras, promovendo maior organização, redução de falhas e facilidade de acesso às informações por parte dos gestores.

1.2 Justificativa

A escolha deste projeto se justifica pela necessidade real de modernização enfrentada por pequenos empreendimentos do setor de hospedagem, como a pousada Chalés Água de Coco, que atualmente depende de controles manuais realizados por meio de planilhas no Excel. Esse tipo de gestão, embora comum em pequenos negócios, apresenta diversas limitações, como a suscetibilidade a erros humanos, dificuldade de atualização simultânea, falta de integração entre os dados e ausência de acessibilidade remota.

Com o crescimento da demanda por eficiência operacional e qualidade no atendimento ao cliente, torna-se essencial a adoção de soluções tecnológicas que automatizem processos, centralizem informações e proporcionem maior controle gerencial. Um sistema web desenvolvido sob medida representa uma alternativa viável e eficaz, oferecendo funcionalidades específicas para o contexto da pousada, além de ser acessível a partir de qualquer dispositivo conectado à internet.

Além disso, o desenvolvimento deste projeto contribui academicamente ao proporcionar a aplicação prática dos conhecimentos adquiridos ao longo do curso de Tecnologia da Informação, abrangendo áreas como análise de requisitos, modelagem de dados, programação web, experiência do usuário e segurança da informação. Por fim, a solução proposta tem potencial de gerar impacto direto e positivo na gestão do negócio parceiro, tornando este trabalho relevante tanto do ponto de vista acadêmico quanto social e econômico.

1.3 Análise da Concorrência

1.3.1 SimpleHotel

SimpleHotel é um sistema de gestão hoteleira desenvolvido para hotéis e pousadas de pequeno a grande porte, com foco na automação e integração dos processos operacionais e administrativos do setor de hospitalidade. A plataforma monetiza por meio de cobrança por planos pré-pagos mensais, com variação conforme o número de acomodações. Taxas adicionais são aplicadas para uso de módulos como emissão de NFSe, envio de SMS e motor de reservas. Suas principais funcionalidades são gerenciador de reservas, controle financeiro e de estoque, gerenciamento de eventos e day use, integração com canais de venda (Booking, Expedia, Hoteis.com, Decolar), web check-in, emissão de notas fiscais, módulo de governança e suporte multicanal. A plataforma é uma aplicação web, acessível via navegador. Armazenamento em nuvem e sincronização em tempo real com OTAs. Suporte remoto com atendimento humanizado.

1.3.2 HospedaJá

HospedaJá é uma plataforma online de gestão para hotéis e pousadas de pequeno e médio porte, com foco na organização de reservas, hospedagens, acomodações e finanças. A plataforma monetiza por meio de cobrança de mensalidades conforme plano escolhido, variando por número de quartos e usuários. Suas principais funcionalidades são controle de reservas e hospedagens, mapa de ocupação, gestão financeira, controle de estoque de produtos e serviços, geração de relatórios gerenciais. A plataforma é uma aplicação web baseada em nuvem, que contém um sistema de backup automático, faz suporte técnico via e-mail e chamados.

1.3.3 Comparativo

Funcionalidades	Chalés	simpleshotel	hospedajá
Gerenciamento de Quartos	X		X
Gerenciamento de Hospedes	X	X	X
Gerenciamento de Reservas	X	X	X
Controle Financeiro	X	X	X
Relatórios	X		X
Controle de Estoque		X	X
Baixo custo	X		

Fonte: Elaborado pelos Autores.

Quadro 1: Comparativo de funcionalidades entre Chalés Água De Coco e seus concorrentes

2 Revisão de Literatura

2.1

3 Gestão do Projeto

Esta seção detalha as principais estratégias empregadas no desenvolvimento da aplicação web de gestão para a pousada Chalés Água de Coco. Dessa forma, nela são apresentadas as metodologias, ferramentas e práticas que foram adotadas para o planejamento, execução e monitoramento do projeto, com o objetivo de garantir uma entrega organizada, eficiente e alinhada aos objetivos estabelecidos pelas partes interessadas.

3.1 Organização da Equipe

Na gestão desse projeto, a organização da equipe representou um marco fundamental e teve como foco dividir as funções e atividades necessárias para o desenvolvimento da aplicação web de gestão da pousada.

3.1.1 Funções e Responsabilidades

A designação das funções e responsabilidades foi realizada de maneira estratégica e levou em consideração as competências técnicas e experiências prévias de cada membro da equipe, visando a entrega do produto final dentro do prazo estipulado. A função de cada membro e suas respectivas responsabilidades estão detalhadas no quadro "2": Função e Responsabilidades da Equipe do Projeto.

3.2 Metodologias de gestão e desenvolvimento

Para assegurar que o produto final seja entregue em pleno alinhamento com as expectativas da cliente, a equipe envolvida no projeto optou pela adoção da metodologia ágil Scrum como ferramenta de gestão e desenvolvimento do projeto. Essa decisão fundamentou-se na familiaridade da equipe com a estrutura, na capacidade do Scrum de otimizar a organização, divisão e planejamento de atividades do projeto, e em sua relevância como framework de gerenciamento— dado seu uso extensivo no contexto de desenvolvimento de softwares complexos. Segundo Schwaber e Sutherland (2020), o Scrum é um framework estruturado desenvolvido na década de 1990. Ele foi criado com o intuito de auxiliar equipes na criação e gerenciamento de produtos complexos. Para isso o Scrum tem como pilares fundamentais a transparência, a inspeção e a adaptação. A transparência garante que todos os aspectos significativos do processo estejam visíveis e claros para as partes interessadas a todo momento. A inspeção envolve o acompanhamento regular dos artefatos e progresso, a fim da detecção precoce de problemas. Por fim, a adaptação refere-se à capacidade de

Integrante	Função	Responsabilidades
Anna Julia	Analista de Cronograma	Criar e atuar na manutenção e monitoramento do cronograma do projeto, além de oferecer suporte nas práticas de gestão
Guilherme Akio	Engenheiro de Dados e Administrador de Banco de Dados (DBA)	Implementar, administrar e otimizar o banco de dados da aplicação, garantindo a integridade, segurança e performance dos dados
Guilherme Bittencourt	Documentador Técnico e Engenheiro de Software	Criar e manter a documentação técnica, garantido a clareza e acessibilidade das informações do projeto. Além de definir e organizar a estrutura da aplicação.
Kelly Radchelle	Gerente de Projeto	Coordenar a equipe e gerenciar as atividades do projeto, a fim de facilitar as tomadas de decisões e assegurar a comunicação entre as partes interessadas do projeto
Rafael Teixeira	Desenvolvedor Frontend e UI/UX Designer	Criar e implementar a interface da aplicação, garantindo uma boa experiência de usuário (UX) e design de interface (UI), além de desenvolver a lógica de apresentação.
Ricardo Carriel	Desenvolvedor Backend e Administrador de Servidores	Desenvolver a lógica de negócio da aplicação, configurar e administrar o servidor de aplicação além de garantir a integração da aplicação

Fonte: Elaborado pelos autores.

Quadro 2: Função e Responsabilidades da Equipe do Projeto

fazer ajustes no processo em resposta aos problemas anteriormente detectados na inspeção, com o objetivo de otimizar os resultados. Sabendo que para operacionalizar esses pilares e assegurar um ciclo de desenvolvimento iterativo e incremental, a metodologia Scrum sugere a definição de papéis específicos dentro do time, estabelece a realização de uma sequência de eventos formais e o uso de artefatos específicos, os integrantes da equipe desenvolveram as tarefas e eventos sugeridos pelo Scrum para a gestão e desenvolvimento do projeto.

3.2.1 Time Scrum

Como estabelecido no Guia do Scrum (SCHWABER; SUTHERLAND, 2020), os membros da equipe do projeto assumem papéis específicos que compõem um time Scrum: Dono do Produto (Product Owner), Scrum Master e Time de Desenvolvimento. O Dono do Produto (Product Owner) é o representante das partes interessadas (Stakeholders) e tem como responsabilidade principal gerenciar o backlog do Produto, a fim de maximizar

o valor do produto e otimizar o trabalho do Time de Desenvolvimento. O Scrum Master é o responsável por promover e facilitar a aplicação da teoria e das práticas do framework, atuando como um líder-servidor ao auxiliar a equipe na retirada de impedimentos que venham afetar seu progresso. O Time de Desenvolvimento é responsável por entregar um incremento (versão potencialmente usável do produto) ao final de cada sprint, atuando de maneira auto-organizada e multifuncional. Ciente disso, realizou-se a distribuição dos papéis de um time scrum entre os integrantes da equipe e registrou-se na “Tabela 1”.

Tabela 1 – Função dos Integrantes da Equipe

Integrante	Função		
	<i>Product Owner</i>	<i>Scrum Master</i>	<i>Time de Desenvolvimento</i>
Anna Julia	X		
Guilherme Akio			X
Guilherme Bittencourt			X
Kelly Radchelle	X		
Rafael Teixeira			X
Ricardo Carriel			X

Fonte: Elaborado pelos autores.

3.3 Artefatos

No scrum, os artefatos são elementos fundamentais que ajudam a equipe a consolidar a transparência no processo de desenvolvimento. Cientes das suas importâncias, o time scrum realizou o planejamento inicial dos artefatos: Product Backlog e Sprints Backlog.

3.3.1 Product Backlog

O Product Backlog consiste em uma lista ordenada de todos os itens de trabalho, sendo elas as funcionalidades, os requisitos e os aprimoramentos necessários para a produção do produto, e que oferecem o máximo valor e utilidade para o cliente. Diante disso, foi elaborado pelo product owner o backlog de produto inicial do projeto da aplicação web de gestão da pousada Chalés Água de Coco (??) com base nas histórias de usuário levantadas pela equipe, visto que, elas expressam as necessidades e expectativas sob a perspectiva da usuária e principal stakeholder.

Ordem	Item	Domínio	Prioridade
1	Configurar ambiente de hospedagem	Infraestrutura e Configuração	Essencial
2	Configurar repositório Git e fluxo de versionamento	Infraestrutura e Configuração	Essencial
3
4	40	65	...
5	40	65	...
6	40	65	...
7	40	65	...
8	40	65	...
9	40	65	...
10	40	65	...

Fonte: Autor.

Quadro 3: Product backlog

4 Desenvolvimento do Projeto

4.1 Escopo do Projeto

4.1.1 Requisitos Funcionais

Código	Descrição	Prioridade	Regra de Negócio
RF01	O sistema deve permitir o cadastro de reservas, associando um quarto a um período (data de check-in e check-out)	Alta	Não Aplicável
RF02	O sistema deve permitir o cadastro de reservas, associando um quarto a um período (data de check-in e check-out)	Alta	RN03, RN01
RF03	Uma reserva deve estar associada a um quarto disponível para que ela seja cadastrada.	Alta	RN11
RF04	O sistema deve exigir os dados pessoais do hóspede para que a reserva seja cadastrada: nome completo, endereço completo, CPF, telefone e e-mail.	Alta	RN05
RF05	A proprietária deve conseguir reservar quartos para um cliente em nome de outra pessoa responsável, registrando os dados do hóspede e, opcionalmente, do responsável.	Média	RN06
RF06	O sistema deve exigir o pagamento de 50 por cento do valor da estadia para confirmar o cadastro da reserva (a ser pago no momento da reserva ou em um prazo definido).	Média	RN01
RF07	O sistema deve permitir o registro da comprovação do pagamento	Média	RN01

Fonte: Elaborado pelos autores.

Código	Descrição	Prioridade	Regra de Negócio
RF08	A proprietária deve conseguir cancelar ou remarcar uma reserva, com possível registro do motivo	Média	RN02
RF09	A proprietária deve poder visualizar todas as reservas, com detalhes do hóspede, quarto reservado e período	Alta	RN15
RF10	A proprietária deve conseguir cadastrar mais de uma reserva no nome de um mesmo cliente	Média	RN04
RF11	A proprietária deve conseguir reservar um mesmo quarto para diferentes clientes em datas seguidas respeitando os horários de check-in e check-out configurados para quarto	Alta	RN11
RF12	A proprietária deve conseguir acessar o histórico de reservas de um cliente	Média	RN15
RF13	O sistema deve mudar o status do quarto para ocupado após a realização do check-in	Alta	Não Aplicável
RF14	O sistema mandar deve uma notificação para o hospede após a confirmação da reserva	Baixa	Não Aplicável

Fonte: Elaborado pelos autores.

Quadro 5: Requisitos Funcionais - Parte 2

Código	Descrição	Prioridade	Regra de Negócio
RF15	O sistema deve permitir o cadastro de novos quartos, incluindo informações como número/nome do quarto, capacidade (número de hóspedes) tipo (ex: chale, simples solteiro, simples casal, etc.), e preço por noite.	Alta	Não Aplicável
RF16	A proprietária deve conseguir editar as informações dos quartos já cadastrados	Alta	Não Aplicável
RF17	A proprietária deve poder visualizar todos os quartos cadastrados.	Alta	Não Aplicável
RF18	O sistema deve permitir a visualização dos quartos disponíveis no período de tempo selecionado para a reserva	Média	RN03, RN11
RF19	A proprietária deve conseguir mudar o status de um quarto (ex: disponível, indisponível, em manutenção).	Alta	RN10

Fonte: Elaborado pelos autores.

Quadro 6: Requisitos Funcionais - Parte 3

Código	Descrição	Prioridade	Regra de Negócio
RF20	A proprietária deve poder registrar as despesas da pousada, categorizando-as (ex manutenção, limpeza, contas de consumo), especificando a data, o valor, a categoria e uma descrição da despesa	Média	Não Aplicável
RF21	A proprietária deve conseguir cadastrar gastos fixos e gastos variáveis.	Média	Não Aplicável
RF22	A proprietária deve poder registrar receitas, associando-as a uma reserva ou a outras fontes de receita, especificando a data, o valor e uma descrição da receita.	Média	RN13
RF23	O sistema deve permitir que a proprietária visualize todas as transações financeiras (receitas e despesas) em um determinado período.	Média	Não Aplicável
RF24	O sistema deve permitir a filtragem das transações por tipo (receita/despesa), data e categoria	Média	Não Aplicável
RF25	O sistema deve ser capaz de gerar um balanço financeiro simples para um período selecionado, mostrando o total de receitas, o total de despesas e o saldo	Média	Não Aplicável

Fonte: Elaborado pelos autores.

Quadro 7: Requisitos Funcionais - Parte 4

4.1.2 Requisitos Não Funcionais

Código	Módulo	Descrição
RNF01	Usabilidade	A interface do sistema deve ser intuitiva, responsiva (compatível e adaptada tanto para dispositivos desktop quanto mobile) e de fácil utilização, de modo que as tarefas essenciais da gestão da pousada sejam realizadas de forma eficiente e com mínimo esforço de aprendizado pela proprietária. Para isso, deve-se adotar os princípios de interface amigável como a priorização da simplicidade e da clareza, padrões de interface consistentes e acessíveis.
RNF02	Usabilidade	O sistema deve fornecer mensagens de feedback claras, objetivas e contextualizadas para todas as ações realizadas pela usuária, como confirmação de reserva (exemplo: reserva efetuada com sucesso) ou notificações de erros (exemplo: falha ao cadastrar um quarto), garantindo uma interação segura e satisfatória.
RNF03	Performance	O sistema deve apresentar um tempo de resposta baixo, com carregamento das páginas e execução de ações da proprietária entre 2 e 3 segundos, para garantir uma navegação fluida.
RNF04	Performance	O sistema deve ser capaz de lidar com a carga de trabalho estimada desde o registro e a consulta simultânea de dados à gestão de múltiplas reservas, sem degradação significativa no desempenho, este que deverá se manter estável mesmo em períodos de maior demanda, considerando o perfil sazonal do negócio.
RNF05	Segurança	O sistema deve garantir a segurança das informações da pousada e dos hóspedes através da implementação de mecanismos robustos de autenticação e autorização, de forma a assegurar que apenas a usuária autorizada consiga acessar ou alterar dados na aplicação.

Fonte: Elaborado pelos autores.

Quadro 8: Requisitos Não Funcionais - Parte 1

Código	Módulo	Descrição
RNF06	Segurança	Os dados sensíveis devem ser protegidos conforme as melhores práticas propostas pela LGPD (Lei Geral de Proteção de Dados Pessoais), incluindo: utilização de criptografia para proteger dados em trânsito (HTTPS) e em repouso, implementação de políticas de autenticação robusta e minimização da coleta de dados.
RNF07	Confiabilidade	O sistema deve estar disponível e funcionando corretamente por pelo menos 99 por cento do tempo, a fim de garantir que a proprietária tenha acesso ao sistema sempre que necessário, inclusive nos períodos com maior movimento de hóspedes na pousada.
RNF08	Confiabilidade	O sistema deve implementar mecanismos de tratamento de erros para que falhas e perdas de dados sejam prevenidas.
RNF09	Confiabilidade	O deploy da aplicação deve ser realizado em uma infraestrutura de nuvem (Amazon EC2), a fim de proporcionar maior estabilidade, flexibilidade à aplicação e permitir que possíveis atualizações e manutenções tenham impacto mínimo para a usuária.
RNF10	Escalabilidade	Embora o sistema, inicialmente, seja voltado para uma única usuária, a arquitetura deve ser projetada de forma a permitir futuras expansões no número de usuários e funcionalidades sem grandes refatorações.
RNF11	Documentação	O sistema deve possuir uma documentação completa, objetiva e atualizada, incluindo código-fonte, a arquitetura da aplicação, os fluxos de uso e as especificações de APIs possivelmente integradas.
RNF12	Documentação	A documentação deve estar versionada e organizada em no repositório Git — o GitHub —, este que deve ser utilizado no controle de versão da aplicação e colaboração entre os membros da equipe.
RNF13	Documentação	O desenvolvimento deve seguir as boas práticas de codificação e padrões recomendados para aplicações Django, a fim de assegurar a manutenibilidade, extensibilidade e integridade do sistema ao longo do seu ciclo de vida.

Fonte: Elaborado pelos autores.

Quadro 9: Requisitos Não Funcionais - Parte 2

4.1.3 Regras de Negócio

Regra	Descrição
RN01	A confirmação de uma reserva deve ocorrer mediante o pagamento de 50 por cento do valor total.
RN02	Cancelamentos e remarcações são permitidos, sujeitos a regras e taxas específicas.
RN03	As reservas devem ser registradas com, no mínimo, 2 dias de antecedência da data de entrada.
RN04	Um mesmo hóspede pode ter mais de uma reserva ativa.
RN05	Para efetuar uma reserva, os seguintes dados do hóspede são obrigatórios: nome completo, endereço completo, CPF, telefone e e-mail
RN06	Um responsável pode realizar reservas em nome de outros hóspedes.
RN07	Um responsável pode realizar reservas em nome de outros hóspedes.
RN08	Os horários padrão são: check-in das 16h às 22h; check-out das 8h às 14h.
RN09	A emissão de recibos ou comprovantes após check-in/out não é obrigatória.
RN10	Um quarto pode ficar indisponível para manutenção.
RN11	É permitido reservar um mesmo quarto para hóspedes diferentes em datas seguidas.
RN12	Atualmente, não há oferta de serviços adicionais vinculados à reserva.
RN13	São aceitas as formas de pagamento: Pix, dinheiro e cartão (com taxa da operadora).
RN14	Deve ser enviada a confirmação da reserva para o cliente com todos os dados necessários.
RN15	Apenas a proprietária deve acessar as informações dos hóspedes e reservas.

Fonte: Elaborado pelos autores.

Quadro 10: Regras de Negócio

4.2 História de Usuário

4.2.1 Descrição

4.3 Arquitetura

4.3.1 Desenho da Arquitetura

A arquitetura do sistema foi criada para atender às necessidades de um ambiente web moderno, seguro e escalável, com foco na praticidade do uso para a gestora da pousada.

A aplicação adota o padrão MTV (Model-Template-View), próprio do framework Django, no qual a separação entre dados, lógica de apresentação e lógica de negócio é claramente definida. Esse padrão proporciona uma estrutura organizada, facilitando tanto a manutenção quanto a adição de novas funcionalidades ao longo do tempo.

No front-end, a aplicação utiliza templates HTML integrados às views do Django, o que permite a criação de páginas dinâmicas. Os dados vindos do back-end são inseridos diretamente nos templates, tornando a navegação mais fácil, sem a necessidade de frameworks.

O back-end é totalmente desenvolvido com o framework Django, em Python, onde são tratadas as requisições, regras de negócio e persistência de dados. O Django oferece um conjunto de ferramentas prontas que aumentam a produtividade no desenvolvimento e garantem a segurança da aplicação.

Para o banco de dados, foi escolhido o PostgreSQL, compatível com o Django, sendo ideal para armazenar as reservas, dados de hóspedes, acomodações e demais informações do sistema.

A aplicação está hospedada na Amazon Web Services (AWS), a escolha dela se deu por sua alta disponibilidade, segurança avançada, escalabilidade sob demanda e facilidade de integração com outras ferramentas. Dentre os serviços da AWS utilizados, destacam-se: Amazon EC2, para hospedagem da aplicação Django; Amazon RDS (PostgreSQL), para o banco de dados gerenciado; Amazon S3, para armazenamento de arquivos estáticos e mídias; AWS Backup e Monitoramento, para garantir segurança e integridade dos dados.

A arquitetura permite que o sistema cresça ao longo do tempo, incorporando novos módulos e funcionalidades como integração com gateways de pagamento, envio automático de e-mails, e etc, mantendo a manutenibilidade, segurança e eficiência como foco do projeto.

4.3.2 Diagrama da Arquitetura

4.3.2.1 Diagrama de Implantação

A Figura 1 mostra o funcionamento da arquitetura do sistema.

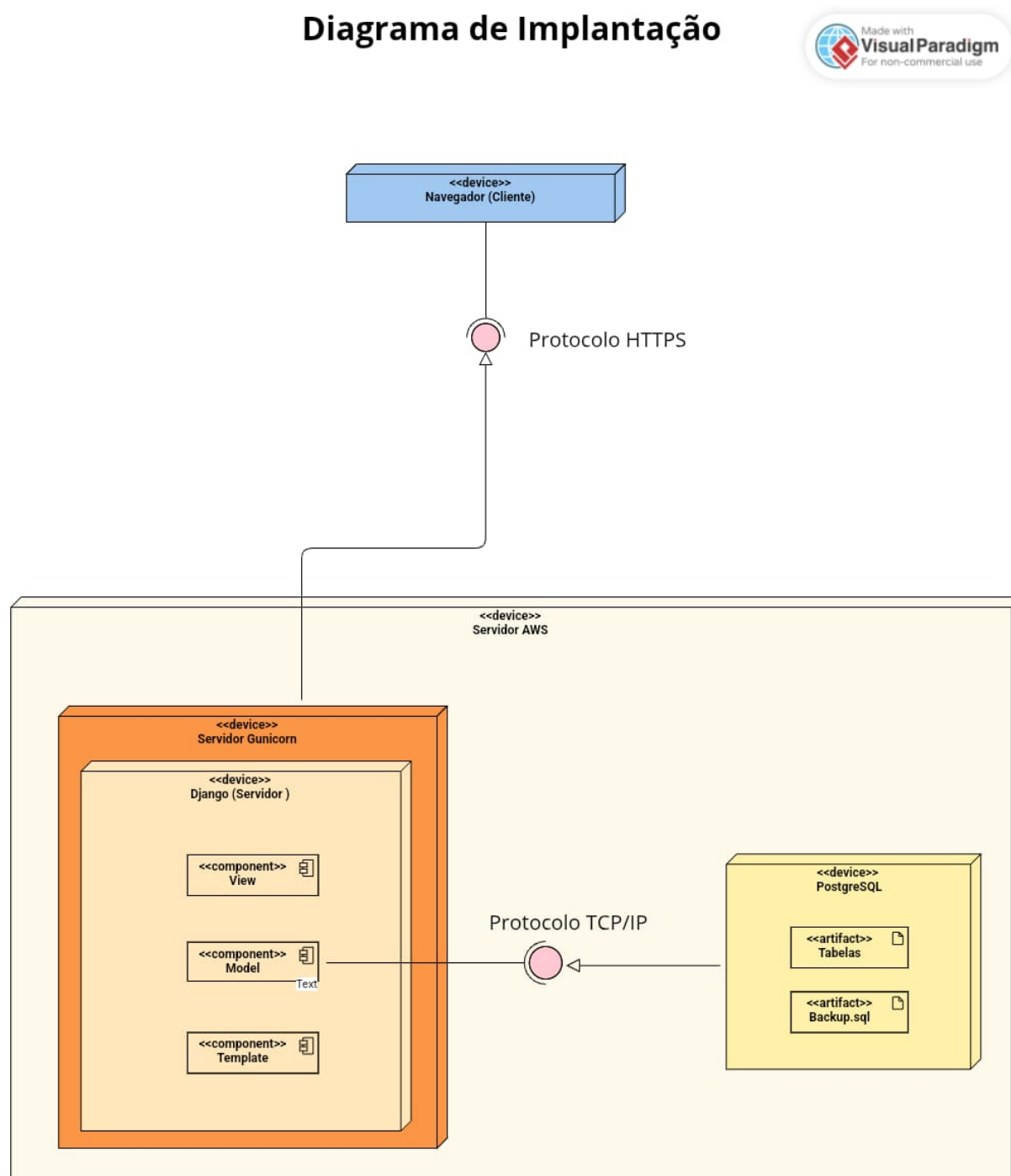


Figura 1 – Diagrama de Implantação desenvolvido no Online Visual-Paradigm

4.3.2.2 Diagrama de Componentes

A Figura 2 mostra o funcionamento da arquitetura do sistema.

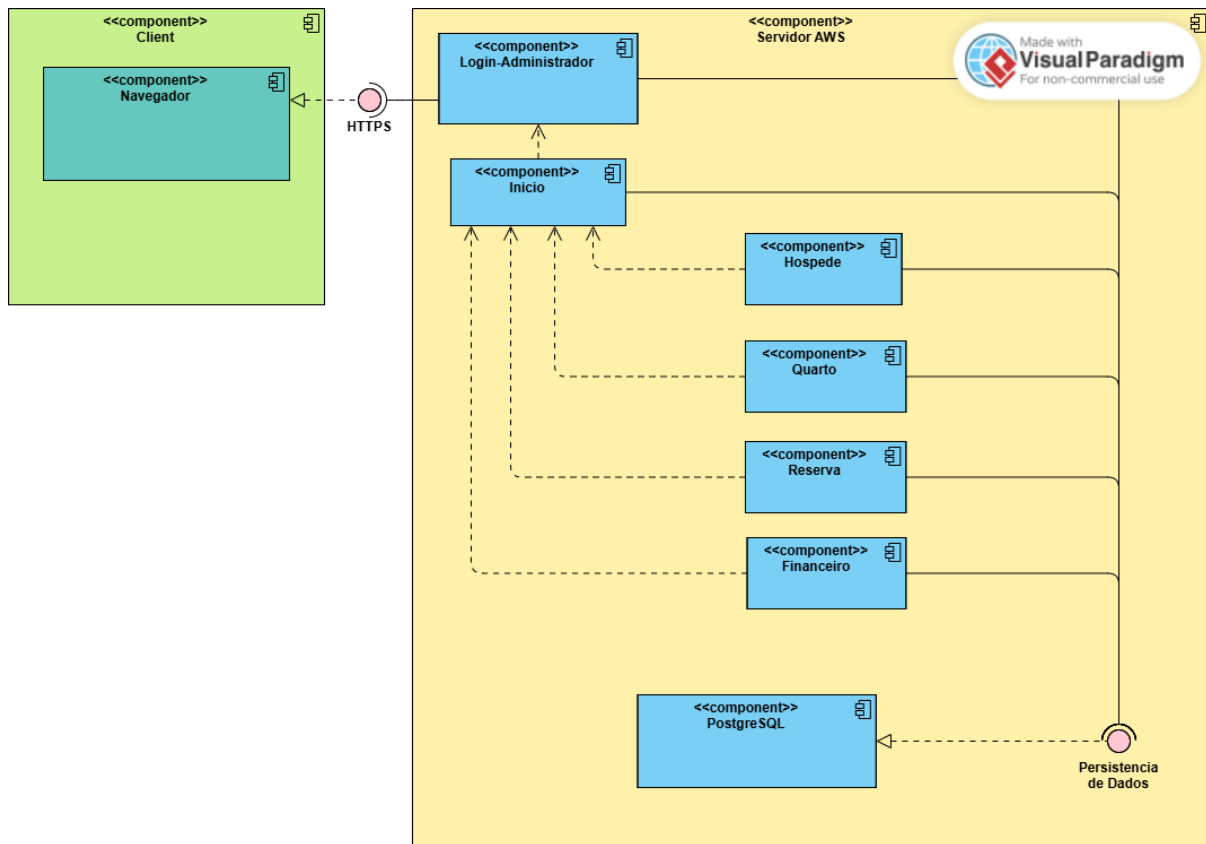


Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm

4.4 Tecnologias

4.4.1 Front-End

O Django é um framework web de alto nível baseado em Python que oferece uma série de recursos que o tornam ideal para o desenvolvimento de sistemas como o nosso gerenciador de reservas de quartos. Uma de suas principais vantagens é a rapidez no desenvolvimento, já que ele vem com diversas funcionalidades já prontas.

O desenvolvimento do Front-end da aplicação Chalés Água de Coco se dá pela combinação de templates HTML associados a Views (Django) para gerar páginas dinâmicas. Os templates exibem essas informações de forma estruturada na interface do usuário, permitindo que elementos HTML sejam preenchidos com dados fornecidos pelo servidor, assim utilizando o padrão MTV (Model-Template-View).

O que facilita a manutenção e a escalabilidade do sistema. Isso é essencial em um sistema de reservas, que pode crescer em funcionalidades como calendário de disponibilidade, gestão de hóspedes, geração de relatórios, envio de notificações, entre outros.

4.4.2 Back-End

O desenvolvimento do back-end desta aplicação Django é baseada no padrão MTV(Model-Template-View), assim funcionando com manipulação de dados e lógica de negócio através das Views, que são responsáveis por processar requisições, acessar o banco de dados e enviar informações para os templates.

Outro ponto importante que influenciou na nossa escolha é a segurança, pois o Django já vem com proteções contra ataques comuns, como injeção de SQL, o que é essencial em aplicações que lidam com dados pessoais e financeiros dos clientes.

4.4.3 Banco de Dados

O Banco de Dados escolhido para esta aplicação foi o PostgreSQL, sendo um banco escalável e flexível, este SGBD pode suportar grandes volumes de dados e de usuários além de ser compatível com uma grande gama de linguagens de programação. O PostgreSQL também é uma ótima opção por ser acessível, já que sua licença é livre, assim sem custos de licenciamento e a liberdade para modificar ou implementar o código-fonte da maneira que for necessária.

4.4.4 Infraestrutura

Agora falando um pouco sobre a Amazon Web Services. Ela é uma das plataformas de computação em nuvem mais completas e confiáveis do mercado. Utilizar a AWS para hospedar nosso sistema de gerenciamento de reservas traz várias vantagens:

Alta disponibilidade e escalabilidade: a AWS permite escalar a infraestrutura de acordo com a demanda, garantindo que o sistema continue funcionando bem mesmo em períodos de alta procura.

Confiabilidade e desempenho: a infraestrutura da AWS é robusta, distribuída globalmente e projetada para evitar falhas. Isso assegura que o sistema fique no ar com alta performance e baixa latência.

Segurança: a AWS oferece diversas camadas de segurança, incluindo criptografia de dados, controle de acesso, backups automatizados e monitoramento contínuo, protegendo tanto os dados dos hóspedes quanto as informações administrativas da pousada.

Serviços integrados: além do serviço de hospedagem, a AWS oferece bancos de dados gerenciados, armazenamento, envio de e-mails, monitoramento, entre outros — todos integráveis com o sistema Django de forma eficiente.

Por conta desses motivos, nós concluímos que a combinação de Django com a AWS é uma escolha estratégica e poderosa para o desenvolvimento do nosso sistema de reservas de quartos em uma pousada. Enquanto o Django acelera o desenvolvimento e garante um sistema seguro e bem estruturado, a AWS fornece a base tecnológica para garantir desempenho, estabilidade e escalabilidade. Juntos, eles possibilitam a entrega de uma solução profissional e confiável, algo que nós buscamos para o desenvolvimento do nosso sistema.

4.5 Ferramentas de Apoio

4.5.1 GitHub

O GitHub foi utilizado para controle de versão e colaboração durante o desenvolvimento do sistema. A plataforma permite armazenar e gerenciar o código-fonte, realizar revisões e integrar funcionalidades de forma eficiente. O GitHub facilitou a organização do fluxo de trabalho, o rastreamento de mudanças e a colaboração entre os membros da equipe, promovendo maior controle e transparência no ciclo de desenvolvimento.

4.5.2 BRModelo

O BRModelo foi utilizado para a modelagem lógica e relacional do banco de dados. A ferramenta oferece uma interface intuitiva para construção de diagramas entidade-relacionamento (DER), o que auxiliou na estruturação clara das tabelas, relacionamentos e chaves do sistema. O uso do BRModelo contribuiu diretamente para a coerência e integridade do esquema de dados implementado no PostgreSQL.

4.5.3 Visual Paradigm Online

O Visual Paradigm Online foi utilizado na criação dos diagramas de Implantação e Componentes. Esta ferramenta auxiliou na documentação da arquitetura do sistema, contribuindo para uma melhor compreensão dos fluxos e interações entre os componentes. A versão online possibilitou colaboração remota e armazenamento em nuvem, o que otimizou a produtividade da equipe.

4.5.4 Latex

O LaTeX foi utilizado na produção e formatação do trabalho acadêmico. Por meio de seu sistema de marcação, foi possível obter um alto nível de controle sobre a estrutura e apresentação do documento, garantindo consistência, qualidade e organização.

4.5.5 Google Meet

O Google Meet foi utilizado como plataforma de comunicação e realização de encontros virtuais da equipe ao longo do desenvolvimento do projeto. As reuniões periódicas possibilitaram a discussão de tarefas, alinhamento de prazos e entregas mais organizadas.

4.6 Manutenibilidade

A manutenibilidade do sistema de reservas para pousadas desenvolvido neste projeto é assegurada por meio de práticas estruturadas de engenharia de software, que facilitam a correção de erros, inclusão de novas funcionalidades e adaptação a futuras necessidades.

O sistema foi concebido com uma arquitetura modular, respeitando os princípios de separação de responsabilidades. Isso permite que diferentes partes do sistema, como interface, regras de negócio e persistência de dados, sejam modificadas de forma independente, minimizando impactos colaterais e reduzindo o tempo de manutenção.

Além disso, foram adotados padrões de codificação consistentes e bem documentados, com o intuito de facilitar a leitura e compreensão do código por outros desenvolvedores. Esses padrões promovem a reutilização e a extensibilidade do sistema.

A utilização do sistema de controle de versão Git, em conjunto com a plataforma GitHub, possibilita o rastreamento detalhado de alterações, revisão de código e colaboração eficaz entre os membros da equipe. Isso garante maior controle sobre o histórico de desenvolvimento e facilita a identificação e resolução de falhas.

A aplicação também contará com testes automatizados, cobrindo os principais fluxos da aplicação, como testes unitários para funções críticas e testes de integração entre os módulos.

Complementando essas práticas, será elaborada uma documentação técnica e funcional completa, abrangendo instruções de uso, instalação, configuração e manutenção.

Por fim, o projeto segue um ciclo de desenvolvimento bem definido, com etapas de planejamento, codificação, testes, implantação e manutenção. Essa abordagem estruturada proporciona maior previsibilidade, qualidade e agilidade na evolução contínua da aplicação, assegurando sua longevidade e adaptabilidade.

4.7 Segurança, Privacidade e Legislação

A segurança da informação é um aspecto central no desenvolvimento do sistema da Pousada Chalés Água de Coco. Para garantir a proteção dos dados dos hóspedes e da própria aplicação, foi adotado o framework Django, que já incorpora diversas camadas de segurança por padrão. Além disso, foram implementadas boas práticas relacionadas à privacidade e à conformidade com a legislação brasileira de proteção de dados.

O Django fornece mecanismos robustos contra ataques comuns na web, como injeção de SQL, execução remota de código, ataques Cross-Site Scripting (XSS) e falsificação de requisições entre sites (CSRF). Essas proteções são ativadas por padrão e complementadas com práticas adicionais durante o desenvolvimento.

Entre os recursos utilizados, destacam-se:

Sistema de autenticação integrado: O Django oferece um módulo completo para autenticação e autorização de usuários, permitindo o controle de acesso baseado em permissões. O sistema foi configurado para exigir credenciais válidas e proteger páginas sensíveis com autenticação obrigatória.

Proteção contra CSRF (Cross-Site Request Forgery): Todas as requisições POST são protegidas por tokens CSRF, garantindo que ações críticas só possam ser executadas por usuários autenticados a partir da própria aplicação.

Escapamento automático de HTML (XSS): O mecanismo de templates do Django realiza automaticamente o escapamento de dados inseridos pelos usuários, o que impede a execução de scripts maliciosos.

Hash de senhas com algoritmos seguros: As senhas dos usuários são armazenadas de forma segura com algoritmos de hash modernos (como PBKDF2), tornando inviável a recuperação das senhas mesmo em caso de vazamento da base de dados.

Validação de entradas e uso de ORM: Ao utilizar o ORM (Object-Relational Mapper) do Django, o sistema evita o uso direto de comandos SQL, o que mitiga o risco de injeção de código malicioso no banco de dados.

Gerenciamento de sessões seguro: O Django gerencia sessões de forma criptografada e armazena os identificadores de sessão com proteção contra falsificação. O sistema pode ser configurado para expirar sessões inativas e exigir reautenticação para operações sensíveis.

Além dessas funcionalidades nativas, o sistema conta com certificado SSL/TLS para garantir a criptografia do tráfego entre cliente e servidor, além de backups automáticos armazenados na nuvem (AWS), oferecendo resiliência em caso de falhas técnicas.

4.7.1 Conformidade com a LGPD

Para atender à Lei Geral de Proteção de Dados (LGPD - Lei nº 13.709/2018), o sistema foi desenvolvido com foco na coleta mínima de dados, transparência e controle por parte do usuário. As finalidades de uso das informações são claramente definidas, e o usuário pode, a qualquer momento, solicitar a remoção de seus dados ou a revisão do consentimento.

Com isso, o sistema não apenas assegura uma navegação segura e confiável, mas também respeita os direitos fundamentais de privacidade e liberdade informacional previstos na legislação brasileira.

4.7.2 Protocolos de Segurança na Comunicação

A segurança na troca de informações no sistema “Chalés Água de Coco” foi uma prioridade desde as fases iniciais de desenvolvimento. Para garantir a integridade, a confidencialidade e a autenticidade dos dados transmitidos, adotaram-se protocolos amplamente reconhecidos e confiáveis no ambiente web moderno.

4.7.2.1 Certificação Digital com SSL/TLS

A aplicação foi implementada com suporte ao protocolo TLS (Transport Layer Security), sucessor do SSL, por meio de certificado digital instalado no servidor web em nuvem (AWS). Este certificado criptografa a comunicação entre o cliente (navegador do usuário) e o servidor, tornando os dados transmitidos — como informações cadastrais e de reservas — inacessíveis a terceiros.

A emissão do certificado TLS envolveu a geração de um par de chaves criptográficas e o envio de um pedido de assinatura (CSR) a uma autoridade certificadora confiável, que validou o domínio e forneceu o certificado. Este foi instalado e configurado no ambiente AWS, garantindo que todas as requisições utilizem conexões criptografadas via HTTPS.

4.7.2.2 Protocolo HTTPS

A utilização do protocolo HTTPS (HyperText Transfer Protocol Secure) é fundamental para proteger sessões de navegação dos usuários. Com ele, todas as requisições realizadas ao sistema de reservas da pousada são criptografadas, evitando ataques como man-in-the-middle e a exposição de dados sensíveis.

A aplicação redireciona automaticamente qualquer tentativa de acesso via HTTP para HTTPS, reforçando as boas práticas de segurança. O uso de HTTPS também contribui para a credibilidade da plataforma junto aos usuários, além de ser um critério relevante para mecanismos de busca e navegadores modernos.

4.7.2.3 Compatibilidade com a Arquitetura Django

A estrutura do Django, utilizada no desenvolvimento da aplicação, já oferece suporte nativo a conexões seguras. As configurações de segurança foram ajustadas para forçar o uso de HTTPS em todas as rotas públicas e administrativas, além da inclusão de headers HTTP como Strict-Transport-Security, X-Content-Type-Options, X-Frame-Options e Content-Security-Policy, reforçando a proteção contra ataques como clickjacking, XSS e sniffing de conteúdo.

4.8 Modelagem do Banco de Dados

4.8.1 Modelo Entidade-Relacionamento - MER

A Figura 3 mostra o Modelo Entidade-Relacionamento (MER) do sistema.

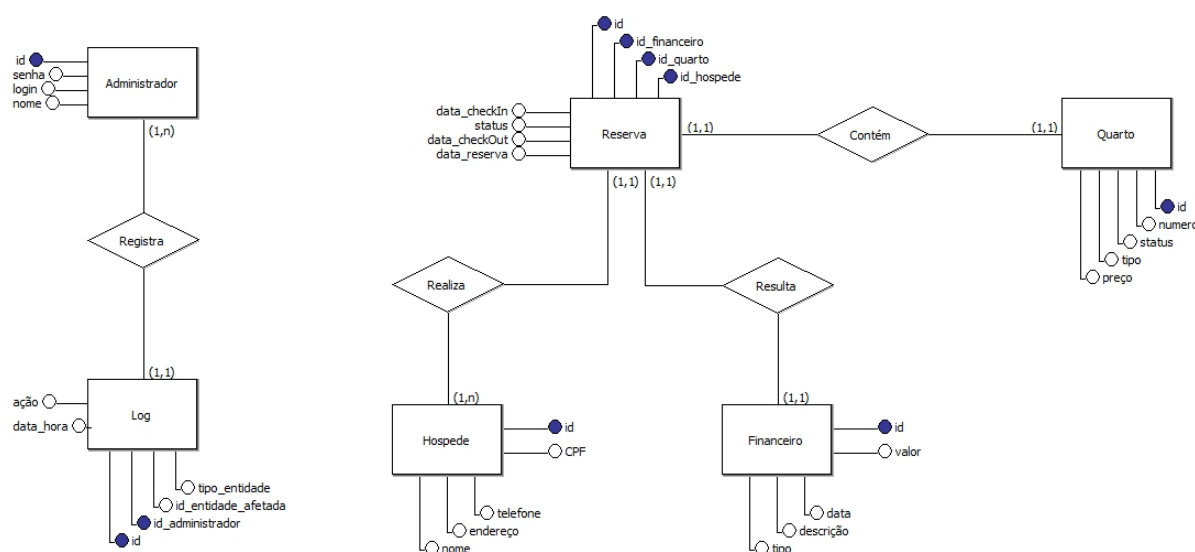


Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo

4.8.2 Diagrama Entidade-Relacionamento - DER

A Figura 4 mostra o Diagrama Entidade-Relacionamento (DER) do sistema.

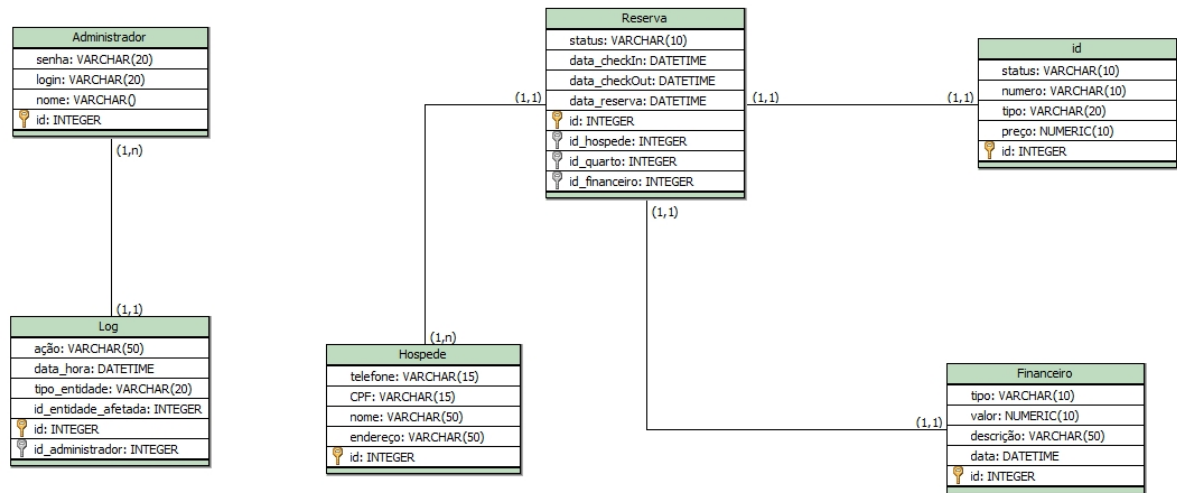


Figura 4 – Diagrama Entidade-Relacionamento (MER) desenvolvido no brModelo

4.9 Entregas

5 Viabilidade Financeira

Visando a viabilidade financeira do desenvolvimento da aplicação de gestão de reservas de quarto, conduzimos um estudo dos custos necessários e produzimos relatórios referentes a diferentes cenários. Os dados utilizados no estudo são concretos e retirados de fontes seguras, nos ajudando a visualizar a factibilidade do projeto e ter melhor percepção para a tomada de decisões estratégicas.

5.1 Custos

5.1.1 Custo Estrutural

Tabela 2 – Resumo dos custos de Equipamentos e Serviços

Itens	Quantidade	Custo mensal	Total 4 meses	Total 9 meses
Notebooks	6	R\$ 0,00	R\$ 0,00	R\$ 0,00
Roteadores	6	R\$ 0,00	R\$ 0,00	R\$ 0,00
Internet (Assinatura)	6	R\$ 114,97	R\$ 2.759,28	R\$ 6.208,38
Eletricidade (Fatura)	6	R\$ 3,44	R\$ 82,56	R\$ 185,76
Total	24	R\$ 118,41	R\$ 2.841,84	R\$ 6.394,14

Fonte: Elaborado pelos autores.

Tabela 3 – Resumo dos custos de Infraestrutura

Tipo	Serviço	Mensal	Total 4 meses	Total 9 meses
Proxy Reverso	Servidor Nginx	R\$ 0,00	R\$ 0,00	R\$ 0,00
Aplicação	Servidor Gunicorn	R\$ 0,00	R\$ 0,00	R\$ 0,00
Framework Web	Servidor Django	R\$ 0,00	R\$ 0,00	R\$ 0,00
Banco de Dados	PostgreSQL	R\$ 0,00	R\$ 0,00	R\$ 0,00
Hospedagem	AWS	R\$ 0,00	R\$ 0,00	R\$ 0,00
Total		R\$ 0,00	R\$ 0,00	R\$ 0,00

Fonte: Elaborado pelos autores.

5.1.2 Custo Mão de Obra

Tabela 4 – Quantidade e horas trabalhadas por função

Função	Quantidade	Horas/Dia	Dias/Mês	Total de Horas/Mês
Analista de Cronograma (PMO)	1	6	22	25,90
Engenheiro de Dados (DBA)	1	6	22	47,30
Analista de Documentação	1	6	22	75,30
Gerente de Projeto (PM)	1	6	22	78,90
Desenvolvedor Front-End	1	6	22	150,00
Desenvolvedor Back-End	1	6	22	66,00
Total Mão de Obra	6	36	132	443,40

Fonte: Elaborado pelos autores.

Tabela 5 – Custos por função

Função	Custo Hora (R\$)	Custo Mensal (R\$)	Total 4 meses (R\$)	Total 9 meses (R\$)
Analista de Cronograma (PMO)	16,00	155,40	621,60	1.398,60
Engenheiro de Dados (DBA)	13,00	283,80	1.135,20	2.554,20
Analista de Documentação	15,00	451,80	1.807,20	4.066,20
Gerente de Projeto (PM)	16,00	473,40	1.893,60	4.260,60
Desenvolvedor Front-End	13,00	900,00	3.600,00	8.100,00
Desenvolvedor Back-End	13,00	396,00	1.584,00	3.564,00
Total Mão de Obra	86,00	2.660,40	10.641,60	23.943,60

Fonte: Elaborado pelos autores.

5.1.3 Custo Total

Tabela 6 – Custo Total por Categoria

Categoria	Custo Mensal (R\$)	Custo Total (4 meses) (R\$)	Custo Total (9 meses) (R\$)
Mão de Obra	23.943,60	95.774,40	215.492,40
Estrutura	118,41	473,64	1.065,69
Total	24.062,01	96.248,04	216.558,09

Fonte: Elaborado pelos autores.

5.2 Cenários

5.2.1 Cenário Otimista

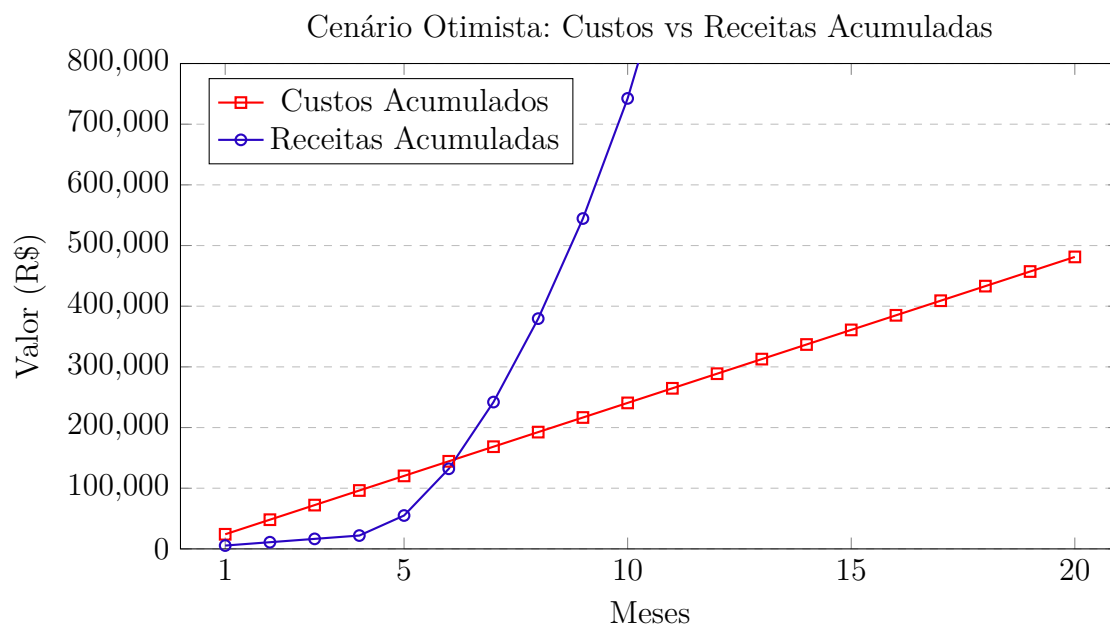


Figura 5 – Comparação dos custos e receitas acumuladas no cenário otimista

5.2.2 Cenário Pessimista

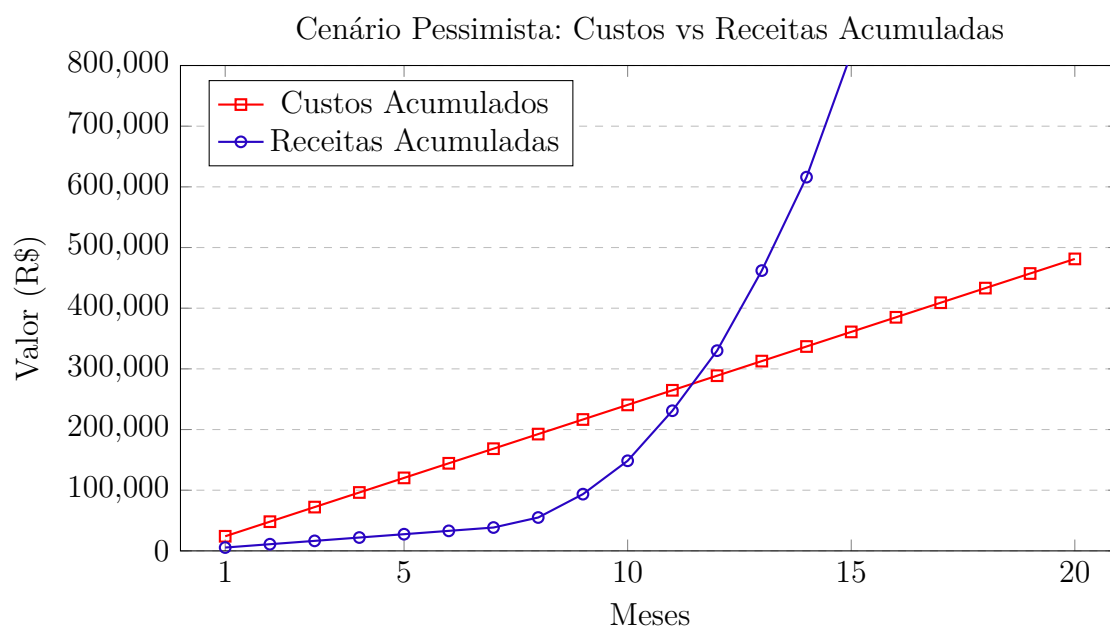


Figura 6 – Comparação dos custos e receitas acumuladas no cenário pessimista

5.2.3 Cenário Realista

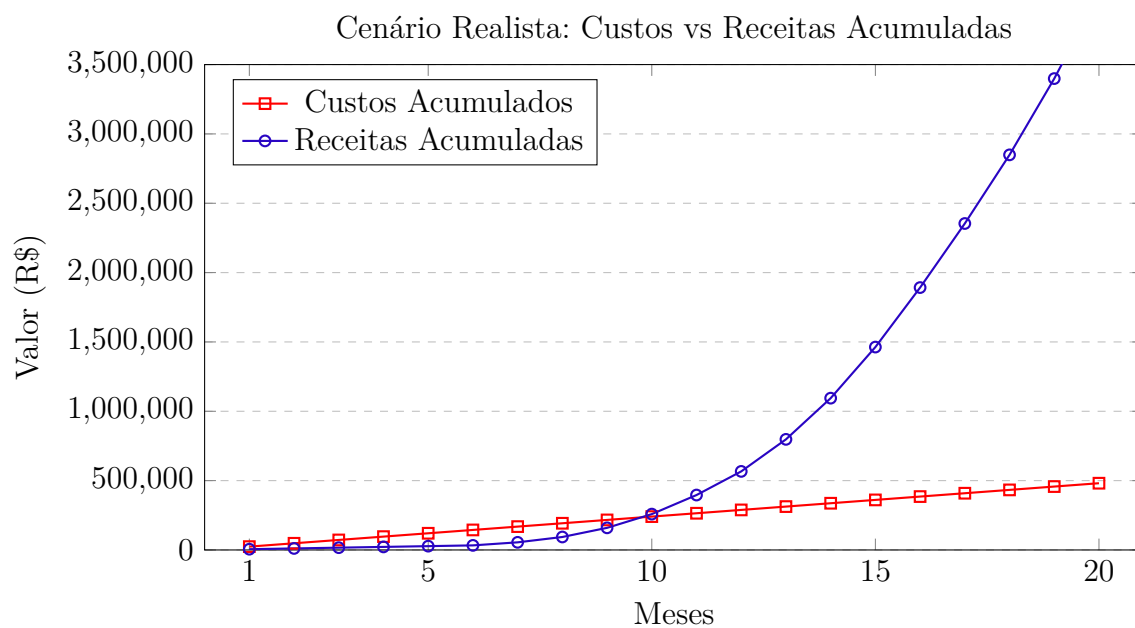


Figura 7 – Comparação dos custos e receitas acumuladas no cenário realista

6 Considerações Finais

7 Conclusão

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.

Sed eleifend, eros sit amet faucibus elementum, urna sapien consectetur mauris, quis egestas leo justo non risus. Morbi non felis ac libero vulputate fringilla. Mauris libero eros, lacinia non, sodales quis, dapibus porttitor, pede. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi dapibus mauris condimentum nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet erat. Nulla varius. Etiam tincidunt dui vitae turpis. Donec leo. Morbi vulputate convallis est. Integer aliquet. Pellentesque aliquet sodales urna.

Referências

Apêndices

APÊNDICE A – Quisque libero justo

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

APÊNDICE B – Nullam elementum urna vel imperdiet sodales elit ipsum pharetra ligula ac pretium ante justo a nulla curabitur tristique arcu eu metus

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

Anexos

ANEXO A – Morbi ultrices rutrum lorem.

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

ANEXO B – Cras non urna sed feugiat cum sociis natoque penatibus et magnis dis parturient montes nascetur ridiculus mus

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

ANEXO C – Fusce facilisis lacinia dui

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.