

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

Projeto 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 Desenvolvi-
mento de Sistemas.

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 .	64
Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm .	65
Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo .	70
Figura 4 – Diagrama Entidade-Relacionamento (DER) desenvolvido no brModelo	71
Figura 5 – Comparação dos custos e receitas acumuladas no cenário otimista . .	75
Figura 6 – Comparação dos custos e receitas acumuladas no cenário pessimista .	76
Figura 7 – Comparação dos custos e receitas acumuladas no cenário realista . .	77

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 - Parte 1	24
Quadro 4 – Product Backlog - Parte 2	25
Quadro 5 – Product Backlog - Parte 3	26
Quadro 6 – Product Backlog - Parte 4	27
Quadro 7 – Sprints Backlog - Parte 1	28
Quadro 8 – Sprints Backlog - Parte 2	29
Quadro 9 – Sprints Backlog - Parte 3	30
Quadro 10 – Sprints Backlog - Parte 4	31
Quadro 11 – Identificação dos Riscos do Projeto - Parte 1	32
Quadro 12 – Identificação dos Riscos do Projeto - Parte 2	33
Quadro 13 – Análise e Planejamento dos Riscos - Parte 1	34
Quadro 14 – Análise e Planejamento dos Riscos - Parte 2	35
Quadro 15 – Mecanismos de Monitoramento dos Riscos - Parte 1	36
Quadro 16 – Questionário Aplicado à Proprietária - Parte 1	40
Quadro 17 – Questionário Aplicado à Proprietária - Parte 3	41
Quadro 18 – Regras de Negócio	42
Quadro 19 – Requisitos Funcionais - Parte 1	43
Quadro 20 – Requisitos Funcionais - Parte 2	44
Quadro 21 – Requisitos Funcionais - Parte 3	45
Quadro 22 – Requisitos Funcionais - Parte 4	46
Quadro 23 – Requisitos Funcionais - Parte 5	47
Quadro 24 – Requisitos Não Funcionais - Parte 1	48
Quadro 25 – Requisitos Não Funcionais - Parte 2	49

Lista de tabelas

Tabela 1 – Função dos Integrantes da Equipe	23
Tabela 2 – Resumo dos custos de Equipamentos e Serviços	73
Tabela 3 – Resumo dos custos de Infraestrutura	74
Tabela 4 – Quantidade e horas trabalhadas por função	74
Tabela 5 – Custos por função	74
Tabela 6 – Custo Total por Categoria	75

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
AWS	Amazon Web Services
CSRF	Cross-Site Request Forgery (Falsificação de Requisição entre Sites)
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
LGPD	Lei Geral de Proteção de Dados
MVC	Model-View-Controller (Modelo-Visão-Controlador)
MTV	Model-Template-View (Modelo-Template-Visualização)
ORM	Object-Relational Mapper (Mapeador Objeto-Relacional)
PBKDF2	Password-Based Key Derivation Function 2
POST	Post Method (Método de Envio via HTTP)
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security
URL	Uniform Resource Locator
XSS	Cross-Site Scripting (Script Entre Sites)

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 <i>Scrum</i>	22
3.3	Artefatos	23
3.3.1	Product Backlog	23
3.3.2	<i>Sprint Backlog</i>	27
3.4	Gestão de Riscos	31
3.4.1	Identificação dos Riscos do Projeto	32
3.4.2	Análise e Planejamento dos Riscos	34
3.4.3	Monitoramento dos Riscos	35
3.5	Repositório da Aplicação	37
3.5.1	Definição do repositório da aplicação	37
4	DESENVOLVIMENTO DO PROJETO	39
4.1	Escopo do Projeto	39
4.1.1	Regras de Negócio	41
4.1.2	Requisitos do Sistema	43
4.1.2.1	Requisitos Funcionais	43
4.1.2.2	Requisitos Não Funcionais	47
4.2	História de Usuário	50
4.2.1	Descrição das Histórias de Usuário	50
4.3	Arquitetura	63
4.3.1	Desenho da Arquitetura	63

4.3.1.1	Estrutura da Solução	63
4.3.1.2	Banco de Dados e Infraestrutura	63
4.3.2	Diagrama da Arquitetura	63
4.3.2.1	Diagrama de Implantação	63
4.3.2.2	Diagrama de Componentes	64
4.4	Tecnologias	65
4.4.1	Django	65
4.4.1.1	Front-end	65
4.4.1.2	Back-end	66
4.4.1.3	Banco de Dados	66
4.4.2	Justificativa da Escolha	66
4.4.2.1	Justificativa Técnica	66
4.4.2.2	Infraestrutura com AWS	67
4.5	Ferramentas de Apoio	67
4.5.1	GitHub	67
4.5.2	BRModelo	67
4.5.3	Visual Paradigm Online	67
4.5.4	Latex	68
4.5.5	Google Meet	68
4.6	Manutenibilidade	68
4.7	Segurança, Privacidade e Legislação	69
4.7.1	Segurança da Aplicação	69
4.7.2	Segurança na Comunicação	69
4.7.2.1	Conformidade com a LGPD	69
4.8	Modelagem do Banco de Dados	70
4.8.1	Modelo Entidade-Relacionamento - MER	70
4.8.2	Diagrama Entidade-Relacionamento - DER	70
4.9	Entregas	71
4.9.1	Desenvolvimento de um Tema - 08/04	71
4.9.2	Desenho da Aplicação - 29/04	71
4.9.3	Prova de Conceito - 20/05	71
4.9.4	Análise e Documentação - 10/06	72
4.9.5	Produto Mínimo Viável - 24/06	72
5	VIABILIDADE FINANCEIRA	73
5.1	Custos	73
5.1.1	Custo Estrutural	73
5.1.2	Custo Mão de Obra	74
5.1.3	Custo Total	75

5.2	Cenários	75
5.2.1	Cenário Otimista	75
5.2.2	Cenário Pessimista	76
5.2.3	Cenário Realista	76
6	CONSIDERAÇÕES FINAIS	78
7	CONCLUSÃO	79
	REFERÊNCIAS	80
	APÊNDICES	83
	APÊNDICE A – DIÁRIO DE BORDO	84
A.1	1° SEMANA	84
A.2	2° SEMANA	84
A.3	3° SEMANA	84
A.4	4° SEMANA	85
A.5	5° SEMANA	85
A.6	6° SEMANA	85
A.7	7° SEMANA	86
A.8	8° SEMANA	86
A.9	9° SEMANA	86
A.10	10° SEMANA	86
A.11	11° SEMANA	86
A.12	12° SEMANA	87
A.13	13° SEMANA	87

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, 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, tem o seu armazenamento em nuvem.

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 de gestão aplicadas 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 entre os membros.

3.1.1 Funções e Responsabilidades

A divisão das funções e responsabilidades foi realizada de maneira estratégica e levou em consideração as competências técnicas de cada membro da equipe, visando a entrega do produto final dentro do prazo e escopo estipulados. 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 a entrega do produto final ocorresse dentro do prazo estipulado e estivesse 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 ([SCHWABER; SUTHERLAND, 2013](#)), o *Scrum* é um *framework* estruturado desenvolvido na década de 1990, com o objetivo de auxiliar equipes na criação e gerenciamento de produtos complexos. Por 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. Enquanto a inspeção envolve o acompanhamento regular

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 (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	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	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

dos artefatos e progresso, auxiliando na identificação precoce de problemas e garantindo a transparência dos processos. Seguida pela adaptação que refere-se à capacidade de fazer ajustes no processo em resposta aos problemas anteriormente identificados na inspeção, com o objetivo de otimizar os resultados da equipe.

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, 2013](#)), 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.

- i) O Dono do Produto (*Product Owner*) é o representante das partes interessadas (*Stakeholders*) e tem como responsabilidade principal gerenciar o *backlog* do produto. Seu objetivo é maximizar o valor do produto para o usuário final e otimizar o trabalho do Time de Desenvolvimento.
- ii) O *Scrum Master* é o responsável por promover e facilitar a aplicação da teoria e das práticas do *framework*, a fim de ajudar a equipe a superar problemas que possam afetar seu progresso.
- iii) 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.

Diante disso, houve a distribuição dos papéis de um time *Scrum* entre os integrantes da equipe registrada na “[Tabela 1](#)”.

Tabela 1 – Função dos Integrantes da Equipe

Integrante	Função		
	<i>Product Owner</i>	<i>Scrum Master</i>	Time de Desenvolvimento
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 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

No Guia do *Scrum* ([SCHWABER; SUTHERLAND, 2013](#)), o *Product Backlog* é definido como uma lista ordenada de todos os itens de trabalho, incluindo funcionalidades e requisitos necessários para a produção do produto, que oferecem o máximo valor e utilidade para o cliente.

Ciente disso, o *product owner* do projeto elaborou o *backlog* inicial da aplicação *web* de gestão da pousada Chalés Água de Coco (Quadro 3), com base nas histórias de usuário levantadas pela equipe. Isso porque, elas refletem fielmente as necessidades e expectativas da usuária e principal *stakeholder* garantindo que o desenvolvimento em total alinhamento.

Código	Item	Categoria	Prioridade
1	Definir e configurar ambiente de desenvolvimento	Requisito Técnico	ALTA
2	Definir ambiente de hospedagem/publicação	Requisito Técnico	ALTA
3	Organizar repositório e fluxo Git	Requisito Técnico	ALTA
4	Levantar requisitos funcionais e não funcionais	Modelagem de dados	ALTA
5	Mapear casos de uso	Modelagem de dados	ALTA
6	Documentar o levantamento e registrar os requisitos	Documentação	ALTA
7	Criar Modelo Entidade-Relacionamento (MER)	Modelagem de dados	ALTA
8	Criar Diagrama de Entidade-Relacionamento (DER)	Modelagem de dados	ALTA
9	Criar Diagrama de Componentes	Arquitetura	ALTA
10	Criar Diagrama de Implantação	Arquitetura	ALTA
11	Documentar os diagramas produzidos	Documentação	ALTA
12	Configurar ambiente do servidor	Requisito Técnico	ALTA
13	Configurar banco de dados PostgreSQL	Requisito Técnico	ALTA
14	Implementar login e logout de usuário	Autenticação e Segurança	ALTA
15	Criar funcionalidade de cadastro de quartos	Gestão de Quartos	ALTA
16	Criar funcionalidade de exclusão de cadastro de quartos	Gestão de Quartos	MÉDIA
17	Criar funcionalidade de listagem de quartos	Gestão de Quartos	MÉDIA
18	Criar funcionalidade de edição do cadastro de quartos	Gestão de Quartos	MÉDIA

Fonte: Elaborado pelos autores.

Quadro 3: Product Backlog - Parte 1

Código	Item	Categoria	Prioridade
19	Criar interfaces para gestão de quartos	Gestão de Quartos	MÉDIA
20	Implementar funcionalidade de alteração manual do status (disponível, indisponível, em manutenção) do quarto	Gestão de Quartos	MÉDIA
21	Criar funcionalidade de cadastro de hóspedes	Gestão de Hóspedes	ALTA
22	Criar funcionalidade de exclusão do cadastro de hóspedes	Gestão de Hóspedes	MÉDIA
23	Criar funcionalidade de edição do cadastro de hóspedes	Gestão de Hóspedes	MÉDIA
24	Criar interfaces para gestão de hóspedes	Gestão de Hóspedes	MÉDIA
25	Criar funcionalidade de cadastro de reservas	Gestão de Reservas	ALTA
26	Criar funcionalidade de exclusão do cadastro de reservas	Gestão de Reservas	ALTA
27	Criar funcionalidade de edição do cadastro de reservas	Gestão de Reservas	ALTA
28	Criar interfaces para gestão de Reservas	Gestão de Reservas	ALTA
29	Criar funcionalidade para visualização do histórico de reservas por hóspede	Gestão de Reservas	MÉDIA
30	Criar lógica de validação de disponibilidade de quartos para reservas	Gestão de Reservas	ALTA
31	Implementar atualização automática do status do quarto para “ocupado” após o <i>check-in</i>	<i>Check-in</i> e <i>Check-out</i>	ALTA
32	Criar funcionalidade de atualização do <i>check-out</i>	<i>Check-in</i> e <i>Check-out</i>	ALTA
33	Integrar ambientes, <i>back-end</i> e <i>front-end</i>	Requisito técnico	ALTA
34	Documentar a configuração e o código desenvolvido	Documentação	ALTA
35	Criar funcionalidade de cadastro de receitas	Gestão Financeira	BAIXA
36	Criar funcionalidade para registrar comprovação de pagamento de reservas	Gestão Financeira/Gestão de Reservas	BAIXA

Fonte: Elaborado pelos autores.

Quadro 4: Product Backlog - Parte 2

Código	Item	Categoria	Prioridade
37	Criar funcionalidade de exclusão do cadastro de receitas	Gestão Financeira	BAIXA
38	Criar funcionalidade de edição do cadastro de receitas	Gestão Financeira	BAIXA
39	Criar interface para gestão de receitas	Gestão Financeira	BAIXA
40	Criar funcionalidade de cadastro de despesas	Gestão Financeira	BAIXA
41	Criar funcionalidade de exclusão do cadastro de despesas	Gestão Financeira	BAIXA
42	Criar funcionalidade de edição do cadastro de despesas	Gestão Financeira	BAIXA
43	Criar interface para gestão de despesas	Gestão Financeira	BAIXA
44	Criar funcionalidade para criar relatórios de quartos	Gestão Financeira	BAIXA
45	Criar funcionalidade para criar relatórios de hóspedes	Gestão Financeira	BAIXA
46	Criar funcionalidade para criar relatórios de reservas	Gestão Financeira	BAIXA
47	Criar balanço financeiro simples (receitas, despesas, saldo) por período	Gestão Financeira	BAIXA
48	Criar funcionalidade de filtragem de receita/despesa (data, categoria)	Gestão Financeira	BAIXA
49	implementar envio de notificação (<i>e-mail</i>) para hóspede após confirmação da reserva	Notificações	BAIXA
50	Implementar criptografia	Autenticação e Segurança	BAIXA
51	Definir escopo dos testes	Testes	MÉDIA
52	Identificar cenários de teste	Testes	MÉDIA
53	Elaborar casos de teste	Testes	MÉDIA
54	Definir as ferramentas de teste	Testes	MÉDIA
55	Estabelecer ambiente de teste	Testes	MÉDIA
56	Definir critérios de aceitação	Testes	MÉDIA
57	Preparar dados de teste	Testes	MÉDIA
58	Executar testes gerais	Testes	MÉDIA
59	Executar testes SSL	Testes	MÉDIA
60	Analisar e otimizar <i>headers</i> de segurança	Autenticação e Segurança	MÉDIA
61	Realizar ajustes de segurança	Autenticação e Segurança	MÉDIA
62	Documentar resultados dos testes	Testes	MÉDIA
63	Documentar componentes e estilos	Documentação	ALTA

Fonte: Elaborado pelos autores.

Quadro 5: Product Backlog - Parte 3

Código	Item	Categoria	Prioridade
64	Documentar o plano e a execução de testes	Documentação	MÉDIA
65	Registrar escolhas e mudanças de rumo	Documentação	ALTA
66	Documentar problemas ocorridos e lições aprendidas	Documentação	ALTA
67	Elaborar o plano de implantação	Implantação	MÉDIA
68	Realizar implantação do sistema	Implantação	MÉDIA
69	Revisão final da documentação técnica	Documentação	ALTA
70	Treinamento da proprietária da pou-sada para uso da aplicação	Implantação	MÉDIA

Fonte: Elaborado pelos autores.

Quadro 6: Product Backlog - Parte 4

3.3.2 *Sprint Backlog*

O *Sprint Backlog* é um plano elaborado pelo time de desenvolvimento, formado por um conjunto de itens selecionados da *Product Backlog*. Esse plano possui um período de tempo determinado para sua realização, chamado de *Sprint*.

Esse artefato promove a transparência das atividades do projeto, tornando visíveis e acessíveis para todos os membros da equipe as atividades que devem ser realizadas na *Sprint*, garantindo que todos estejam cientes do que está sendo desenvolvido.

Por isso, o time implementou o ?? que registra todas as *Sprints Backlog* planejadas para o decorrer do projeto.

Para isso, considerando a flexibilidade oferecida por essa metodologia dividiu-se as *sprints* do projeto em períodos de quinze dias (duas semanas), período dentro do estipulado pelo Guia do *Scrum* (SCHWABER; SUTHERLAND, 2013). Essa escolha de duração para as *Sprints* levou em consideração que a equipe atuaria majoritariamente de forma remota, com tempo diário individual limitado para o trabalho. Este contexto demandava prazos mais longos para a execução das tarefas.

As atividades incluídas nas *Sprints Backlog* foram baseadas nos itens da *Product Backlog*, no prazo de entrega do MVP e nas demais tarefas essenciais do projeto, como realização de testes e a elaboração da documentação técnica.

Sprint	Período	Objetivo	Atividades
1	08/04/2025 a 22/04/2025	Realizar o planejamento do projeto	<ul style="list-style-type: none"> - Definir e configurar ambiente de desenvolvimento. - Definir ambiente de hospedagem/publicação. - Organizar repositório e fluxo Git.
2	22/04/2025 a 06/05/2025	Iniciar a fase de modelagem e documentação dos requisitos	<ul style="list-style-type: none"> - Levantar requisitos funcionais e não funcionais. - Mapear casos de uso. - Documentar o levantamento e registrar os requisitos.
3	06/05/2025 a 20/05/2025	Finalizar a modelagem de dados e definir a arquitetura do sistema	<ul style="list-style-type: none"> - Criar do Diagrama de Componentes. - Criar do Diagrama de Implantação. - Criar do modelo relacional (MER). - Criar do Diagrama de Entidade e Relacionamento (DER). - Documentar os diagramas produzidos.
4	20/05/2025 a 03/06/2025	Configurar o ambiente do servidor, banco de dados e iniciar o desenvolvimento das funcionalidades para o MVP	<ul style="list-style-type: none"> - Configurar ambiente do servidor. - Configurar banco de dados PostgreSQL. - Implementar login e logout de usuários. - Criar funcionalidade de cadastro de quartos. - Criar funcionalidade de exclusão de cadastro de quartos. - Criar funcionalidade de listagem de quartos.

Fonte: Elaborado pelos autores.

Quadro 7: Sprints Backlog - Parte 1

Sprint	Período	Objetivo	Atividades
5	03/06/2025 a 17/06/2025	Desenvolver as funcionalidades de edição de quartos, interfaces e a gestão básica de hóspedes	<ul style="list-style-type: none"> - Criar funcionalidade de edição do cadastro de quartos. - Criar interfaces para gestão de quartos. - Implementar funcionalidade de alteração manual do status (disponível, indisponível, em manutenção) do quarto. - Criar funcionalidade de cadastro de hóspedes. - Criar funcionalidade de exclusão do cadastro de hóspedes. - Criar funcionalidade de edição do cadastro de hóspedes. - Criar interfaces para gestão de hóspedes. - Consolidar a documentação técnica do MVP.
6	17/06/2025 a 24/06/2025	Concluir as funcionalidades principais de reservas, check-in/check-out e integração para a entrega do MVP	<ul style="list-style-type: none"> - Criar funcionalidade de cadastro de reservas. - Criar funcionalidade de exclusão do cadastro de reservas. - Criar funcionalidade de edição do cadastro de reservas. - Criar interfaces para gestão de reservas. - Criar lógica de validação de disponibilidade de quartos para reservas. - Implementar atualização automática do status do quarto para “ocupado” após o check-in. - Criar funcionalidade de atualização do check-out. - Integrar ambientes, back-end e front-end. - Documentar a configuração e o código desenvolvido (Focado no MVP).

Fonte: Elaborado pelos autores.

Quadro 8: Sprints Backlog - Parte 2

Sprint	Período	Objetivo	Atividades
7	12/08/2025 a 26/08/2025	Começar o desenvolvimento das funcionalidades de gestão financeira	<ul style="list-style-type: none"> - Criar funcionalidade de cadastro de receitas. - Criar funcionalidade de exclusão do cadastro de receitas. - Criar funcionalidade de edição do cadastro de receitas. - Criar interface para gestão de receitas. - Criar funcionalidade de cadastro de despesas. - Criar funcionalidade de exclusão do cadastro de despesas.
8	26/08/2025 a 10/09/2025	Finalizar o CRUD de despesas e adicionar funcionalidades importantes para a gestão financeira	<ul style="list-style-type: none"> - Criar funcionalidade de edição do cadastro de despesas. - Criar interface para gestão de despesas. - Aprimorar cadastro de despesas para incluir tipo (fixo/variável). - Criar funcionalidade para registrar comprovação de pagamento de reservas. - Implementar criptografia. - Documentar a configuração e o código desenvolvido (Integrar ao do MVP).
9	10/09/2025 a 25/09/2025	Implementar as funcionalidades associadas aos relatórios financeiros e a funcionalidade de filtragem	<ul style="list-style-type: none"> - Criar funcionalidade para criar relatórios de quartos. - Criar funcionalidade para criar relatórios de hóspedes. - Criar funcionalidade para criar relatórios de reservas. - Criar balanço financeiro simples (receitas, despesas, saldo) por período. - Criar funcionalidade de filtragem de receita/despesa (data, categoria).

Fonte: Elaborado pelos autores.

Quadro 9: Sprints Backlog - Parte 3

Sprint	Período	Objetivo	Atividades
10	25/09/2025 a 10/10/2025	Implementar criptografia e preparar o ambiente para os testes de segurança	<ul style="list-style-type: none"> - Implementar criptografia. - Definir escopo dos testes. - Identificar cenários de teste. - Elaborar casos de teste. - Definir as ferramentas de teste. - Estabelecer ambiente de teste. - Definir critérios de aceitação. - Preparar dados de testes.
11	11/10/2025 a 25/10/2025	Executar testes gerais e de segurança, e documentar os resultados	<ul style="list-style-type: none"> - Executar testes gerais. - Documentar resultados dos testes. - Executar testes SSL. - Analisar e otimizar Headers de segurança. - Realizar ajustes de segurança.
12	26/10/2025 a 09/11/2025	Finalizar a documentação do projeto e planejar a implantação	<ul style="list-style-type: none"> - Executar testes gerais. - Documentar componentes e estilos - Documentar o plano e a execução de testes. - Registrar escolhas e mudanças de rumo. - Documentar problemas ocorridos e lições aprendidas. - Elaborar o plano de implantação.
13	10/11/2025 a 24/11/2025	Realizar a implantação do sistema e o treinamento da proprietária	<ul style="list-style-type: none"> - Realizar a implantação da aplicação. - Revisão final da documentação técnica. - Treinamento da proprietária da pousada para uso da aplicação. - Entrega do produto final

Fonte: Elaborado pelos autores.

Quadro 10: Sprints Backlog - Parte 4

3.4 Gestão de Riscos

Diante da flexibilidade e do incentivo à melhoria contínua proporcionados pela metodologia *Scrum*, a equipe do projeto considerou pertinente a implementação de mecanismos de apoio à gestão de riscos, com o objetivo de fortalecer as práticas de inspeção e adaptação realizadas ao longo das *sprints*.

Segundo (SOMMERVILLE, 2011, p. 416), o gerenciamento de riscos é um processo iterativo fundamental para prever os riscos associados ao desenvolvimento de um projeto,

pois promove a compreensão dos riscos com vistas à sua previsão, detecção e tratamento, de modo a não comprometer a entrega do produto final.

Ainda segundo Sommerville ([SOMMERVILLE, 2011](#), p. 416), os riscos podem ser definidos como elementos ou eventos de origem multifatorial que, caso ocorram, impactam negativamente o cronograma, os custos, a qualidade ou o escopo do projeto.

Nesse sentido, a gestão de riscos foi uma atividade de suma importância para o desenvolvimento do sistema web, pois forneceu mecanismos que auxiliaram a equipe a identificar precocemente possíveis problemas, compreendê-los e tratá-los de maneira eficaz durante todo o desenvolvimento.

Dessa forma, a equipe do projeto do desenvolvimento do sistema de gestão da Pousada Chalés Água de Coco elaborou um processo de identificação dos riscos iniciais do projeto e, a partir disso, definiu estratégias de mitigação para cada um deles, com o objetivo de garantir o cumprimento dos prazos estabelecidos e evitar retrabalho.

3.4.1 Identificação dos Riscos do Projeto

O primeiro passo para o desenvolvimento das ações voltadas à gestão de riscos foi a identificação dos principais riscos associados ao projeto do sistema web de gestão da Pousada Chalés Água de Coco. Esses riscos foram levantados a partir das características inerentes ao desenvolvimento de uma aplicação *web*, da equipe *Scrum* e do próprio negócio (??).

Código	Risco	Afeta	Descrição
R01	Requisitos mal definidos ou incompletos	Projeto	Os requisitos levantados são inconsistentes, exigindo retrabalho e falhas na produção
R02	Atraso no cronograma	Projeto	Comunicação deficiente ou dificuldades nas entregas geram atrasos no cronograma produção
R03	Perda de dados ou inconsistência	Projeto e Produto	Há falhas na modelagem ou na lógica do negócio que resultam em falhas de produção

Fonte: Elaborado pelos autores.

Quadro 11: Identificação dos Riscos do Projeto - Parte 1

Código	Risco	Afeta	Descrição
R04	Tamanho do Projeto Subestimado	Projeto e Produto	A equipe não conseguiu dimensionar o trabalho exigido pelo projeto o que exige um retrabalho de planejamento das tarefas
R05	Prazo de Desenvolvimento Subestimado	Projeto	A equipe encontrou obstáculos maiores que o previsto durante o desenvolvimento do projeto que exigem maior tempo para o desenvolvimento
R06	Integrantes chaves estão doentes em momentos críticos do projeto	Projeto	A ausência de um integrante importante para a realização de uma tarefa exige que os outros integrantes tenham que assumir sem dominação do assunto
R07	Vazamento de dados de hóspedes	Negócio	O sistema armazena dados pessoais e sensíveis e não possui uma estrutura de segurança fortificada
R08	Custos do projeto subestimados	Projeto e Produto	Há custos adicionais não previstos no planejamento
R09	Risco de integração de módulos (Django)	Projeto e Produto	A equipe enfrenta dificuldades em integrar as entidades do sistema decorrente de um mau planejamento de comunicação entre elas, segundo a arquitetura Django (MVT)
R10	Problemas de segurança no ambiente de produção	Produto	A implantação do sistema expõe falhas de segurança não identificados durante o desenvolvimento

Fonte: Elaborado pelos autores.

Quadro 12: Identificação dos Riscos do Projeto - Parte 2

3.4.2 Análise e Planejamento dos Riscos

Identificados os riscos associados ao projeto, a equipe analisou e definiu a probabilidade de impacto e de ocorrência, além das estratégias de prevenção e contingência para cada um. Essas informações estão registradas no [Quadro 13](#).

Risco	Impacto	Probabilidade de Ocorrência	Estratégias de Prevenção	Estratégias de Contingência
R01	Alto	Alta	Fazer revisões e validações periódicas dos requisitos, conforme o andamento do projeto	Interromper as atividades de desenvolvimento para revisão dos requisitos com o time de desenvolvimento e o <i>product owner</i>
R02	Médio	Alta	Monitorar o andamento das tarefas do projeto e solucionar problemas que possam vir a impactar o andamento do cronograma	Reavaliar o <i>backlog</i> das <i>sprints</i> restantes e reorganizar as tarefas, priorizando a execução daquelas que são essenciais para a entrega final
R03	Alto	Alta	Revisar e validar os elementos da modelagem de dados antes de iniciar o desenvolvimento; Definir lógicas de salvamento e atualização de dados	Revisar os documentos da modelagem de dados; identificar e corrigir as falhas na lógica do negócio
R04	Alto	Baixa	Definir claramente as funcionalidades do sistema, evitando a adição de requisitos que ampliem o sistema para além do planejado	Priorizar a entrega das funcionalidades definidas nos requisitos iniciais do projeto
R05	Médio	Alta	Definir junto a com o time de desenvolvimento o número de <i>sprints</i> viáveis para o desenvolvimento; Monitorar as atividades e obstáculos nas <i>sprints</i> de desenvolvimento para prever possíveis atrasos no cronograma	Reorganizar tarefas das <i>sprints</i> de desenvolvimento, priorizando a realização de tarefas associadas as funcionalidades essenciais do sistema

Fonte: Elaborado pelos autores.

Quadro 13: Análise e Planejamento dos Riscos - Parte 1

Risco	Impacto	Probabilidade de Ocorrência	Estratégias de Prevenção	Estratégias de Contingência
R06	Médio	Baixa	Distribuir as tarefas do projeto de forma estratégica para que em nenhum momento o projeto fique altamente dependente de um integrante, além de desenvolver uma documentação robusta para que outro integrante consiga dar continuidade a tarefa	Remanejar as tarefas entre os integrantes disponíveis
R07	Crítico	Baixa	Estabelecer medidas de segurança e criptografia no banco de dados do projeto	Desconectar o sistema; identificar a origem do vazamento; realizar medidas de contenção de danos propostas pela LGPD; aplicar e documentar as correções de segurança
R08	Médio	Média	Desenvolver um bom levantamento de custos antes de iniciar o projeto	Revisar funcionalidades e priorizar as funcionalidades que caibam no orçamento disponível
R09	Alto	Média	Realizar validações dos diagramas e o MER do sistema antes e durante a fase de desenvolvimento do sistema	Revisar os diagramas e o MER; Identificar falhas de integração entre os módulos; Corrigir as falhas
R10	Alto	Média	Definir um plano de testes que abranja os diferentes aspectos e funcionalidades do sistema	Desconectar o sistema; identificar a origem do problema de segurança; corrigir e documentar o erro

Fonte: Elaborado pelos autores.

Quadro 14: Análise e Planejamento dos Riscos - Parte 2

3.4.3 Monitoramento dos Riscos

Levantados os riscos do projeto do sistema web e definidas as estratégias de prevenção e resposta a cada um, foi elaborado um conjunto de elementos para auxiliarem no monitoramento contínuo dos riscos. Esse conjunto está registrado na [Quadro 15](#), no qual estão os indicadores de risco, os mecanismos que devem ser utilizados na identificação de alterações, sendo que esses elementos devem ser monitorados a cada *sprint* realizada, durante as reuniões pós *sprint*.

Risco	Indicadores	Métricas
R01	<ul style="list-style-type: none"> - Time <i>scrum</i> frequentemente tem dúvidas sobre o que deve ser feito. - Solicitações frequentes de adição de novas funcionalidades 	<ul style="list-style-type: none"> - Número de dúvidas abertas durante as reuniões de desenvolvimento.
R02	<ul style="list-style-type: none"> - Constante adiamento de entregas. - Tempo de desenvolvimento de tarefas maior do que o previsto. 	<ul style="list-style-type: none"> - Porcentagem de tarefas entregues por <i>sprint</i>.
R03	<ul style="list-style-type: none"> - Carência de backups. - Carência de testes de restauração. 	<ul style="list-style-type: none"> - Frequência (tempo médio) de realização de <i>backups</i>. - Frequência (tempo médio) de falhas registradas nos testes de banco de dados.
R04	<ul style="list-style-type: none"> - Desenvolvimento das funcionalidades está mais complexo que o esperado. - Aumento de tarefas não previstas nas <i>sprints backlogs</i>. 	<ul style="list-style-type: none"> - Número de novas tarefas adicionadas nas <i>sprints backlogs</i>.
R05	<ul style="list-style-type: none"> - Tarefas sendo remanejadas constantemente para futuras <i>sprints</i>. 	<ul style="list-style-type: none"> - Número de remanejamento de tarefas por <i>sprints</i>.
R06	<ul style="list-style-type: none"> - Carência de comunicação entre os membros do time. - Carência de documentação e <i>code review</i> para facilitar substituições em tarefas. 	<ul style="list-style-type: none"> - Número da concentração de tarefas por integrante.
R07	<ul style="list-style-type: none"> - Dados sensíveis armazenados sem criptografia. - Carência de testes de segurança. - Códigos com permissões e bibliotecas mal configurados. 	<ul style="list-style-type: none"> - Número do uso de dados sensíveis sem criptografia. - Número de inconformidades com a LGPD (<i>checklist</i>).
R08	<ul style="list-style-type: none"> - Gastos com ferramentas para funcionalidades não previstas 	<ul style="list-style-type: none"> - Número de recursos contratados fora do planejamento inicial.
R09	<ul style="list-style-type: none"> - Módulos desenvolvidos em paralelo sem o uso de práticas de integração contínua. - Carência de testes de integração. - Modelos mal padronizados. 	<ul style="list-style-type: none"> - Número de <i>bugs</i> relacionados à comunicação entre módulos.
R10	<ul style="list-style-type: none"> - Carência de mecanismos como HTTPS e <i>headers</i> de segurança. - Dados sensíveis expostos em variáveis de ambiente. 	<ul style="list-style-type: none"> - Número de falhas identificadas durante testes de segurança.

Fonte: Elaborado pelos autores.

Quadro 15: Mecanismos de Monitoramento dos Riscos - Parte 1

Portanto, dado seu caráter fiscalizatório, a aplicação dos mecanismos de gestão

de riscos no desenvolvimento do sistema *web* da Pousada Chalés Água de Coco assegura a capacidade de inspeção e adaptação contínua do projeto. De forma tal, a contribuir significativamente para a entrega de um produto final seguro e alinhado com os requisitos e regras do negócio.

3.5 Repositório da Aplicação

Nesta seção é apresentada a estrutura do ambiente de desenvolvimento do sistema *web* de gestão da Pousada Chalés Água de Coco. O código-fonte do sistema foi estruturado em aplicações modulares, arquivos de configuração geral do ambiente *Django* e arquivos auxiliares. Essa configuração segue as boas práticas do *framework*, promovendo baixo acoplamento e alta coesão, o que garante um bom encapsulamento entre os módulos. Os códigos foram mantidos simples e otimizados para aproveitar as características dinâmicas do *Python* e as boas práticas do *framework*. Além disso, foram aplicadas boas práticas de organização e arquitetura para garantir facilidade de manutenção e evolução do sistema.

3.5.1 Definição do repositório da aplicação

A estrutura dos diretórios e arquivos do sistema foi definida seguindo o padrão recomendado pelo *Django*. Essa organização visa facilitar o desenvolvimento, manutenção e segurança do sistema.

1. **/code/core/setup:** Armazena os scripts de configuração do *framework* do sistema *Django*.
 - a) **settings.py:** Define o diretório base do projeto (`BASE_DIR`), configura variáveis sensíveis como `SECRET_KEY`, modo de *debug*, domínios autorizados (`ALLOWED_HOSTS`), lista de aplicativos instalados (`INSTALLED_APPS`), *middlewares*, arquivos de URLs, sistema de templates e banco de dados (*PostgreSQL*).
 - b) **urls.py:** Define as rotas *URLs* do projeto.
 - c) **wsgi.py:** Arquivo padrão para *deploy* da aplicação via servidor WSGI (*Web Server Gateway Interface*). Define a *callable application* para a comunicação entre o servidor WSGI e o servidor *Django*.
 - d) **manage.py:** Arquivo de execução dos comandos administrativos do *Django* via terminal. Define a configuração padrão, importa e executa o *Django*.
2. **/code/core/apps:** Armazena as aplicações internas do sistema, sendo que cada aplicação representa um módulo do sistema.

- a) **/code/core/hóspede:** Armazena os arquivos relativos às funcionalidades do módulo de gestão de hóspedes, são eles:
 - i. **models.py:** Define a estrutura da tabela “Hóspede” no banco de dados.
 - ii. **forms.py:** Define os formulários baseados no modelo “Hóspede”. É usado para criar e editar objetos do tipo “Hóspede” na *interface web*.
 - iii. **urls.py:** Faz o mapeamento das URLs da aplicação “Hóspedes”.
 - iv. **views.py:** Possui as funções do *app* “Hóspede”, como: `hospede_list`, `hospede_create` e `hospede_update`.
 - b) **/code/core/quarto:** Armazena os arquivos relativos às funcionalidades do módulo de gestão de quartos, são eles:
 - i. **models.py:** Define a estrutura da tabela “Quarto” no banco de dados.
 - ii. **forms.py:** Define os formulários baseados no modelo “Quarto”, usados para atividades como criar e editar quartos na *interface web*. Utiliza *widgets* para estilizar *inputs* e *labels*.
 - iii. **urls.py:** Faz o mapeamento das URLs da aplicação “Quartos”, como: listar os quartos e abrir o formulário de criação de um objeto do tipo “Quarto”.
 - iv. **views.py:** Possui as funções do *app* “Quarto”, como: `excluir_quarto` e `tipos_quarto`.
3. **/code/core/templates:** Armazena os arquivos *HTML* do sistema.
- a) **/code/setup/templates/shared/base.html:** *Template* utilizado como estrutura principal de todas as páginas do sistema, definindo um *layout* comum. Utiliza ícones do *FontAwesome* e é estilizado pelo arquivo *output.css*, criado com o *Tailwind CSS* e *DaisyUI*.
4. **/code/setup/static/css/output.css:** *CSS* gerado pela biblioteca *Tailwind CSS* com base nas configurações do sistema. Contém variáveis, estilos base e camadas para definir cores, espaçamentos, tamanhos de fonte, *padding*s, bordas e define comportamentos padrão.

Essa estrutura foi armazenada em um repositório remoto no *Git Hub* ([SOUSA et al., 2025](#)), para facilitar o gerenciamento, a colaboração entre os desenvolvedores e o controle do versionamento do código-fonte do sistema de gestão.

4 Desenvolvimento do Projeto

Nesta seção são detalhadas as tecnologias, ferramentas e práticas adotadas para o desenvolvimento da aplicação *web* de gestão dos processos da pousada Chalés Água de Coco.

4.1 Escopo do Projeto

Esse projeto visa automatizar os processos de gestão de reservas, quartos, hóspedes, finanças e produção de relatórios da pousada, com o objetivo de proporcionar à proprietária uma ferramenta eficiente e organizada para a realização de todas as operações essenciais do seu negócio. Além disso, busca mitigar os riscos associados ao modelo atual de administração, caracterizado por uma alta vulnerabilidade a erros humanos, dificuldade de atualização em tempo real e ausência de acessibilidade remota, decorrentes do seu caráter predominantemente manual.

O escopo deste projeto prevê que o produto final seja constituído por funcionalidades organizadas em cinco módulos principais: módulo de gestão de reservas, módulo de gestão de hóspedes, módulo de gestão de quartos, módulo de gestão financeira e painel administrativo. Com uma interface intuitiva e integrada e desenvolvido exclusivamente para o uso de uma única usuária: a proprietária da pousada.

No entanto, o escopo do projeto não abrange funcionalidades como a integração da aplicação com sistemas de pagamento ou serviços de hospedagem. Ainda que não exclua a possibilidade de que, em etapas futuras, o sistema seja atualizado para a implementação de integrações com serviços de mensageria, com o objetivo de ampliar a automação dos serviços da pousada.

Por fim, Para a definição do escopo, a equipe responsável aplicou um conjunto de perguntas ([Quadro 16](#)) à proprietária com o objetivo de compreender as regras de negócio e realizar o levantamento dos requisitos, funcionais e não funcionais, do sistema.

Pergunta	Resposta
Há um prazo mínimo ou máximo para fazer uma reserva?	Preferencialmente, antecipadamente. Nas plataformas coloco 2 dias de antecedência.
A reserva é confirmada apenas com pagamento ou pode ser feita sem pagamento antecipado?	Confirmada pelo pagamento de pelo menos metade do valor da reserva.
É permitido cancelar uma reserva? Até quantas horas antes do check-in? Há cobrança de taxa?	Sim! Temos políticas de cancelamentos e remarcações.
Um hóspede pode fazer mais de uma reserva ativa ao mesmo tempo?	Sim.
Quais dados do hóspede são obrigatórios para fazer uma reserva?	Nome completo, endereço completo, CPF, telefone, e-mail.
É comum haver reservas feitas por um responsável em nome de outros hóspedes?	Sim.
Há um limite de pessoas por quarto? Como isso é controlado?	No check in.
Qual é o horário padrão de check-in e check-out? Há tolerância?	Check in a partir das 16h até às 22h e check-out a partir das 8h até às 14h. Depende de se tem entrada de outro hóspede em seguida.
O check-in pode ser feito antes do horário? E o check-out após o horário?	Depende, se tiver saída de hóspede anterior. Depende se tiver entrada em seguida.
Quem realiza o check-in e check-out? Você ou os funcionários?	Eu ou sozinhos, com orientações minhas.
Há necessidade de gerar comprovante ou recibo após check-in ou check-out?	Não.
A pousada possui quantos quartos? Como eles são classificados (ex: simples, casal, com ar)?	16. Quartos simples simples para casal (com e sem ar), chalés com cozinha para até 4 pessoas, com e sem ar e flats, com e sem ar.
Há períodos em que quartos são bloqueados para manutenção?	Sim.
Um mesmo quarto pode ser reservado para diferentes hóspedes em dias seguidos?	Sim.
A pousada oferece serviços extras?	Não.
Quais formas de pagamento são aceitas (Pix, cartão, dinheiro)?	As três formas, porém no cartão tem taxa da operadora.

Fonte: Elaborado pelos Autores.

Quadro 16: Questionário Aplicado à Proprietária - Parte 1

Pergunta	Resposta
Os pagamentos são feitos no check-in, no check-out ou antecipadamente?	Metade na reserva e o restante na chegada
É necessário gerar comprovante ou recibo no sistema?	Eu envio uma confirmação de reserva com todas as informações.
Você gostaria que o sistema enviasse confirmação automática de reserva por WhatsApp ou e-mail?	Sim.
Há interesse em receber alertas automáticos de check-in, check-out ou cancelamento?	Sim.
Somente você vai usar o sistema ou os funcionários também?	Somente eu.
O sistema será usado no computador, celular ou ambos?	Em ambos.
Quais relatórios são mais importantes no dia a dia? (ex: ocupação diária, reservas da semana, totais de pagamento)	Ocupação diária, reservas da semana, totais de pagamento, período. . .
É importante ter um histórico de cada hóspede e das reservas anteriores?	Sim.

Fonte: Elaborado pelos Autores.

Quadro 17: Questionário Aplicado à Proprietária - Parte 3

4.1.1 Regras de Negócio

As regras de negócio são diretrizes que delimitam as relações entre os objetos do negócio, estabelecendo condições e restrições que orientam as empresas nas suas operações e processos internos (IBM, 2025).

Por isso, o levantamento das regras de negócio da Pousada Chalés Água de Coco, resultado da análise das respostas fornecidas pela proprietária ao questionário aplicado e da análise do site institucional da pousada (COCO, 2025) pela equipe, foi fundamental para orientar a modelagem e o desenvolvimento das funcionalidades do sistema. Além de assegurar que a aplicação atenda as necessidades da pousada de forma coerente, segura e eficiente, e apoiar, no contexto de sua gestão, os processos de tomada de decisão.

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	Cada quarto deve ter um número máximo de hóspedes.
RN08	Os horários padrão de check-in e check-out são: check-in das 16h às 22h; check-out das 8h às 14h. Em períodos de temporada os horários mudam para: check-in das 15h às 22h; check-out das 8h às 11h.
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	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.
RN16	As tarifas das reservas podem sofrer alterações conforme: demanda, datas especiais, período de antecedência da reserva, quantidade de diárias, tipo de acomodação e quantidade de pessoas.
RN17	São permitidos early check-in e late check-out com cobrança de tarifas adicionais.
RN18	Toda a gestão financeira da pousada deve ser realizada dentro do sistema web de gestão.

Fonte: Elaborado pelos autores.

Quadro 18: Regras de Negócio

4.1.2 Requisitos do Sistema

Segundo Sommerville (SOMMERVILLE, 2011, p.57), os requisitos de um sistema são descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Assim, considerando a importância desses elementos para a modelagem da aplicação web de gestão para a Pousada Chalés Água de Coco, foi realizado o levantamento dos requisitos funcionais e não funcionais. Esse processo é resultante da análise das respostas obtidas por meio do questionário aplicado à proprietária (Quadro 16) e das informações disponibilizadas no site institucional da empresa (COCO, 2025).

4.1.2.1 Requisitos Funcionais

Os requisitos funcionais são aqueles que descrevem as funcionalidades e serviços do sistema *web* de gestão da pousada. Tais requisitos estão listados no Quadro 19.

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF01	A aplicação deve possuir um sistema de login para a proprietária acessar a aplicação de forma segura	Alta	RN15
RF02	O sistema deve permitir que a proprietária altere sua senha de acesso	Alta	RN15
RF03	O sistema deve permitir que a proprietária recupere sua senha via <i>e-mail</i>	Alta	RN15
RF04	O sistema deve permitir o cadastro de novas reservas, associando um quarto a um período (data de check-in e check-out)	Alta	RN04; RN11
RF05	O sistema deve permitir a edição de reservas já cadastradas	Alta	RN02
RF06	O sistema deve impedir o cadastro de reservas que não estejam associadas a quartos disponíveis	Alta	RN07; RN10

Fonte: Elaborado pelos autores.

Quadro 19: Requisitos Funcionais - Parte 1

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF07	O sistema deve exigir os dados pessoais do hóspede para que a reserva seja cadastrada: nome completo, endereço completo, CPF/Passaporte, telefone e e-mail	Alta	RN05
RF08	O sistema deve permitir que a proprietária reserve quartos para um hóspede em nome de outra pessoa responsável, registrando os dados do hóspede e do responsável	Média	RN06
RF09	O sistema deve exigir o pagamento de 50% do valor da estadia para confirmar o cadastro da reserva (a ser pago no momento da reserva ou em um prazo definido)	Média	RN01;RN13
RF10	O sistema deve permitir o registro de pagamento da reserva (forma, valor, data)	Média	RN01;RN13
RF11	A proprietária deve conseguir cancelar ou alterar uma reserva, com possível registro do motivo	Média	RN02
RF12	A proprietária deve conseguir cadastrar mais de uma reserva no nome de um mesmo hóspede	Média	RN04
RF13	A proprietária deve conseguir reservar um mesmo quarto para diferentes clientes em datas seguidas, respeitando os horários de <i>check-in</i> e <i>check-out</i> configurados para o quarto	Alta	RN08; RN11
RF14	A proprietária deve conseguir acessar o histórico de reservas de um hóspede	Média	RN04

Fonte: Elaborado pelos autores.

Quadro 20: Requisitos Funcionais - Parte 2

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF15	O sistema deve permitir configurar tarifas de reserva com base nas regras de negócio da pousada	Média	RN16
RF16	O sistema deve enviar uma notificação para o hóspede via <i>e-mail</i> após a confirmação da reserva	Baixa	RN14
RF17	O sistema deve impedir o registro de reservas com menos de 2 dias de antecedência da data do <i>check-in</i>	Média	RN03
RF18	O sistema deve permitir a edição dos dados de hóspedes já cadastrados	Média	RN05
RF19	A proprietária deve poder fazer o cadastro de quartos, incluindo informações como número/nome do quarto, capacidade (número de hóspedes), tipo (ex: chalé, simples solteiro, simples casal, etc.), e preço por noite	Alta	RN07; RN10
RF20	AA proprietária deve conseguir editar as informações dos quartos já cadastrados	Alta	RN07; RN10
RF21	O sistema deve permitir a visualização dos quartos disponíveis no período de tempo selecionado para a reserva	Média	RN07; RN11
RF22	A proprietária deve conseguir mudar o status de um quarto (ex: disponível, ocupado, em manutenção) manualmente, se necessário	Alta	RN10

Fonte: Elaborado pelos autores.

Quadro 21: Requisitos Funcionais - Parte 3

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF23	O sistema deve gerar relatórios de ocupação de quartos em períodos definidos	Média	RN04; RN07; RN08; RN10; RN11
RF24	O sistema deve mudar o status do quarto quando for realizado o registro do <i>check-out</i>	Alta	RN08; RN10
RF25	O sistema deve mudar o status do quarto de reservado para ocupado no registro do <i>check-in</i>	Alta	RN08; RN10
RF26	O sistema deve permitir à proprietária configurar os horários padrão de <i>check-in</i> e <i>check-out</i>	Alta	RN08
RF27	O sistema deve permitir a geração de recibos ou comprovantes de pagamento e estadia, sob demanda	Média	RN09
RF28	O sistema deve permitir o registro de <i>early check-in</i> e <i>late check-out</i> para uma reserva	Média	RN17
RF29	O sistema deve permitir a configuração de tarifas adicionais para <i>early check-in</i> e <i>late check-out</i>	Média	RN17
RF30	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	RN18
RF31	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	RN18

Fonte: Elaborado pelos autores.

Quadro 22: Requisitos Funcionais - Parte 4

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF32	O sistema deve permitir a edição das transações financeiras registradas (receitas e despesas)	Média	RN18
RF33	O sistema deve permitir a exclusão das transações financeiras registradas	Média	RN18
RF34	O sistema deve permitir que a proprietária visualize todas as transações financeiras (receitas e despesas) em um determinado período	Média	RN18
RF35	O sistema deve permitir a filtragem das transações por tipo (receita/despesa), data e categoria	Média	RN18
RF36	O sistema deve gerar relatórios financeiros detalhados automaticamente por período e por categoria	Média	RN18
RF37	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	RN18
RF38	O sistema deve gerar relatórios de faturamento por período	Média	RN18
RF39	O sistema deve apresentar um painel (dashboard) com métricas-chave da pousada	Média	Requisito essencial para a gestão e visualização do negócio
RF40	O sistema deve permitir o envio de notificações automáticas à proprietária sobre eventos importantes (ex: <i>check-ins</i> iminentes)	Baixa	Requisito de suporte à gestão operacional

Fonte: Elaborado pelos autores.

Quadro 23: Requisitos Funcionais - Parte 5

4.1.2.2 Requisitos Não Funcionais

Os requisitos não funcionais descrevem, por sua vez, as restrições e características de qualidade que devem ser aplicadas às funções e serviços prestados pelo sistema web.

Estes requisitos estão listados no [Quadro 24](#)

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 24: 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 25: Requisitos Não Funcionais - Parte 2

4.2 História de Usuário

As histórias de usuário são descrições curtas e informais das funcionalidades esperadas do sistema, sob a perspectiva do usuário final. Dessa forma, elas possuem alto valor para o desenvolvimento do escopo, visto que, auxiliam a equipe a compreender e priorizar as necessidades reais dos usuários. Por isso, foram levantadas e listadas as histórias de usuário para a aplicação web de gestão da Pousada Chalés Água de Coco neste tópico, como parte da definição e organização das funcionalidades do sistema.

4.2.1 Descrição das Histórias de Usuário

A seguir, apresentam-se as descrições das histórias de usuário levantadas para o desenvolvimento do sistema.

1. Acesso ao Sistema:

- **Descrição:** Como proprietária, quero fazer login com *ee-mail* e senha para acessar o sistema de forma segura.
- **Critérios de Aceitação:**
 - A proprietária não consegue acessar o sistema caso apresente credenciais inválidas.
 - O acesso é protegido por criptografia e sessão segura (HTTPS).
- **Requisito Funcional Relacionado:** RF01.

2. Alteração de Senha:

- **Descrição:** Como proprietária, quero alterar minha senha para manter minha conta segura.
- **Critérios de Aceitação:**
 - A proprietária só consegue alterar a senha após *login* bem sucedido.
 - O sistema força a proprietária a criar uma nova senha forte.
 - O sistema exibe mensagem de sucesso após o cadastro da nova senha.
- **Requisito Funcional Relacionado:** RF02.

3. Recuperação de Senha:

- **Descrição:** Como proprietária, quero recuperar minha senha caso eu a esqueça, para conseguir acesso ao sistema novamente.
- **Critérios de Aceitação:**
 - A proprietária consegue solicitar recuperação de senha via *e-mail*.

- A proprietária recebe *e-mail* com um link de recuperação temporário.

- **Requisito Funcional Relacionado:** RF03.

4. Cadastro de Reservas:

- **Descrição:** Como proprietária, quero cadastrar novas reservas no sistema para ter maior controle das hospedagens.
- **Critérios de Aceitação:**
 - O sistema fornece um formulário para cadastrar a reserva.
 - A reserva é cadastrada após o envio do formulário.
 - O sistema exibe uma mensagem de confirmação do cadastro
 - Os dados da reserva são salvos no banco de dados.
- **Requisito Funcional Relacionado:** RF04.

5. Edição de Reservas:

- **Descrição:** Como proprietária, quero editar reservas já cadastradas, para que eu possa alterar ou corrigir informações importantes.
- **Critérios de Aceitação:**
 - O sistema permite que a proprietária altere os campos do formulário de reserva.
 - A proprietária consegue salvar a reserva com os novos dados.
 - Os dados são atualizados no banco de dados.
- **Requisito Funcional Relacionado:** RF05.

6. Bloqueio do Cadastro de Reservas:

- **Descrição:** Como proprietária, quero que eu seja impedida de cadastrar reservas de quartos já ocupados para evitar conflitos de hospedagem.
- **Critérios de Aceitação:**
 - O sistema impede que a proprietária cadastre a reserva de um quarto que tenha o status diferente de “disponível” na data selecionada.
 - O sistema exibe uma mensagem de alerta avisando que o quarto não está disponível.
- **Requisito Funcional Relacionado:** RF06.

7. Reserva com Dados do Hóspede:

- **Descrição:** Como proprietária, quero registrar os dados pessoais de cada hóspede ao criar ou gerenciar uma reserva, para manter um registro completo e atualizado dos hóspedes.
- **Critérios de Aceitação:**
 - No formulário de cadastro de reserva devem ser exigidos o preenchimento de campos com os dados do hóspede (nome completo, endereço completo, CPF/Passaporte, telefone e *e-mail*).
 - Todos os campos devem ter validação de formato.
 - A proprietária deve conseguir buscar e selecionar hóspedes já cadastrados para novas reservas.
 - O sistema deve validar os campos obrigatórios para garantir a integridade dos dados.
 - Os dados dos hóspedes devem ser armazenados de forma segura.
- **Requisito Funcional Relacionado:** RF07.

8. Reserva em Nome de um Responsável:

- **Descrição:** Como proprietária, quero cadastrar uma reserva em nome de um responsável pelo hóspede para atender situações em que outra pessoa cuida da reserva.
- **Critérios de Aceitação:**
 - O formulário de cadastro de reserva deve ter uma opção para indicar que a reserva está sendo feita por um responsável.
 - Opção selecionada: O sistema exibe os campos para registro dos dados pessoais do responsável.
 - Opção selecionada: O sistema mantém os campos para registro dos hóspedes.
 - Após concluído o cadastro da reserva o sistema deve deixar visível quem é o responsável e quem é o hóspede.
- **Requisito Funcional Relacionado:** RF08.

9. Pagamento de Confirmação de Reserva:

- **Descrição:** Como proprietária, quero registrar o pagamento da estadia para confirmar a reserva e manter a organização das receitas da pousada.
- **Critérios de Aceitação:**

- No cadastro da reserva: O sistema calcula e exibe automaticamente o valor correspondente a 50% da estadia.
- O sistema permite o cadastro do pagamento parcial da estadia.
- Caso o pagamento não seja registrado durante o cadastro da reserva, o sistema permite que a proprietária defina um prazo para o registro.
- O registro do pagamento deve atualizar o status da reserva para “Confirmada”, sendo ele o registro parcial ou completo do pagamento.
- O sistema permite que a proprietária anexe um comprovante de pagamento ao registro.
- Após o registro do pagamento, o sistema gera um comprovante de reserva e pagamento, sob demanda da proprietária.

- **Requisito Funcional Relacionado:** RF09, RF10 e RF27.

10. Cancelamento ou Alteração de Reservas:

- **Descrição:** Como proprietária, quero cancelar ou remarcar uma reserva, com a possibilidade de registrar o motivo para manter o controle sobre mudanças.
- **CrITÉRIOS de Aceitação:**
 - O sistema permite que a proprietária selecione uma reserva cadastrada e escolha entre alterar ou cancelar.
 - Proprietária seleciona cancelar: sistema disponibiliza formulário com campo opcional para o registro do motivo do cancelamento.
 - Proprietária seleciona alterar: sistema disponibiliza o formulário de cadastro da reserva para a proprietária fazer as alterações.
 - Proprietária seleciona alterar: Novas alterações são salvas no banco de dados.
 - Novo status da reserva é salvo no banco de dados.
 - Proprietária seleciona cancelar: o status do quarto ocupado na reserva é atualizado para “Disponível”.

- **Requisito Funcional Relacionado:** RF11.

11. Cadastro de Múltiplas Reservas:

- **Descrição:** Como proprietária, quero cadastrar múltiplas reservas para um mesmo hóspede, para facilitar reservas recorrentes.
- **CrITÉRIOS de Aceitação:**
 - O sistema permite que a proprietária selecione um cliente já cadastrado ao iniciar um novo cadastro de reserva.

- O sistema permite que mais de uma reserva ativa esteja associada a um mesmo hóspede (id).
- Cada reserva deve ser feita de forma separada.
- Cada reserva deve ser independente mesmo que seja do mesmo cliente.
- O sistema deve registrar separadamente o pagamento de cada reserva.

- **Requisito Funcional Relacionado:** RF12.

12. Cadastro de Múltiplas Reservas de Quarto:

- **Descrição:** Como proprietária, quero reservar um mesmo quarto para diferentes hóspedes em datas seguidas, para otimizar a ocupação dos quartos.
- **CrITÉRIOS de Aceitação:**
 - O sistema permite que um mesmo quarto seja associado a reservas consecutivas, desde que o horário de *check-in* da nova reserva seja posterior ao horário de *check-out* da reserva anterior.
 - Os horários de *check-in* e *check-out* devem estar configurados no sistema e devem ser respeitados durante o cadastro de reserva de um quarto.
 - O sistema deve impossibilitar que a proprietária faça reservas que se sobreponham parcial ou totalmente ao período de outra reserva já associada ao quarto.
- **Requisito Funcional Relacionado:** RF13.

13. Acesso de Histórico de Reservas:

- **Descrição:** Como proprietária, quero visualizar o histórico de reservas dos hóspedes, para entender suas preferências e frequência de reservas.
- **CrITÉRIOS de Aceitação:**
 - A funcionalidade de histórico de reservas deve estar disponível na página do módulo de reservas.
 - O sistema permite que a proprietária busque reservas através do hóspede ou do quarto.
 - Ao selecionar o id: sistema exibe lista com todas as reservas associadas, tenham sido elas canceladas, realizadas ou futuras.
 - A proprietária consegue acessar detalhes completos de cada reserva listada.
 - O histórico de reservas deve ser atualizado automaticamente a cada nova reserva ou alteração de reserva.
- **Requisito Funcional Relacionado:** RF14.

14. Configuração de Tarifas:

- **Descrição:** Como proprietária, quero configurar as tarifas de reserva no sistema para aplicar preços personalizados.
- **Critérios de Aceitação:**
 - O sistema permite que a proprietária defina o preço por noite para cada tipo de quarto.
 - O sistema permite que a proprietária configure tarifas diferenciadas por período (alta/baixa temporada).
 - O sistema aplica automaticamente as tarifas ao valor total de uma reserva.
 - O sistema permite que a proprietária edite e salve novas regras de tarifação sem apagar reservas anteriores.
- **Requisito Funcional Relacionado:** RF15.

15. Notificações de confirmação para clientes:

- **Descrição:** Como proprietária, quero que meus clientes recebam informações da confirmação da reserva via *e-mail*, para evitar os envios manuais para cada cliente.
- **Critérios de Aceitação:**
 - É realizado o registro do pagamento parcial ou total da reserva e o sistema marca a reserva como confirmada.
 - O sistema envia uma notificação automática ao hóspede, via *e-mail*.
 - O sistema registra que a notificação foi enviada.
 - Caso o envio da notificação falhe, o sistema deve alertar a proprietária para enviar manualmente.
- **Requisito Funcional Relacionado:** RF16.

16. Impedimento de Cadastro de Reserva:

- **Descrição:** Como proprietária, quero que o sistema impeça o cadastro de reservas com menos de 2 dias de antecedência para o *check-in*, para ter tempo suficiente para preparar a recepção do hóspede.
- **Critérios de Aceitação:**
 - Durante o cadastro da reserva, o sistema calcula a diferença entre a data atual e a data de *check-in* solicitada.
 - Caso a diferença seja menor que 2 dias, o sistema impede o cadastro e exibe uma mensagem de erro.

- **Requisito Funcional Relacionado:** RF17.

17. Edição do Cadastro de Hóspede:

- **Descrição:** Como proprietária, quero visualizar e editar os dados dos hóspedes existentes para manter as informações atualizadas.
- **CrITÉrios de Aceitação:**
 - O sistema tem um módulo de gestão de hóspedes, no qual a proprietária consegue acessar uma lista dos hóspedes já cadastrados.
 - A proprietária consegue fazer buscas de hóspedes por id, nome, CPF ou telefone.
 - A proprietária consegue selecionar um hóspede e visualizar seus dados.
 - O sistema oferece uma opção de edição do cadastro do hóspede.
 - O sistema permite a edição dos dados que são atualizados no banco de dados.
- **Requisito Funcional Relacionado:** RF18.

18. Cadastro de Quartos:

- **Descrição:** Como proprietária, quero cadastrar todos os quartos da pousada no sistema, para fazer o controle de disponibilidade e da limpeza dos quartos.
- **CrITÉrios de Aceitação:**
 - O sistema possui um módulo de gestão de quartos com diferentes funcionalidades.
 - A proprietária acessa a funcionalidade “Cadastrar Quartos” e preenche um formulário de cadastro.
 - O formulário possui campos para adicionar: número do quarto, tipo, preço, status inicial, capacidade e observações.
 - O sistema faz a validação dos campos obrigatórios e evita duplicidade de número de quartos.
 - O quarto é registrado e adicionado ao banco de dados.
 - O quarto fica visível na lista de quartos após o registro.
- **Requisito Funcional Relacionado:** RF19.

19. Edição de Cadastro de Quarto:

- **Descrição:** Como proprietária, quero editar o cadastro de um quarto, para caso eu precise adicionar ou remover informações importantes para a reserva.
- **CrITÉrios de Aceitação:**

- A proprietária deve conseguir localizar o quarto pela lista disponível no módulo de gestão de quartos.
- O sistema permite a seleção do quarto.
- O sistema permite que a proprietária edite os dados dos campos disponíveis no formulário.
- As alterações são registradas e atualizadas no banco de dados.

- **Requisito Funcional Relacionado:** RF20.

20. Visualização Quartos Disponíveis:

- **Descrição:** Como proprietária, quero conseguir ver quais os quartos disponíveis durante o cadastro da reserva, para otimizar o atendimento.
- **CrITÉRIOS de Aceitação:**
 - Durante o cadastro de uma nova reserva, o sistema exibe uma lista somente com os quartos disponíveis para o período selecionado (*check-in* e *check-out*).
 - O sistema atualiza automaticamente a lista quando a proprietária altera as datas da reserva.
 - O sistema exibe os principais dados do quarto na seleção: número, tipo, capacidade e preço.
- **Requisito Funcional Relacionado:** RF21.

21. Atualização do Status do Quarto:

- **Descrição:** Como proprietária, quero atualizar manualmente o *status* de um quarto (disponível, reservado, ocupado, em manutenção) para refletir sua condição real e evitar reservas indevidas.
- **CrITÉRIOS de Aceitação:**
 - O sistema permite que a proprietária selecione um quarto e altere manualmente seu *status* atual, no módulo de gestão de quartos.
 - O sistema deve impedir a alteração do *status* caso o quarto esteja associado a alguma reserva, caso o *status* seja diferente de “em manutenção”.
 - A mudança de *status* é atualizada automaticamente.
 - O sistema exibe uma mensagem de confirmação de mudança.
- **Requisito Funcional Relacionado:** RF22.

22. Produção de Relatórios de Ocupação:

- **Descrição:** Como proprietária, quero gerar relatórios de ocupação de quartos por período (taxa de ocupação, quartos vagos) para analisar o desempenho da pousada.
- **Critérios de Aceitação:**
 - O sistema tem uma funcionalidade de geração de relatórios de ocupação no módulo de gestão de quartos.
 - A proprietária consegue selecionar um intervalo de datas para gerar o relatório.
 - O sistema gera um relatório que exibe a taxa de ocupação do período selecionado (% de dias ocupados por quarto).
 - O relatório também exibe a quantidade de quartos disponíveis e ocupados em cada dia do período.
- **Requisito Funcional Relacionado:** RF23.

23. Configuração de Horários de *Check-in* e *Check-out*:

- **Descrição:** Como proprietária, quero configurar os horários de *check-in* e *check-out* para melhor controle de pessoas e, consequentemente, melhor recepção dos hóspedes.
- **Critérios de Aceitação:**
 - O sistema oferece campos de configuração dos horários padrão de *check-in* e *check-out* por temporada.
 - O sistema oferece a opção de configurar horários por temporada (baixa/alta).
 - Após configurados, os horários devem ser armazenados no banco de dados.
 - A proprietária deve conseguir editar os horários configurados.
 - Os horários devem ser associados a disponibilidade para reserva das estadias.
- **Requisito Funcional Relacionado:** RF26.

24. Registro do *Check-out*:

- **Descrição:** Como proprietária, quero registrar o *check-out* de um hóspede para atualizar status do quarto e finalizar reserva.
- **Critérios de Aceitação:**
 - O sistema oferece a opção de registrar o *check-out* manualmente ao selecionar uma reserva.

- Após o *check-out*, o *status* do quarto deve mudar para "disponível" ou "reservado" automaticamente.
- Após o registro do *check-out*, o sistema deve encerrar a reserva e impedir alterações na reserva.
- Caso haja cobrança pendente, o sistema deve alertar antes de finalizar a reserva.

- **Requisito Funcional Relacionado:** RF24.

25. Mudança de Status Pós *Check-in*:

- **Descrição:** Como proprietária, quero que o sistema mude o status do quarto de reservado para ocupado ao registrar o *check-in*, para manter atualizada a ocupação real da pousada.
- **Critérios de Aceitação:**
 - Quando a proprietária registrar o *check-in* de uma reserva, o *status* do quarto deve ser alterado automaticamente para "ocupado".
 - O sistema só deve alterar o *status* do quarto se a reserva estiver confirmada e com o pagamento parcial ou total realizado.
 - O sistema deve exibir uma mensagem de pagamento pendente, caso o pagamento não tenha sido registrado em sua totalidade.
 - O sistema deve exibir a alteração de *status* imediatamente no painel da pousada.
- **Requisito Funcional Relacionado:** RF25.

26. *Early check-in* e *Late check-out*:

- **Descrição:** Como proprietária, quero registrar *early check-in* e *late check-out* em reservas para oferecer mais flexibilidade ao cliente.
- **Critérios de Aceitação:**
 - Durante o cadastro da reserva, o sistema deve permitir marcar uma reserva com *early check-in* e/ou *late check-out*.
 - O sistema deve validar se o quarto estará disponível nos horários solicitados.
 - O sistema deve adicionar as tarifas no valor da estadia.
 - Caso a opção seja marcada posteriormente a algum registro de pagamento, o sistema deve exibir um alerta de pagamento pendente referente ao valor das tarifas ou da soma da tarifa e dos demais valores pendentes.
 - A opção deve estar disponível tanto no cadastro da reserva quanto na edição posterior.

- O sistema deve registrar essa informação no histórico da reserva e no comprovante de pagamento.

- **Requisito Funcional Relacionado:** RF28.

27. Configuração de Tarifas Adicionais:

- **Descrição:** Como proprietária, quero configurar tarifas adicionais para *early check-in* e *late check-out* para facilitar a cobrança correta do cliente.
- **Critérios de Aceitação:**
 - O sistema deve permitir que a proprietária adicione valores fixos ou percentuais adicionais para *early check-in* e *late check-out* aos preços das estadias.
 - O valor total da reserva deve refletir os acréscimos automaticamente.
- **Requisito Funcional Relacionado:** RF29.

28. Registro de Fluxos Financeiros:

- **Descrição:** Como proprietária, quero registrar receitas e despesas da pousada com data, valor, categoria e descrição para manter o controle financeiro.
- **Critérios de Aceitação:**
 - O sistema permite que a proprietária registre receitas ou despesas dentro do módulo de gestão financeira.
 - A proprietária consegue registrar receitas ou despesas a partir do preenchimento de um formulário com os campos obrigatórios: data, categoria, valor e descrição.
 - O sistema armazena no banco de dados a receita/ despesa registrada e salva.
- **Requisito Funcional Relacionado:** RF30 e RF31.

29. Edição e Exclusão dos Registros de Fluxo Financeiro:

- **Descrição:** Como proprietária, quero editar ou excluir uma receita ou despesa registrada para corrigir possíveis erros ou lançamentos equivocados.
- **Critérios de Aceitação:**
 - O sistema permite que a proprietária selecione uma receita/despesa no módulo de gestão financeira e escolha se quer alterar ou excluir o registro.
 - O sistema permite que a proprietária exclua definitivamente um registro após a confirmação do alerta de exclusão.

- A proprietária consegue alterar os campos obrigatórios do formulário de registro de receita/despesa.
- O sistema atualiza o banco de dados, o saldo e o balanço financeiro automaticamente.

- **Requisito Funcional Relacionado:** RF32 e RF33.

30. Visualização das Transações Financeiras:

- **Descrição:** Como proprietária, quero visualizar todas as transações financeiras por período para entender a saúde financeira da pousada.
- **Critérios de Aceitação:**
 - O sistema exibe uma lista de todas as transações por data no módulo de gestão financeira.
 - A proprietária consegue fazer buscas por filtradas: tipo, valor, data e categoria.
 - A filtragem exibe apenas as transações que atendem aos critérios informados.
- **Requisito Funcional Relacionado:** RF34 e RF35.

31. Produção de Relatórios Financeiros:

- **Descrição:** Como proprietária, quero gerar relatórios de financeiros automaticamente por período e categoria para me auxiliar nas decisões financeiras da pousada.
- **Critérios de Aceitação:**
 - No módulo de gestão financeira, o sistema deve oferecer a funcionalidade de produção de relatórios financeiros.
 - O sistema deve oferecer filtros(intervalo de datas e categorias) para serem aplicados na produção.
 - O relatório produzido exibe o total de receitas, total de despesas e saldo final do período selecionado.
 - O relatório deve apresentar gráficos simples para representar os resultados por categoria.
 - O relatório desconsidera registros financeiros excluídos.
- **Requisito Funcional Relacionado:** RF36 e RF38.

32. Balanço Financeiro:

- **Descrição:** Como proprietária, quero gerar um balanço financeiro simples com receitas, despesas e saldo para avaliar lucros e perdas.

- **Critérios de Aceitação:**

- O sistema permite que a proprietária visualize o total de receitas, o total de despesas e o saldo, no painel do módulo de gestão financeira.
- O sistema gera balanços com base nos filtros aplicados.

- **Requisito Funcional Relacionado:** RF37.

33. Painel Inicial:

- **Descrição:** Como proprietária, quero visualizar um *dashboard* inicial com métricas chaves (quartos ocupados, próximas reservas, próximos *check-out*) para ter uma visão rápida e geral da pousada.

- **Critérios de Aceitação:**

- O sistema deve exibir no painel inicial o número de quartos ocupados no dia atual.
- O sistema deve exibir no painel inicial a lista de reservas com *check-in* previstos para as próximas 24 horas.
- O sistema deve exibir no painel inicial a lista de reservas com *check-out* previstos para as próximas 24 horas.
- O sistema atualiza as informações do painel automaticamente a cada acesso.
- A proprietária consegue acessar o painel diretamente após o *login* no sistema.

- **Requisito Funcional Relacionado:** RF39.

34. Notificações internas:

- **Descrição:** Como proprietária, quero receber notificações automáticas (via *dashboard*) sobre eventos importantes para me preparar e melhorar a experiência dos hóspedes.

- **Critérios de Aceitação:**

- O sistema exibe uma mensagem para a proprietária no dashboard sempre que uma reserva é criada, cancelada ou editada.
- O sistema exibe lembretes de pendências de pagamento com base na data de vencimento.

- **Requisito Funcional Relacionado:** RF40.

4.3 Arquitetura

4.3.1 Desenho da Arquitetura

A arquitetura do sistema de reservas da Pousada Chalés Água de Coco foi criada com base no framework Django ([DJANGO, 2024](#)), utilizando o padrão de arquitetura MTV (Model-Template-View). Esse padrão permite a separação lógica das responsabilidades entre modelo de dados, interface de apresentação e regras de controle, garantindo organização, reutilização de código e manutenibilidade da aplicação.

4.3.1.1 Estrutura da Solução

A camada de interface (Template) é responsável pela exibição das páginas HTML aos usuários. Ela recebe os dados dinamicamente da camada de controle (View), que, por sua vez, realiza o processamento necessário, interage com o banco de dados e fornece as informações pertinentes à exibição. A camada de modelo (Model) representa a estrutura do banco de dados relacional, utilizando o ORM (Object-Relational Mapper) nativo do Django para abstrair a manipulação dos dados de forma segura e eficiente. A aplicação foi projetada como uma solução fullstack, sendo o Django responsável tanto pelo front-end quanto pelo back-end. No front-end, a interface é construída com HTML e CSS, que promovem responsividade e usabilidade ([DUCKETT, 2011](#)). No back-end, são implementadas as regras de negócio, controle de autenticação, segurança da aplicação e comunicação com o banco de dados PostgreSQL .

4.3.1.2 Banco de Dados e Infraestrutura

O banco de dados utilizado é o PostgreSQL, devido à sua robustez, suporte a dados relacionais e facilidade de integração com o Django ([POSTGRESQL, 2025](#)). Todas as tabelas, relacionamentos e consultas são gerenciados pelo ORM, o que minimiza riscos de injeção de código malicioso e facilita a manutenção. A infraestrutura de hospedagem está baseada na Amazon Web Services (AWS), garantindo alta disponibilidade, escalabilidade e segurança. O servidor é configurado para suportar conexões seguras via HTTPS, e mecanismos de backup e monitoramento são empregados para garantir a continuidade do serviço ([AWS, 2025](#)). Por fim, a arquitetura proposta está preparada para evoluções futuras, permitindo a integração com APIs externas e relatórios gerenciais, com base em uma estrutura modular e escalável.

4.3.2 Diagrama da Arquitetura

4.3.2.1 Diagrama de Implantação

A Figura 1 mostra o funcionamento da arquitetura do sistema.

Diagrama de Implantação

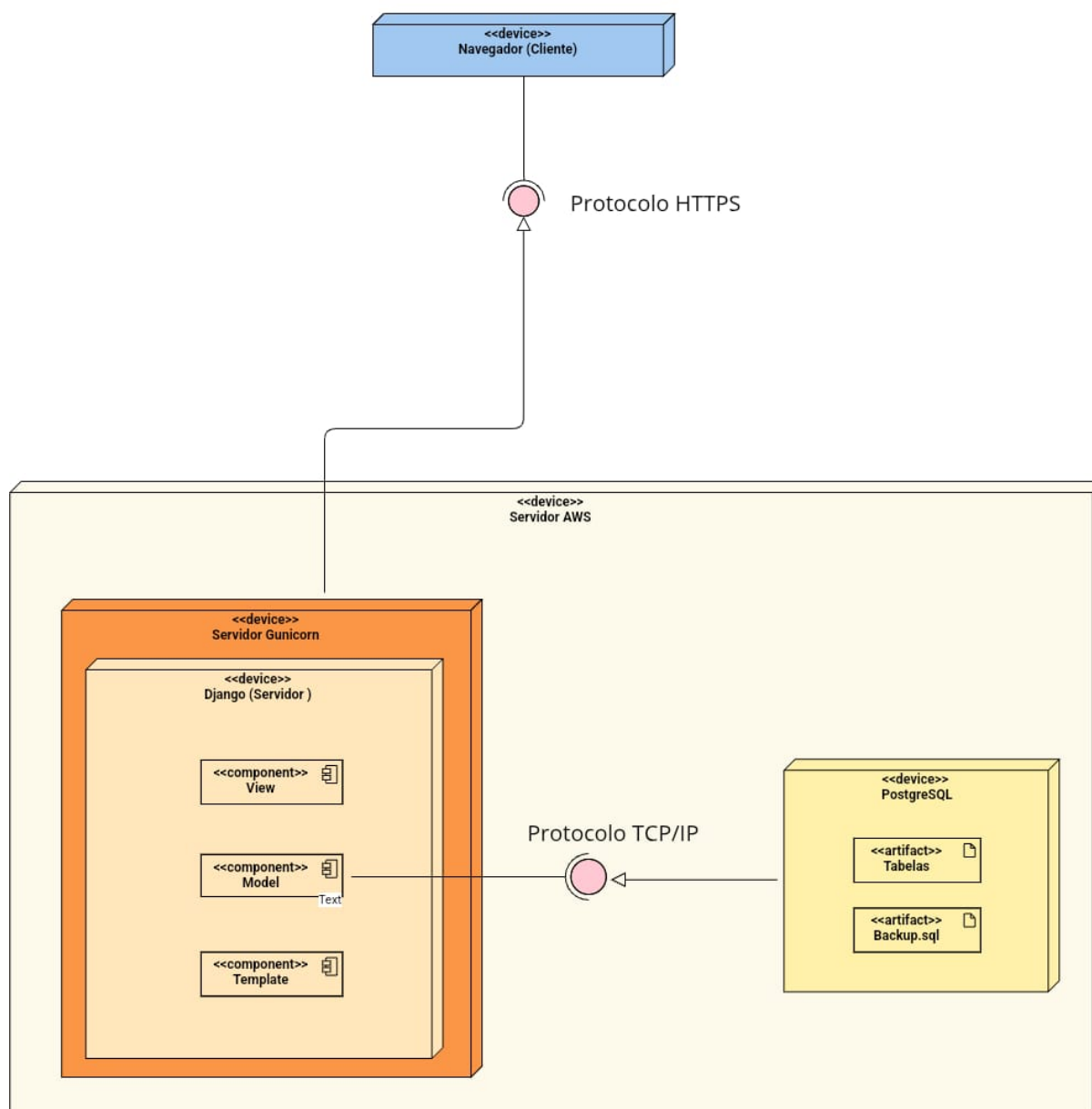


Figura 1 – Diagrama de Implantação desenvolvido no Online Visual-Paradigm

4.3.2.2 Diagrama de Componentes

A Figura 2 mostra o funcionamento da arquitetura do sistema.

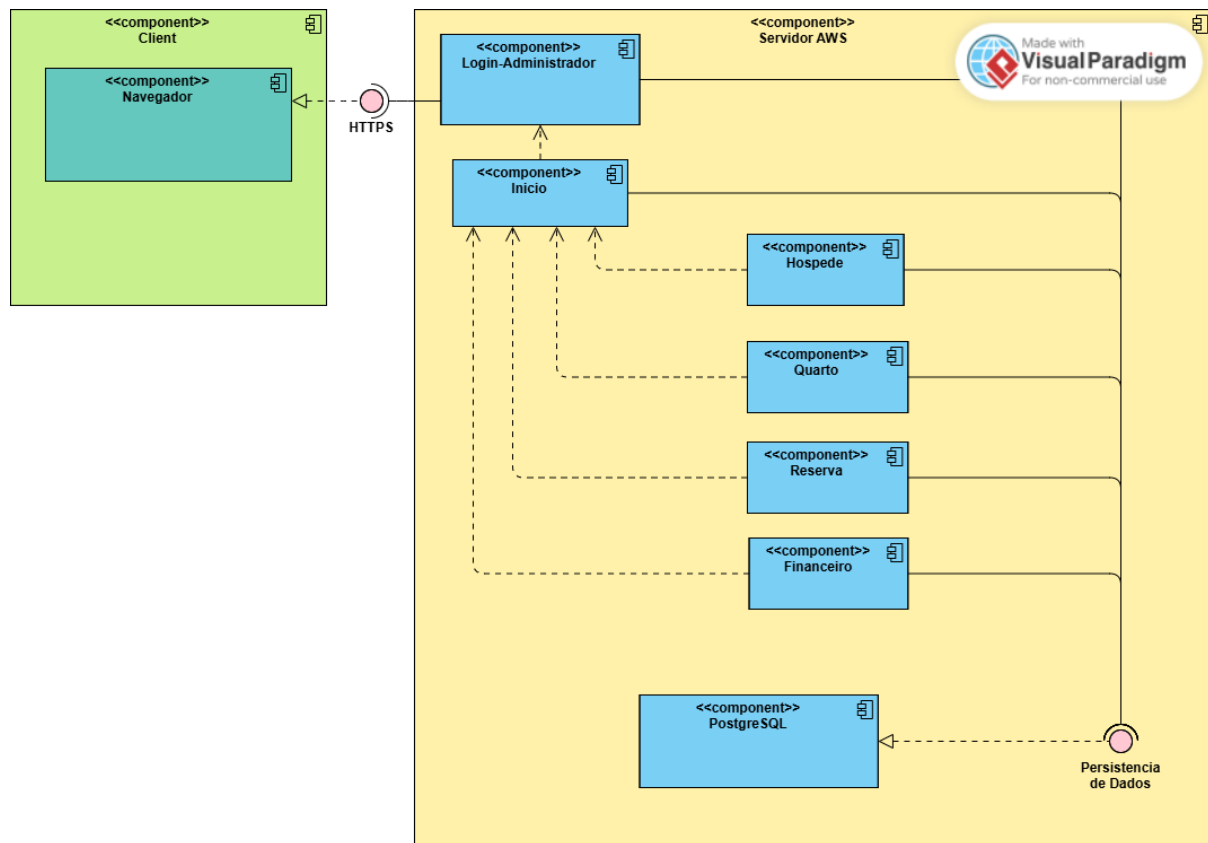


Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm

4.4 Tecnologias

4.4.1 Django

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 (PYTHON, 2025). Uma de suas principais vantagens é a rapidez no desenvolvimento, já que ele vem com diversas funcionalidades já prontas (DJANGO, 2024).

4.4.1.1 Front-end

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.

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.

4.4.1.2 Back-end

O desenvolvimento do back-end desta aplicação Django é baseada no padrão MTV, 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.

4.4.1.3 Banco de Dados

O Banco de Dados é 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 ([POSTGRESQL, 2025](#)).

4.4.2 Justificativa da Escolha

A escolha do framework Django para o desenvolvimento do sistema da Pousada Chalés Água de Coco baseou-se em critérios técnicos, de segurança, escalabilidade e aderência às boas práticas de desenvolvimento web moderno. Django é um framework escrito em Python, que segue o padrão MTV, semelhante ao clássico MVC(Model-View-Controller), promovendo uma clara separação entre as camadas da aplicação ([PYTHON, 2025](#)).

4.4.2.1 Justificativa Técnica

Entre os diferenciais do Django, destacam-se o ORM nativo, que abstrai o uso de SQL e facilita a manipulação segura dos dados; o sistema integrado de autenticação e autorização, que oferece controle de acesso granular; e a proteção nativa contra ataques como SQL Injection, Cross-Site Scripting (XSS) e Cross-Site Request Forgery (CSRF). Esses recursos reduzem significativamente o tempo de desenvolvimento e aumentam a segurança da aplicação. Além disso, o Django é um software de código aberto, com forte comunidade ativa, documentação completa e contínua evolução. Essa característica o torna ideal para projetos acadêmicos e corporativos, permitindo a entrega de soluções confiáveis e bem estruturadas ([DJANGO, 2024](#)).

4.4.2.2 Infraestrutura com AWS

A escolha pela Amazon Web Services (AWS) como provedora da infraestrutura em nuvem está relacionada à sua capacidade de oferecer escalabilidade, disponibilidade e segurança ([AWS, 2025](#)). Os recursos de computação elástica, gerenciamento de banco de dados, balanceamento de carga, armazenamento e backup são fundamentais para a operação de um sistema que lida com informações sensíveis de clientes e reservas. A integração entre Django e AWS ocorre de forma transparente, possibilitando o uso de serviços como S3 (armazenamento de mídia), RDS (gerenciamento de banco de dados relacional) e CloudWatch (monitoramento), ampliando o potencial da aplicação e assegurando a continuidade do serviço com mínimo risco de falhas.

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 ([GITHUB, 2024](#)).

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 ([BRMODELO, 2024](#)). 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 ([VISUAL-PARADIGM, 2024](#)). 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 ([LATEX, 2024](#)).

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 ([GOOGLE, 2024](#)). 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 criado 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 ([GITHUB, 2024](#)). 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.

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, a proteção de dados e o cumprimento das normas legais foram diretrizes centrais na concepção do sistema de reservas da Pousada Chalés Água de Coco. Para tanto, foram adotadas boas práticas de desenvolvimento seguro, recursos nativos do framework Django e medidas alinhadas à Lei Geral de Proteção de Dados (LGPD – Lei nº 13.709/2018).

4.7.1 Segurança da Aplicação

O framework Django oferece proteção automática contra ameaças comuns da web, como injeção de SQL, execução remota de código, Cross-Site Scripting (XSS) e falsificação de requisições entre sites (CSRF) (DJANGO, 2024). Além disso, foram implementadas as seguintes práticas:

Sistema de autenticação e autorização: Controle de acesso baseado em permissões, exigência de credenciais válidas e proteção de páginas sensíveis por autenticação obrigatória.

Proteção contra CSRF: Utilização de tokens em requisições POST, garantindo que apenas usuários legítimos possam executar ações críticas.

Escapamento automático de HTML (XSS): O mecanismo de templates evita a execução de scripts maliciosos. Armazenamento seguro de senhas: Utilização de algoritmos modernos de hash, como PBKDF2, inviabilizando a recuperação das senhas mesmo em caso de vazamento.

Gerenciamento seguro de sessões: Identificadores criptografados, proteção contra falsificação e expiração automática de sessões inativas.

4.7.2 Segurança na Comunicação

Todas as comunicações entre cliente e servidor são realizadas via HTTPS, com uso de certificado SSL/TLS. O certificado foi emitido por uma autoridade certificadora confiável, após validação de domínio e instalação no servidor da AWS (AWS, 2025). A criptografia das conexões assegura a confidencialidade, integridade e autenticidade dos dados transmitidos. A aplicação redireciona automaticamente requisições HTTP para HTTPS e adiciona cabeçalhos HTTP de segurança, prevenindo de possíveis ataques.

4.7.2.1 Conformidade com a LGPD

Para atender aos princípios da LGPD, o sistema foi projetado com foco na coleta mínima de dados, na transparência quanto ao uso das informações e no controle por parte do titular dos dados. São assegurados os seguintes direitos:

Solicitação de remoção de dados pessoais;

Alteração de consentimento previamente fornecido;

Acesso claro às finalidades do tratamento dos dados.

Além disso, os dados são armazenados de forma segura e submetidos a backups regulares em ambiente de nuvem, garantindo resiliência e conformidade com os princípios de segurança, prevenção e responsabilização da LGPD (BRASIL, 2018).

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.

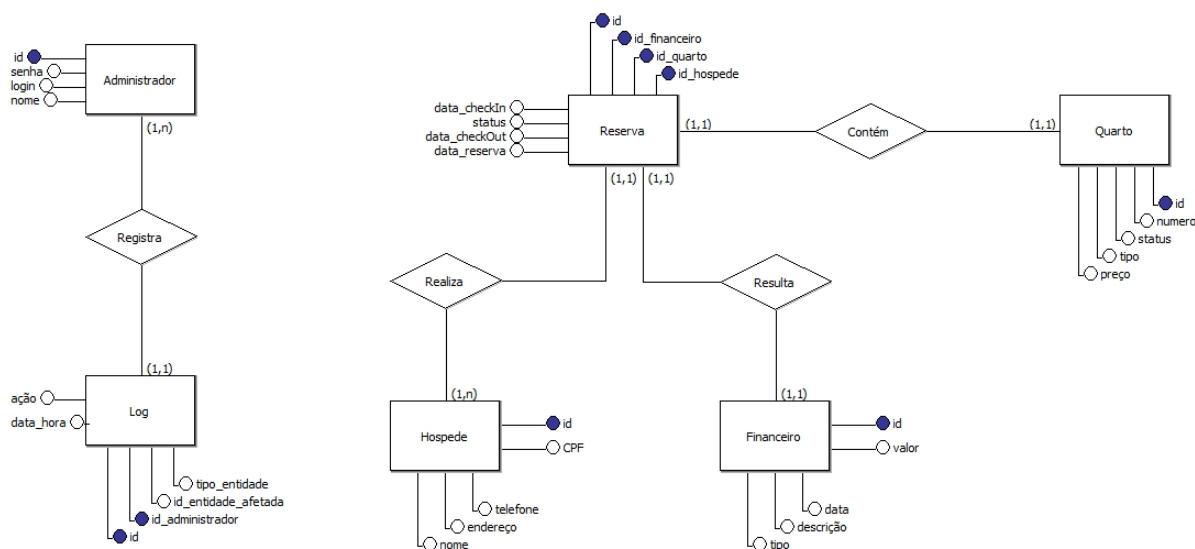


Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo

4.8.2 Diagrama Entidade-Relacionamento - DER

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

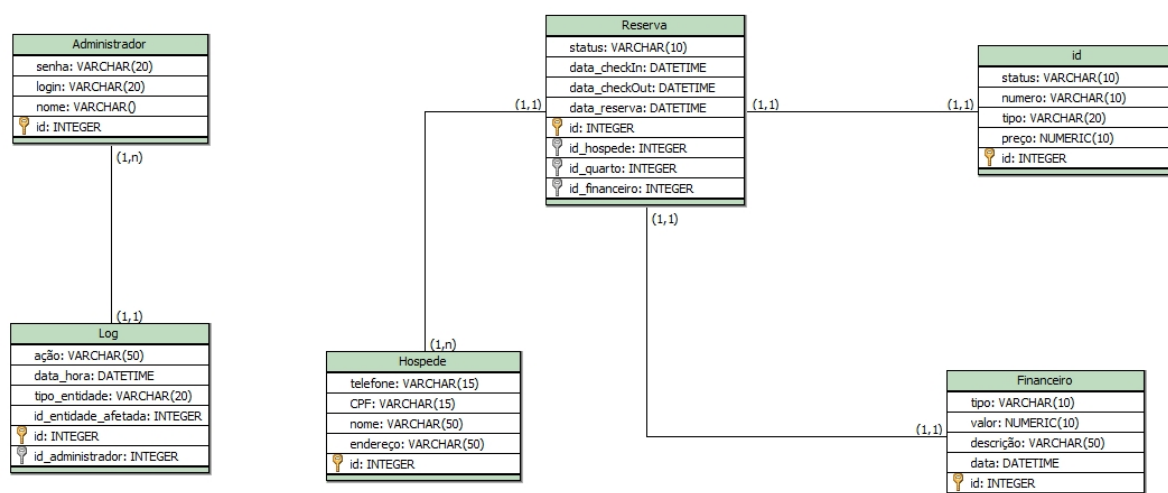


Figura 4 – Diagrama Entidade-Relacionamento (DER) desenvolvido no brModelo

4.9 Entregas

4.9.1 Desenvolvimento de um Tema - 08/04

Nesta etapa inicial, foi definido o tema central do projeto: o desenvolvimento de um sistema web para automatizar os processos administrativos da pousada Chalés Água de Coco. A escolha foi baseada em uma demanda real identificada junto ao empreendimento, permitindo alinhar os objetivos acadêmicos com a solução de um problema concreto.

4.9.2 Desenho da Aplicação - 29/04

Foi elaborada a arquitetura do sistema utilizando os diagramas UML de Componentes e de Implantação, utilizando o padrão MTV (Model-Template-View) do framework Django. Nessa fase, também foi realizada a modelagem inicial do banco de dados relacional utilizando o PostgreSQL.

4.9.3 Prova de Conceito - 20/05

Na etapa da prova de conceito, foi desenvolvido um sistema funcional de cadastro de hóspedes, com a aplicação já hospedada na infraestrutura da Amazon Web Services (AWS). Nesta versão inicial, foi implementado um CRUD completo (Create, Read, Update, Delete) utilizando o banco de dados relacional PostgreSQL, gerenciado por meio do ORM nativo do Django. Essa entrega permitiu validar a viabilidade técnica da solução, testar a integração entre as camadas da aplicação e comprovar o funcionamento do ambiente de produção na nuvem.

4.9.4 Análise e Documentação - 10/06

Nesta etapa, foi realizada a entrega da documentação referente ao Projeto de Conclusão de Curso (PCC), contendo toda a análise do problema, justificativas, objetivos, levantamento de requisitos, arquitetura do sistema, escolha das tecnologias e desenvolvimento. A documentação formaliza todas as etapas do projeto, desde sua concepção até a implementação da solução proposta, servindo como registro técnico e acadêmico do trabalho desenvolvido.

4.9.5 Produto Mínimo Viável - 24/06

O Produto Mínimo Viável (MVP) entregue contempla as funcionalidades essenciais do sistema, focando na gestão de hóspedes, acomodações e reservas. Essas funcionalidades já permitem à pousada Chalés Água de Coco substituir o controle manual por um sistema digital centralizado e acessível. A parte de controle financeiro, embora planejada, será desenvolvida em uma fase posterior, prevista para o próximo semestre.

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

A parte de custos por estrutura está divida entre Equipamentos e Serviços, que diz respeito aos equipamentos e serviços necessários para que cada colaborador do time consiga desempenhar suas funções. E Infraestrutura, que aborda os custos relativos a sustentação do sistema.

Considerando que a equipe trabalha no regime home office e utiliza seus próprios equipamentos, entendemos os gastos como nulos. Partindo assim para tarifas referentes a energia e internet ([ANATEL, 2025](#)). Já nos custos indicados na parte de infraestrutura foram considerados nulos pois o sistema será desenvolvido utilizando serviços gratuito (lembrando que essa análise se trata de uma estimativa, e que caso haja alguma mudança significativa no projeto ou necessidade de serviços pagos, esses custos podem mudar) ([ENEL, 2025](#)).

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

Na imagem a seguir temos uma tabela detalhada com os custos referentes a mão de obra do desenvolvimento do projeto. Na primeira coluna temos a especificação dos papéis de cada profissional, em seguida quantidade de pessoas por cargo, horas trabalhadas ao dia e dias trabalhados ao mês(média). Também podemos checar o preço por hora de cada profissional, assim finalizamos com as estimativas de custo mensal, total 4 meses (correspondente ao tempo em que a equipe já trabalhou no projeto até então) e projeção 9 meses (correspondente ao tempo total estimado de projeto).

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	132
Engenheiro de Dados (DBA)	1	6	22	132
Analista de Documentação	1	6	22	132
Gerente de Projeto (PM)	1	6	22	132
Desenvolvedor Front-End	1	6	22	132
Desenvolvedor Back-End	1	6	22	132
Total Mão de Obra	6	36	132	792

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	2.112,00	8.448,00	19.008,00
Engenheiro de Dados (DBA)	13,00	1.716,00	6.864,00	15.444,00
Analista de Documentação	15,00	1.980,00	7.920,00	17.820,00
Gerente de Projeto (PM)	16,00	2.112,00	8.448,00	19.008,00
Desenvolvedor Front-End	13,00	1.716,00	6.864,00	15.444,00
Desenvolvedor Back-End	13,00	1.716,00	6.864,00	15.444,00
Total Mão de Obra	86,00	11.352,00	45.408,00	102.168,00

Fonte: Elaborado pelos autores.

5.1.3 Custo Total

Na imagem a seguir temos a junção dos custos por mão de obra e custos por estrutura, sendo visualizadas em 3 períodos diferentes: Custo Mensal - Referente a gastos de um único mês. Custo Total (4 meses) - Custos referentes aos meses que foram trabalhados até a data de escrita deste texto. Custo Total (9 meses) - Custos referentes ao total de meses programado para a realização do projeto.

Tabela 6 – Custo Total por Categoria

Categoria	Custo Mensal (R\$)	Custo Total (4 meses) (R\$)	Custo Total (9 meses) (R\$)
Mão de Obra	11.352,00	45.408,00	102.168,00
Estrutura	710,46	2.841,84	6.394,14
Total	12.062,46	48.249,84	108.562,14

Fonte: Elaborado pelos autores.

5.2 Cenários

5.2.1 Cenário Otimista

Na imagem a seguir podemos observar a projeção construída pela equipe de um cenário otimista, levando em consideração os custos já apresentados anteriormente de mão de obra e estrutura. Podemos observar que de acordo com essa projeção, o ponto de equilíbrio será alcançado por volta de 6 meses.

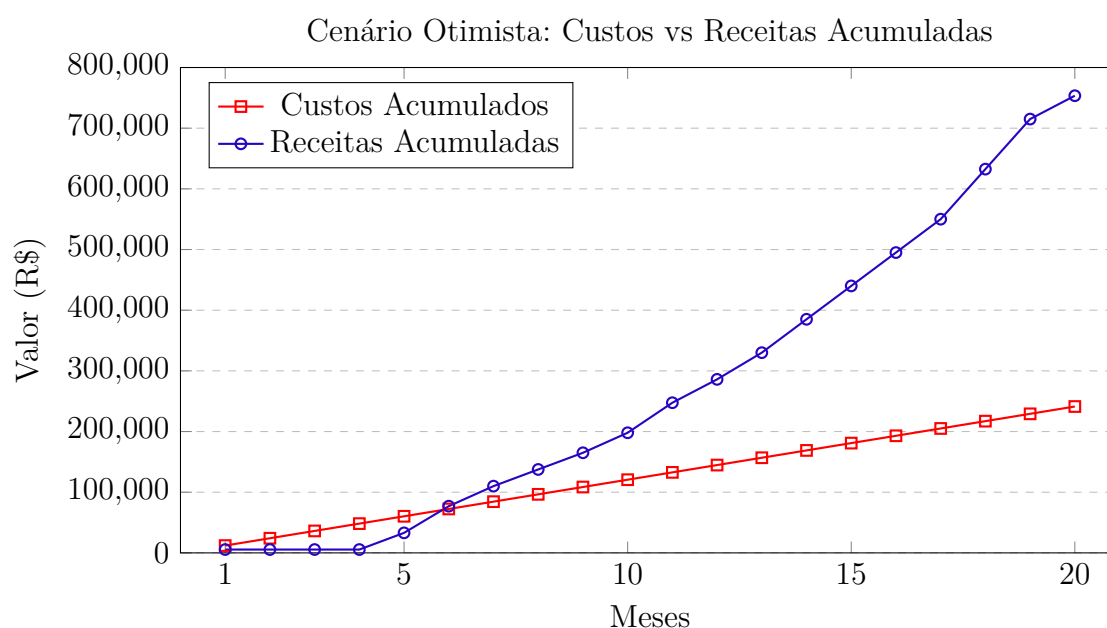


Figura 5 – Comparação dos custos e receitas acumuladas no cenário otimista

5.2.2 Cenário Pessimista

Na imagem a seguir podemos observar a projeção construída pela equipe de um cenário pessimista, levando em consideração os custos já apresentados anteriormente de mão de obra e estrutura. Podemos observar que de acordo com essa projeção, o ponto de equilíbrio será alcançado por volta de 20 meses.

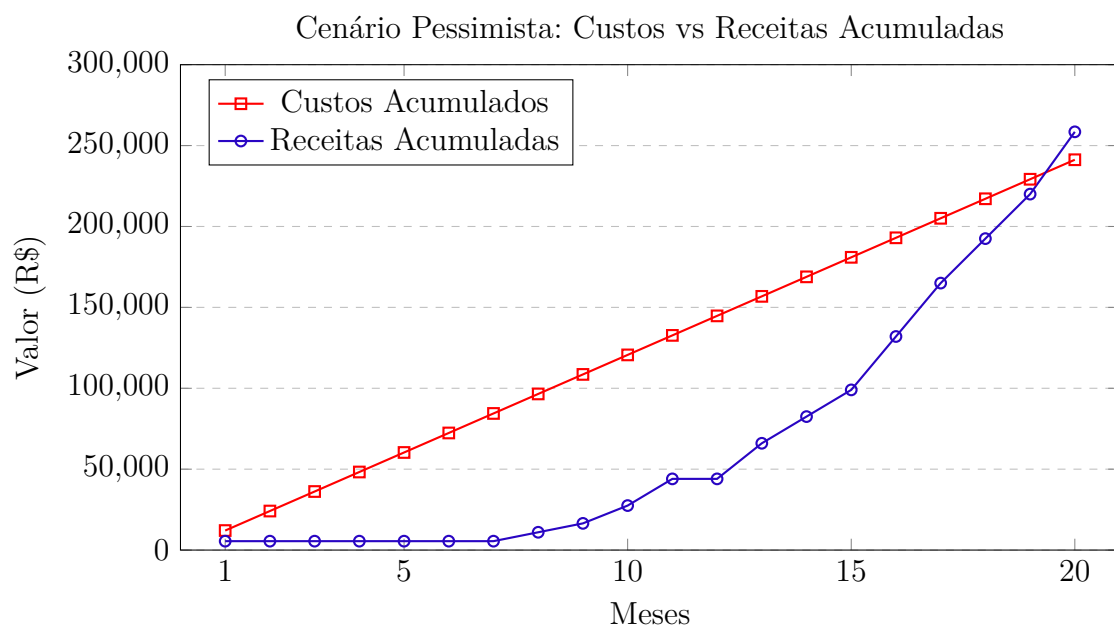


Figura 6 – Comparação dos custos e receitas acumuladas no cenário pessimista

5.2.3 Cenário Realista

Na imagem a seguir podemos observar a projeção construída pela equipe de um cenário realista, levando em consideração os custos já apresentados anteriormente de mão de obra e estrutura. Podemos observar que de acordo com essa projeção, o ponto de equilíbrio será alcançado por volta de 15 meses.

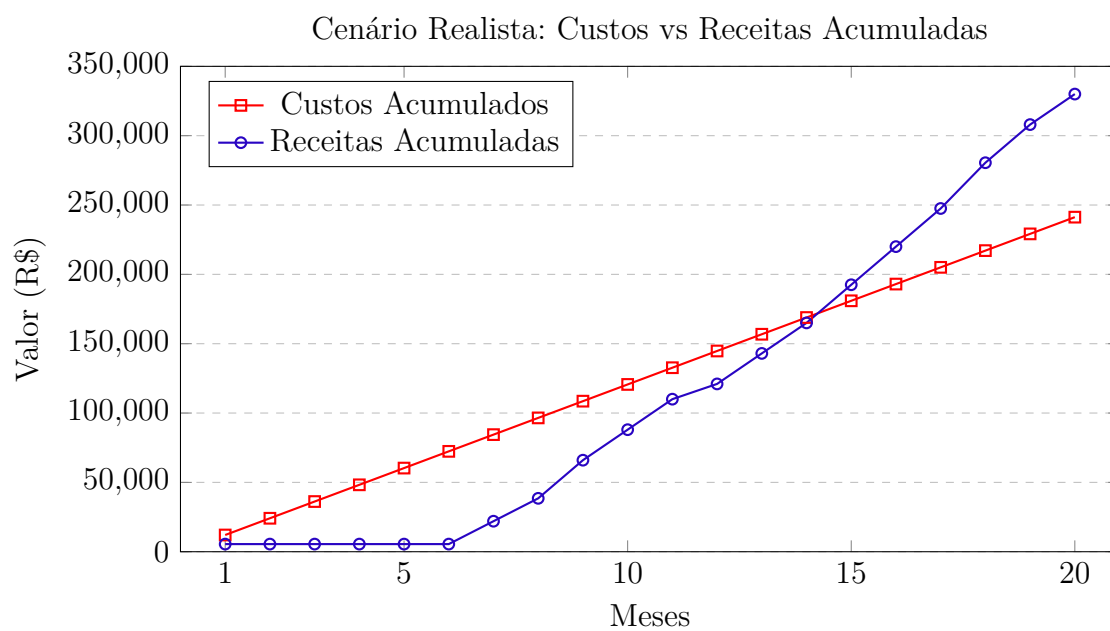


Figura 7 – Comparação dos custos e receitas acumuladas no cenário realista

6 Considerações Finais

O desenvolvimento do sistema web para a pousada Chalés Água de Coco representa uma resposta prática e eficiente à necessidade de modernização enfrentada por pequenos empreendimentos do setor de hospitalidade. Ao longo do projeto, foi possível identificar fragilidades nos métodos tradicionais utilizados para gestão de hóspedes, reservas e finanças — especialmente aqueles baseados em planilhas eletrônicas — que, embora populares, oferecem baixa escalabilidade, alto risco de erro e pouca integração entre processos.

Através de uma parceria direta com os responsáveis pela pousada, foi possível realizar um levantamento detalhado dos requisitos do sistema, o que permitiu construir uma solução personalizada, centrada nas reais necessidades operacionais do negócio. O uso do framework Django, aliado à infraestrutura da Amazon Web Services (AWS), proporcionou uma arquitetura robusta, segura e escalável, capaz de sustentar a aplicação tanto em seu estágio inicial quanto em futuras evoluções.

Além da automatização das principais funções administrativas da pousada, como o controle de reservas e a organização dos dados financeiros, o sistema promoveu melhorias significativas na usabilidade, no acesso remoto às informações e na geração de relatórios gerenciais. Com isso, o projeto atendeu plenamente aos seus objetivos, oferecendo uma ferramenta funcional, acessível via internet e com grande potencial de impacto na rotina de trabalho da pousada.

O projeto também demonstrou, na prática, a aplicabilidade dos conhecimentos adquiridos ao longo do curso, abrangendo aspectos de análise de requisitos, modelagem de dados, desenvolvimento back-end e front-end, segurança da informação e implantação em ambiente de nuvem. Trata-se, portanto, de um produto tecnológico que, além de resolver uma demanda real, reforça a importância da tecnologia na transformação digital de pequenos negócios.

7 Conclusão

Este Projeto de Conclusão de Curso teve como propósito desenvolver um sistema web para automatizar os processos administrativos da pousada Chalés Água de Coco, promovendo uma solução moderna e eficiente em substituição ao modelo tradicional baseado em planilhas. O sistema entregue oferece recursos essenciais para a gestão de hóspedes, reservas, acomodações e controle financeiro, consolidando-se como uma plataforma completa e adaptada à realidade da pousada.

A aplicação da arquitetura MTV com o framework Django, a adoção de práticas seguras de desenvolvimento e o uso da infraestrutura em nuvem da AWS contribuíram para a construção de um sistema robusto e escalável, capaz de oferecer alto desempenho e disponibilidade. A utilização de tecnologias amplamente reconhecidas no mercado assegura não apenas a qualidade técnica do sistema, mas também sua viabilidade para expansão futura.

Como resultado, a pousada passa a contar com uma ferramenta que facilita a tomada de decisões, minimiza falhas operacionais e melhora a organização das informações. O projeto também evidencia como soluções de baixo custo e alto impacto podem ser desenvolvidas e aplicadas em pequenos negócios, promovendo inovação e melhoria contínua.

Referências

- ANATEL. *Documento Técnico Anatel*. 2025. Acessado em 7 de junho de 2025. Disponível em: <<https://sistemas.anatel.gov.br/anexar-api/publico/anexos/download/9c9e947419913b3987bd8cc7510f03f1>>. Citado na página 73.
- AWS. *AWS - Amazon Web Services*. 2025. Disponível em: <<https://aws.amazon.com/documentation/>>. Acesso em: 7 jun. 2025. Citado 3 vezes nas páginas 63, 67 e 69.
- BRASIL. *Lei Geral de Proteção de Dados Pessoais (LGPD) – Lei nº 13.709*. 2018. Diário Oficial da União, Brasília, DF, 15 ago. 2018. Citado na página 70.
- BRMODELO. *BRModelo Web*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://www.brmodeloweb.com/>>. Citado na página 67.
- COCO, P. C. Água de. *Pousada Chalés Água de Coco*. 2025. Acesso em: 10 de abril de 2025. Disponível em: <<https://pousadachalesaguadecoco.com.br/>>. Citado 2 vezes nas páginas 41 e 43.
- DJANGO. *Django Documentation*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://docs.djangoproject.com/pt-br/4.2/>>. Citado 4 vezes nas páginas 63, 65, 66 e 69.
- DUCKETT. *HTML and CSS: Design and Build Websites*. [S.l.]: Wiley, 2011. ISBN 9781118008188. Citado na página 63.
- ENEL. *Tarifa de Energia Elétrica*. 2025. Acessado em 7 de junho de 2025. Disponível em: <https://www.enel.com.br/pt-saopaulo/Para_Voce/tarifa-energia-eletrica.html>. Citado na página 73.
- GITHUB. *GitHub Docs*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://docs.github.com/>>. Citado 2 vezes nas páginas 67 e 68.
- GOOGLE. *Google Meet Help*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://support.google.com/meet/>>. Citado na página 68.
- IBM. *Business Rules*. 2025. Acesso em 4 de junho de 2025. Disponível em: <<https://www.ibm.com/br-pt/topics/business-rules>>. Citado na página 41.
- LATEX. *LaTeX - A document preparation system*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://www.latex-project.org/>>. Citado na página 68.
- POSTGRESQL. *The PostgreSQL Documentation*. [S.l.]: PostgreSQL, 2025. Disponível em: <<https://www.postgresql.org/docs/>>. Acesso em: 7 jun. 2025. Citado 2 vezes nas páginas 63 e 66.
- PYTHON. *Python Official Documentation*. 2025. Disponível em: <<https://docs.python.org/3/>>. Acesso em: 7 jun. 2025. Citado 2 vezes nas páginas 65 e 66.

SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide – Scrum Guide em Português*. [S.l.], 2013. Desenvolvido e mantido por Ken Schwaber e Jeff Sutherland. Disponível em: <<https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Citado 4 vezes nas páginas 21, 22, 23 e 27.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: Pearson, 2011. E-book. Disponível em: <<https://plataforma.bvirtual.com.br>>. Acesso em: 04 jun. 2025. Citado 3 vezes nas páginas 31, 32 e 43.

SOUSA, A. J. L. D. et al. *Sistema Web Gestão Pousada*. 2025. <https://github.com/guilhermebschmidt/SistemaWebGestaoPousada.git>. Acesso em: 9 jun. 2025. Citado na página 38.

VISUAL-PARADIGM. *Visual Paradigm Online Documentation*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://online.visual-paradigm.com/>>. Citado na página 67.

Apêndices

APÊNDICE A – Diário de Bordo

A.1 1º SEMANA

Período: 25/03/2025 a 01/04/2025.

Nessa semana, com os membros da equipe definidos, iniciamos nossas atividades para o desenvolvimento do projeto da disciplina de Projeto de Extensão Integrado I. Assim, essa semana criamos um grupo na plataforma Whatsapp para estabelecermos nossa comunicação e facilitar o levantamento de possíveis temas para discutirmos em sala. Assim, no dia 01/04/2025 analisamos os parceiros disponíveis e a suas principais necessidades e optamos por desenvolver uma aplicação web para a empresa Pousada Chalés Água de Coco.

A.2 2º SEMANA

Período: 01/04/2025 a 08/04/2025.

Durante essa semana aprofundamos nosso conhecimento sobre a empresa parceira escolhida, entendendo sua principal necessidade: automatização dos serviços de gestão. Definimos que iremos desenvolver uma aplicação web de gestão para a proprietária com o objetivo de facilitar e otimizar a administração do negócio. A partir disso, também definimos nosso MVP (Produto Mínimo Viável), levando em consideração nossa capacidade técnica e os serviços da pousada. Preenchemos a primeira planilha de avaliação do grupo e auto-avaliação. Por fim, nos preparamos para apresentar a nossa escolha de tema para o professor orientador, realizada no dia 08/04, na qual também apresentamos nosso MVP. Funcionalidade MVP: Gestão de Reservas e Check-in/Check-out.

A.3 3º SEMANA

Período: 08/04/2025 a 15/04/2025.

Nesta semana começamos efetivamente as tarefas associadas ao desenvolvimento do nosso projeto. Foram elas: - Definimos a metodologia de gestão de projeto Scrum. Usamos como base para nossa escolha: experiências práticas/ conhecimento prévio da ferramenta, isto é, familiaridade. - Definimos a função de cada membro da equipe: Anna Julia: Analista de Cronograma/ PMO. Guilherme Akio: Engenheiro de Dados/ Administrador de Banco de Dados. Guilherme Bittencourt: Analista de Documentação/Arquiteto de Software.

Kelly Radchelle: Gerente de Projeto. Rafael Teixeira: Desenvolvedor Frontend. Ricardo Carriel: Desenvolvedor Backend. - Criamos nosso arquivo no ProjectLibre, adicionamos os principais marcos do projeto e os recursos humanos. - Criamos nossa documentação LateX e fizemos as primeiras alterações no arquivo. - Iniciamos o levantamento dos requisitos funcionais do MVP.

A.4 4° SEMANA

Período: 15/04/2025 a 22/04/2025.

Nesta semana foram realizadas as tarefas e discussões para iniciarmos o desenvolvimento do desenho da nossa aplicação: Analisamos os requisitos funcionais levantados e definimos os requisitos não funcionais essenciais da nossa aplicação. E definimos as plataformas e tecnologias que iremos usar.

A.5 5° SEMANA

Período: 22/04/2025 a 29/04/2025.

Essa semana foram desenvolvidas atividades ligadas ao desenvolvimento da prova de conceito. Foram elas: - Documentação dos Casos de Uso - Kelly - Documentação dos Diagramas de Caso de Uso -Guilherme Bittencourt. Reunião via Google Meet na qual foi discutido o desenho da aplicação com todos os membros da equipe. - Criação do Diagrama de Componentes -Guilherme Bittencourt. - Criação do Diagrama de Implantação -Kelly Radchelle. - Apresentação do Desenho da Aplicação - Guilherme Bittencourt, Guilherme Akio, Rafael Teixeira, Anna Julia e Ricardo Carriel.

A.6 6° SEMANA

Período: 29/04/2025 a 06/05/2025.

Após a entrega da prova de conceito na semana anterior os diagramas apresentados foram editados, corrigindo os pontos levantados pelo professor orientador. - Alterações no diagrama de componentes: Guilherme Bittencourt. - Alterações no diagrama de implantação: Kelly. - Criação do MER:Guilherme Bittencourt. - Criação do repositório Git para versionamento da aplicação: Guilherme Bittencourt. - Levantamento das regras de negócio e requisitos com a proprietária: Kelly, Ricardo.

A.7 7º SEMANA

Período: 06/05/2025 a 13/05/2025.

Nesta semana foram realizadas atividades para a entrega da POC (Prova de Conceito). Foram elas: Início do desenvolvimento do backend e frontend no Django: views, models e templates - Guilherme Akio. Início da configuração do ambiente de hospedagem - Ricardo Carriel. Além de continuarmos com as atividades de documentação e alimentação do nosso repositório Git.

A.8 8º SEMANA

Período: 13/05/2025 a 20/05/2025.

Nessa semana o Ricardo e o Guilherme Akio finalizaram a integração entre o ambiente de hospedagem, criação do banco de dados no postgresQL e o servidor Django. Assim, conseguimos finalizar, entregar e apresentar a prova de conceito. Além disso, aproveitamos para revisar os nossos requisitos e regras de negócio e estruturamos de forma mais completa nossos requisitos não funcionais.

A.9 9º SEMANA

Período: 20/05/2025 a 27/05/2025.

Nesta semana demos continuidade ao desenvolvimento do nosso MVP, focando agora nas funcionalidades e interfaces. Alimentamos nosso repositório git com arquivos referentes a documentação. - Atualizações no código- Ricardo. - Desenvolvimento da Documentação - Kelly. - Desenvolvimento da Documentação - Guilherme Bittencourt.

A.10 10º SEMANA

Período: 27/05/2025 a 03/06/2025.

Nesta semana os esforços da equipe foram voltados para o desenvolvimento e consolidação da documentação do projeto. Dessa forma, todos os integrantes tiveram como atividade a documentação e revisão de algum aspecto do sistema. Além disso, o integrante Rafael fez correções na extensão HTML.

A.11 11º SEMANA

Período: 03/06/2025 a 10/06/2025.

A.12 12° SEMANA

Período: 10/06/2025 a 17/06/2025.

A.13 13° SEMANA

Período: 17/06/2025 a 24/06/2025.