

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

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 – QR Code documento do ProjectLibre	25
Figura 2 – QR Code quadro Kanban no GitHub	26
Figura 3 – Diagrama de Implantação	55
Figura 4 – Diagrama de Componentes	56
Figura 5 – Modelo Entidade-Relacionamento (MER)	65
Figura 6 – Diagrama Entidade-Relacionamento (DER)	65
Figura 7 – Exemplificação da Estrutura do Repositório	66
Figura 8 – QR Code repositório do sistema no GitHub	67
Figura 9 – QR Code Documento de Testes Funcionais	71
Figura 10 – Resultado do teste SSL Server Test	120
Figura 11 – Resultado do teste de Headers de Segurança	121

Lista de quadros

1	Comparativo de funcionalidades entre Chalés Água de Coco e concorrentes	18
2	Divisão das funções e responsabilidades da equipe do projeto	22
3	Product Backlog (Parte 1)	23
3	Product Backlog (Parte 2)	24
4	Identificação dos Riscos do Projeto	27
5	Análise e Planejamento dos Riscos (Parte 1)	28
5	Análise e Planejamento dos Riscos (Parte 2)	29
6	Mecanismos de Monitoramento dos Riscos (Parte 1)	30
6	Mecanismos de Monitoramento dos Riscos (Parte 2)	31
7	Regras de Negócio (Parte 1)	33
7	Regras de Negócio (Parte 2)	34
8	Requisitos Funcionais (Parte 1)	35
8	Requisitos Funcionais (Parte 2)	36
8	Requisitos Funcionais (Parte 3)	37
9	Requisitos Não Funcionais (Parte 1)	38
9	Requisitos Não Funcionais (Parte 2)	39
10	Testes Funcionais: Categoria Quartos	71
11	Questionário Aplicado à Proprietária 1 (Parte 1)	99
11	Questionário Aplicado à Proprietária 1 (Parte 2)	100
11	Questionário Aplicado à Proprietária 1 (Parte 3)	101
12	Questionário Aplicado à Proprietária 2	102
13	Product Backlog Detalhado (Parte 1)	104
13	Product Backlog Detalhado (Parte 2)	105
13	Product Backlog Detalhado (Parte 3)	106
13	Product Backlog Detalhado (Parte 4)	107
14	Sprint Backlog Detalhado (Parte 1)	108
14	Sprint Backlog Detalhado (Parte 2)	109
14	Sprint Backlog Detalhado (Parte 3)	110
14	Sprint Backlog Detalhado (Parte 4)	111
14	Sprint Backlog Detalhado (Parte 5)	112
14	Sprint Backlog Detalhado (Parte 6)	113
14	Sprint Backlog Detalhado (Parte 7)	114
14	Sprint Backlog Detalhado (Parte 8)	115
15	Requisitos Funcionais (Parte 1)	116
15	Requisitos Funcionais (Parte 2)	117

15	Requisitos Funcionais (Parte 3)	118
15	Requisitos Funcionais (Parte 4)	119

Lista de tabelas

Tabela 1 – Resumo dos custos da infraestrutura de desenvolvimento	74
Tabela 2 – Resumo dos custos da infraestrutura de implantação e software	74
Tabela 3 – Quantidade e horas trabalhadas por função	76
Tabela 4 – Custos por função	76
Tabela 5 – Custo Recorrente Total por Categoria	77

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)
MVP	Minimum Viable Product (Produto Mínimo Viável)
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	14
1.1	Objetivos	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
1.2	Justificativa	16
1.3	Análise da Concorrência	17
1.3.1	SimplesHotel	17
1.3.2	HospedaJá	17
1.3.3	Diferenciais do Chalés Água de Coco	17
1.3.4	Comparativo	18
2	REVISÃO DE LITERATURA	19
2.1	Histórico do Turismo e da Hospitalidade	19
2.2	A Gestão Hoteleira e o Impacto da Tecnologia	20
3	GESTÃO DO PROJETO	21
3.1	Metodologia de Gestão e Desenvolvimento	21
3.2	Artefatos	22
3.2.1	Product Backlog	22
3.2.2	Sprint Backlog	24
3.3	Gestão de Riscos	26
3.3.1	Identificação dos Riscos do Projeto	26
3.3.2	Análise e Planejamento dos Riscos	28
3.3.3	Monitoramento dos Riscos	30
4	DESENVOLVIMENTO DO PROJETO	32
4.1	Escopo do Projeto	32
4.1.1	Funcionalidades do Sistema	32
4.1.2	Regras de Negócio	33
4.1.3	Requisitos do Sistema	34
4.1.3.1	Requisitos Funcionais	34
4.1.3.2	Requisitos Não Funcionais	37
4.2	Histórias de Usuário	40
4.2.1	Descrição das Histórias de Usuário	40
4.3	Arquitetura	52
4.3.1	Arquitetura Lógica	53

4.3.1.1	Camada de Apresentação (Template)	53
4.3.1.2	Camada de Controle (View)	53
4.3.1.3	Camada de Modelo (Model)	53
4.3.2	Banco de Dados	53
4.3.3	Infraestrutura em Nuvem	54
4.3.4	Preparação para Evoluções Futuras	54
4.3.5	Diagrama da Arquitetura	54
4.3.5.1	Diagrama de Implantação	54
4.3.5.2	Diagrama de Componentes	55
4.4	Tecnologias	56
4.4.1	Django	56
4.4.1.1	Front-end	57
4.4.1.2	Back-end	57
4.4.2	Banco de Dados	57
4.4.3	Infraestrutura em Nuvem com AWS	57
4.4.4	Boas Práticas de Desenvolvimento	58
4.4.5	Resumo da Escolha Tecnológica	58
4.5	Estilo de Código	58
4.5.1	Estrutura e Organização do Projeto	58
4.5.2	Convenções de Nomenclatura	59
4.5.3	Padrões de Código e Legibilidade	59
4.5.4	Considerações Finais sobre o Estilo	59
4.6	Ferramentas de Apoio	60
4.6.1	GitHub	60
4.6.2	BRModelo	60
4.6.3	Visual Paradigm Online	60
4.6.4	Latex	60
4.6.5	Google Meet	61
4.6.6	ProjectLibre	61
4.7	Manutenibilidade	61
4.8	Segurança, Privacidade e Legislação	62
4.8.1	Segurança da Aplicação	62
4.8.2	Segurança na Comunicação	63
4.8.3	Conformidade com a LGPD	63
4.9	Modelagem do Banco de Dados	64
4.9.1	Modelo Entidade-Relacionamento - MER	64
4.9.2	Diagrama Entidade-Relacionamento - DER	65
4.10	Repositório da Aplicação	66
4.11	Entregas	67

4.11.1	Desenvolvimento de um Tema - 08/04	67
4.11.2	Desenho da Aplicação - 29/04	67
4.11.3	Prova de Conceito - 20/05	68
4.11.4	Análise e Documentação - 10/06	68
4.11.5	Produto Mínimo Viável - 24/06	68
5	PLANO DE TESTES	69
5.1	Objetivo	69
5.2	Escopo do Teste	69
5.3	Análise Estática de Qualidade de Código	70
5.4	Testes Funcionais	71
5.5	Testes Unitários	72
6	VIABILIDADE FINANCEIRA	73
6.1	Custos	73
6.1.1	Custo Estrutural	73
6.1.2	Custo de Pessoal	75
6.1.3	Custo Total	76
6.2	Cenários de Comercialização	77
6.2.1	Cenário A: Aquisição da Propriedade Intelectual	77
6.2.2	Cenário B: Licenciamento de Uso e Manutenção	78
6.2.3	Análise de Viabilidade dos Cenários	79
7	CONSIDERAÇÕES FINAIS	81
8	CONCLUSÃO	82
	REFERÊNCIAS	83
	APÊNDICES	88
	APÊNDICE A – DIÁRIO DE BORDO	89
A.1	1° SEMANA	89
A.2	2° SEMANA	89
A.3	3° SEMANA	89
A.4	4° SEMANA	90
A.5	5° SEMANA	90
A.6	6° SEMANA	91
A.7	7° SEMANA	91

A.8	8° SEMANA	91
A.9	9° SEMANA	92
A.10	10° SEMANA	92
A.11	11° SEMANA	92
A.12	12° SEMANA	93
A.13	13° SEMANA	93
A.14	Período: 13/08/2025 a 26/08/2025.	93
A.15	14° SEMANA	93
A.16	15° SEMANA	94
A.17	16° SEMANA	94
A.18	17° SEMANA	95
A.19	18° SEMANA	95
A.20	19° SEMANA	96
A.21	20° SEMANA	96
A.22	21° SEMANA	97
A.23	22° SEMANA	97
A.24	23° SEMANA	97
A.25	24° SEMANA	98
 APÊNDICE B – QUESTIONÁRIO APLICADO À PROPRIETÁRIA DA POUSADA 99		
B.1	Questionário 1: Levantamento de Requisitos 99	
B.2	Questionário 2: Viabilidade Financeira 102	
 APÊNDICE C – PRODUCT BACKLOG DETALHADO 104		
 APÊNDICE D – SPRINT BACKLOG DETALHADO 108		
 APÊNDICE E – REQUISITOS FUNCIONAIS DETALHADOS 116		
 APÊNDICE F – CERTIFICADOS DE SEGURANÇA 120		
F.1	Certificado SSL 120	
F.2	Certificado Headers de Segurança 121	
 APÊNDICE G – RELATÓRIO DE TESTES FUNCIONAIS - MÓDULO DE GESTÃO DE HÓSPEDES 122		
 APÊNDICE H – RELATÓRIO DE TESTES FUNCIONAIS - MÓDULO DE GESTÃO DE QUARTOS 128		
 APÊNDICE I – RELATÓRIO DE TESTES FUNCIONAIS - MÓDULO DE GESTÃO DE RESERVAS 132		

APÊNDICE J – RELATÓRIO DE TESTES FUNCIONAIS - FINANCEIRO 139

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, essa mudança tem se mostrado essencial para garantir maior eficiência, organização e qualidade no atendimento ao cliente. Segundo dados do Ministério do Turismo, cerca de 89% dos empreendimentos no setor de turismo no Brasil são compostos por micro e pequenas empresas ([MINISTERIO DO TURISMO, 2025](#)), evidenciando a importância de soluções tecnológicas adequadas para esse perfil de organização.

Apesar dos avanços tecnológicos, muitas pousadas e hotéis de pequeno porte ainda utilizam métodos manuais ou ferramentas limitadas, como planilhas eletrônicas, para controlar reservas, hospedagens e finanças. Essa prática, embora funcional em um primeiro momento, apresenta restrições consideráveis, como risco elevado de erros, dificuldade de atualização em tempo real e ausência de acessibilidade remota ([GESTÃO PRO, 2025](#)).

No caso da Pousada Chalés Água de Coco, objeto de estudo deste trabalho, a gestão de suas operações é realizada exclusivamente por meio do *Microsoft Excel*. Essa limitação reforça a necessidade de modernização dos processos administrativos, visto que a utilização de sistemas *web* permite centralizar informações, reduzir falhas e oferecer maior segurança no armazenamento de dados. Estudos sobre o setor hoteleiro demonstram que a adoção de ferramentas digitais modernas favorece a integração das operações e melhora significativamente a experiência do hóspede ([TOTVS, 2024](#)).

Nesse contexto, identificou-se a necessidade de desenvolver uma solução tecnológica personalizada para a pousada, de modo a otimizar a organização dos dados, aprimorar o controle de reservas e disponibilizar relatórios gerenciais que apoiem a tomada de decisão. A parceria estabelecida com a Pousada Chalés Água de Coco possibilitou o levantamento das principais dificuldades enfrentadas na gestão atual, servindo como base para a construção do sistema proposto.

Este trabalho tem, portanto, como objetivo geral o desenvolvimento de um sistema *web* para automatizar os processos administrativos da pousada. Como objetivos específicos, destacam-se: realizar o levantamento de requisitos junto aos gestores da pousada; projetar e implementar uma plataforma integrada para o gerenciamento de hóspedes, reservas e finanças; e avaliar os benefícios da aplicação proposta em relação ao método atual de gestão.

1.1 Objetivos

Os objetivos deste trabalho foram definidos com base nas necessidades identificadas na Pousada Chalés Água de Coco. Para melhor organização e clareza, foram divididos em duas categorias: o objetivo geral, que apresenta a finalidade principal do projeto, e os objetivos específicos, que detalham as etapas necessárias para o seu alcance.

1.1.1 Objetivo Geral

Desenvolver um sistema *web* para automatizar os processos administrativos da pousada Chalés Água de Coco. Dessa forma, o objetivo central deste trabalho é propor uma solução que centralize informações de hóspedes, reservas, acomodações e finanças em uma plataforma única, acessível via Internet, promovendo maior organização, segurança e suporte à tomada de decisão.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral proposto, foram definidos os seguintes objetivos específicos, que orientaram o processo de desenvolvimento do sistema e da documentação técnica:

- Realizar reuniões com a proprietária da pousada para o levantamento de requisitos e regras de negócio, identificando as principais necessidades operacionais e os fluxos administrativos existentes;
- Elaborar a modelagem do banco de dados com base nas informações coletadas, estruturando as entidades e relacionamentos necessários;
- Projetar a arquitetura do sistema conforme o padrão *Model-Template-View* (MTV) do *framework* Django;
- Implementar as funcionalidades essenciais do sistema, incluindo o cadastro e gerenciamento de hóspedes, acomodações, reservas e movimentações financeiras;
- Integrar o sistema com o banco de dados PostgreSQL hospedado na infraestrutura de nuvem AWS, garantindo persistência e segurança das informações;
- Aplicar recursos de segurança e privacidade de dados, tais como autenticação de usuários, criptografia de comunicações via *Hypertext Transfer Protocol Secure* (HTTPS);
- Desenvolver relatórios gerenciais para apoiar a tomada de decisão, com foco em indicadores financeiros e operacionais relevantes à gestão da pousada;

- Implementar testes automatizados (*unit tests*) utilizando *pytest-django* para validar o funcionamento das principais funcionalidades do sistema;
- Documentar detalhadamente o projeto, abrangendo os aspectos de análise, arquitetura, desenvolvimento, segurança, implantação e manutenção do sistema, entre outros;
- Entregar a versão final do código e da documentação técnica, consolidando o produto como uma solução funcional e aplicável ao contexto da pousada Chalés Água de Coco.

1.2 Justificativa

A escolha deste projeto se justifica pela necessidade real de modernização enfrentada por pequenos empreendimentos do setor de hospedagem ([FGV - Fundação Getúlio Vargas, 2022](#)), como a pousada Chalés Água de Coco, que atualmente depende de controles manuais realizados por meio de planilhas no Excel. Esse tipo de gestão, que depende de planilhas, tem base no fato de que estudos recentes mostram que cerca de 65% dos empresários de pequenas empresas utilizam planilhas para controle, o que aponta para limitações estruturais, como erros humanos, falta de integração entre os dados e dificuldade de atualização simultânea e de acesso remoto ([SEBRAE, 2020](#)).

No contexto do setor de turismo e hotelaria, estudos indicam que a transformação digital tem se consolidado como um fator estratégico para aumentar a competitividade e a qualidade dos serviços ([LAUDON; LAUDON, 2020; STAIR; REYNOLDS, 2020](#)). O uso de sistemas de informação pode proporcionar ganhos em eficiência, reduzir custos operacionais e melhorar a experiência do cliente, aspectos essenciais para a sobrevivência de pequenos empreendimentos em um mercado cada vez mais exigente. De acordo com o Ministério do Turismo ([MINISTERIO DO TURISMO, 2025](#)), pequenos meios de hospedagem representam parcela significativa da rede de serviços no Brasil, desempenhando papel relevante no desenvolvimento econômico local. Portanto, a adoção de tecnologias adequadas contribui não apenas para a gestão interna, mas também para a valorização do setor como um todo.

Por fim, a solução proposta tem potencial de gerar impacto direto e positivo na gestão da pousada parceira, ao promover maior organização, acessibilidade e controle das operações. Do ponto de vista acadêmico, o projeto contribui para o avanço do conhecimento na área de sistemas aplicados ao setor de serviços, e do ponto de vista social e econômico, fortalece a sustentabilidade e a competitividade de pequenos empreendimentos locais. Dessa forma, o presente trabalho se mostra relevante tanto no âmbito prático quanto no

científico, atendendo às exigências de um Trabalho de Conclusão de Curso alinhado às necessidades da sociedade e do mercado.

1.3 Análise da Concorrência

No segmento de sistemas de gestão hoteleira, diversas plataformas disputam espaço no mercado oferecendo soluções voltadas para pousadas e hotéis. A seguir, são analisados dois concorrentes diretos do sistema Chalés Água de Coco, com o objetivo de destacar os diferenciais e vantagens competitivas do sistema proposto.

1.3.1 SimplesHotel

O SimplesHotel é uma plataforma de gestão hoteleira voltada para estabelecimentos de pequeno a grande porte. Seu foco está na automação dos processos administrativos e operacionais do setor de hospitalidade. A aplicação é baseada na *Web*, com dados armazenados na nuvem.

A monetização ocorre por meio de planos mensais com preços escalonados conforme o número de acomodações. Alguns módulos importantes, como emissão de Nota Fiscal de Serviço Eletrônica (NFS-e), motor de reservas e envio de mensagens curtas (SMS), são cobrados separadamente, o que pode elevar significativamente o custo final da solução.

Entre suas funcionalidades estão: gerenciador de reservas, controle de estoque e finanças, *web check-in*, emissão de notas fiscais, suporte multicanal e integração com canais de venda como Booking e Expedia.

1.3.2 HospedaJá

O HospedaJá é uma solução de gestão para pousadas e hotéis de pequeno e médio porte, com foco na organização de reservas, hospedagens e finanças. Assim como o SimplesHotel, é uma aplicação *web* baseada em nuvem e oferece *backups* automáticos.

Seu modelo de cobrança também se baseia em planos mensais variáveis conforme o número de quartos e usuários. O suporte é oferecido por e-mail e sistema de chamados.

As funcionalidades principais incluem: controle de reservas e hospedagens, mapa de ocupação, controle financeiro, geração de relatórios e controle de estoque de produtos e serviços.

1.3.3 Diferenciais do Chalés Água de Coco

O sistema Chalés Água de Coco se destaca por oferecer uma solução enxuta, prática e de baixo custo, especialmente desenhada para as necessidades específicas da pousada

homônima. Diferente dos concorrentes, o sistema foi desenvolvido sob medida, com foco na simplicidade, eficiência e controle financeiro, eliminando a dependência de planos caros e de funcionalidades excedentes que muitas vezes não são utilizadas em pequenos negócios.

1.3.4 Comparativo

Quadro 1 – Comparativo de funcionalidades entre Chalés Água de Coco e concorrentes

Funcionalidade	Chalés Água de Coco	SimplesHotel	HospedaJá
Gerenciamento de Quartos	X	X	X
Gerenciamento de Hóspedes	X	X	X
Gerenciamento de Reservas	X	X	X
Controle Financeiro	X	X	X
Relatórios Personalizados	X		X
Controle de Estoque		X	X
Web Check-in		X	
Integração com OTAs (Booking, Expedia etc.)		X	
Cobrança de Módulos Adicionais		X	X
Custo Total Mensal Baixo	X		
Foco em Pousadas Locais (customização)	X		

Fonte: Elaborado pelos autores.

2 Revisão de Literatura

Neste tópico é revisado o histórico do uso da tecnologia na gestão de negócios de hotelaria na literatura científica com o objetivo de contextualizar e destacar a relevância do projeto.

2.1 Histórico do Turismo e da Hospitalidade

O turismo é a atividade de deslocamento geográfico de uma pessoa para fora de seu ambiente habitual por um período de até um ano, com diversas motivações, exceto emprego remunerado no local de destino ([United Nations, 2010](#)). No Brasil, essa atividade se destaca no setor de serviços, representando uma força econômica significativa. Essa atividade faturou R\$ 108 bilhões no primeiro semestre de 2025, um resultado impulsionado, principalmente, pelos meios de hospedagem, que sozinhos movimentaram mais de R\$ 13 bilhões ([FECOMERCIO, 2025](#)).

Historicamente, a hospitalidade está intrinsecamente ligada à evolução do turismo. O turismo surgiu no momento em que o homem abandonou o sedentarismo, motivado pela necessidade de sobrevivência. De forma que, as primeiras hospedagens nasceram como estruturas familiares sem caráter comercial, nas quais pessoas cediam alojamento e alimentação para estranhos para que estes tivessem condições para seguir viagem ([ALDRIGUI, 2007](#)). Por volta de 500 a.C., na Roma Antiga a nobreza percorria grandes distâncias para visitar templos e realizar banhos medicinais, o que demandou a construção de postos de troca de animais e motivou o surgimento das primeiras hospedarias comerciais ([IGNARRA, 2013](#)). Já com o crescimento do turismo e as transformações sociais e econômicas na Idade Moderna, houve o aumento de estruturas físicas e sociais para atender as necessidades dos viajantes. Essa evolução deu origem a uma ampla variedade de meios de hospedagem, dentre eles as pousadas.

O meio de hospedagem qualificado como pousada é, em sua essência, uma versão contemporânea das antigas hospedarias, surgindo como uma alternativa de hospedagem mais acessível e com um ambiente acolhedor ([ZANELLA; ANGELONI, 2006](#)). De acordo com a Portaria nº 100 de 2011 do Ministério do Turismo, uma pousada é classificada como um "empreendimento de característica horizontal, composto de no máximo 30 unidades habitacionais e 90 leitos, com serviços de recepção, alimentação e alojamento temporário"([BRASIL, 2011](#)).

Esses estabelecimentos se distinguem das grandes redes hoteleiras por seu modelo de gestão, que é frequentemente familiar e confere um toque de personalidade à experiência

do hóspede. Por sua estrutura e escala, geralmente são encontradas em áreas rurais, cidades históricas ou locais com natureza exuberante, promovendo uma vivência mais autêntica e imersiva para os viajantes (MOURA, 2023). No entanto, essa mesma característica de negócio de pequeno porte e administração simplificada, que as torna únicas, também acarreta desafios. A gestão, muitas vezes manual e sem o suporte de sistemas robustos, limita o crescimento e a eficiência desses empreendimentos, criando uma lacuna tecnológica que compromete sua capacidade de competir no mercado.

2.2 A Gestão Hoteleira e o Impacto da Tecnologia

A gestão de uma pousada, assim como de outros empreendimentos hoteleiros, visa atender às expectativas dos clientes e garantir sua fidelização (ZANELLA; ANGELONI, 2006). Para isso, suas atividades são estruturadas em diferentes áreas, como Hospedagem, Finanças e *Marketing*, o que requer uma coordenação integrada para garantir uma experiência satisfatória ao hóspede, um fator que é vital para a sobrevivência do negócio (MARTINS; GONDIM, 2011). Em um mercado cada vez mais competitivo, a qualidade do serviço é o principal diferencial. Por isso, o gestor deve ser ágil e prático em suas tomadas de decisão, buscando continuamente informações para se adaptar às mudanças do mercado e inovar em seus serviços (MAURÍCIO; RAMOS, 2011).

Nesse contexto, as tecnologias da informação surgem como ferramentas essenciais para aumentar a competitividade no setor hoteleiro (BUHALIS, 1998). Elas facilitam o acesso a informações estratégicas, auxiliando na customização de produtos, na redução de custos e na construção de relacionamentos mais próximos com os clientes. O uso de sistemas computacionais, em particular, otimiza o fluxo de dados, fornecendo informações precisas que são vitais para o sucesso do negócio (HOFFMANN; OLIVEIRA; ZEFERINO, 2012). Nesse sentido, os softwares PMS (*Property Management System*) se tornaram indispensáveis, centralizando atividades como gestão de reservas, *check-in*, *check-out* e finanças.

Embora os PMS tenham substituído registros em papel e aprimorado a execução de processos, muitos empreendimentos menores, como as pousadas, enfrentam desafios de recursos e falta de capacitação. Essa limitação dificulta a adoção de softwares robustos, mantendo-os distantes de soluções que poderiam fornecer uma visão mais estratégica e completa do negócio (LUNARDI; DOLCI; MAçADA, 2010).

Dante desse cenário, este projeto propõe o desenvolvimento de um sistema *web* de gestão para a pousada Chalés Água de Coco. A ferramenta foi idealizada para atender às necessidades específicas do empreendimento de forma acessível e eficiente, com o objetivo de impulsionar seu crescimento e fortalecer sua competitividade no mercado local, oferecendo suporte à proprietária para tomadas de decisão mais ágeis e assertivas.

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 Metodologia de Gestão e Desenvolvimento

A principal ferramenta de gestão e desenvolvimento do projeto adotada pela equipe responsável pelo projeto foi a metodologia ágil Scrum. 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. Diante disso, a equipe adaptou a metodologia de forma estratégica às condições do projeto, considerando prazos, recursos e estrutura disponíveis.

Para atender às restrições de tempo e à execução majoritariamente remota do projeto, os eventos de inspeção do progresso que são propostos para serem diários, o *daily scrum*, foram realizados de forma assíncrona e em menor frequência. Enquanto as entregas semanais de incrementos utilizáveis do produto foram mantidas, de forma que os eventos de planejamento, revisão e retrospectiva (*Sprint Planning*, *Sprint Review* e *Sprint Retrospective*) foram consolidados em reuniões semanais, assegurando um processo eficiente e a inspeção contínua do projeto.

Adicionalmente, a divisão das funções e responsabilidades dos membros da equipe combinou competências técnicas individuais com papéis Scrum (*Product Owner*, *Scrum Master* e *Development Team*). Essa distribuição facilitou a colaboração entre os integrantes, dada a especialização do trabalho, e tornou os processos mais eficientes. As funções, papel Scrum e responsabilidades de cada membro estão detalhadas no [Quadro 2](#).

Essas adaptações flexibilizaram os processos, conferindo maior autonomia aos integrantes para organizar seus horários e ritmos de trabalho. No entanto, o número reduzido de reuniões e tempo limitado de colaboração síncrona dificultaram a comunicação e a definição de *sprints* mais produtivos, exigindo maior revisão do planejamento.

Quadro 2 – Divisão das funções e responsabilidades da equipe do projeto

Integrante	Função	Papel Scrum	Responsabilidades
Guilherme Akio	Analista de Testes	Time de Desenvolvimento	Implementar, executar e documentar testes; Entrega de funcionalidades relacionadas a qualidade e performance do sistema
Guilherme Bittencourt	Engenheiro de Software	Scrum Master	Promover e facilitar a aplicação da teoria e das práticas do Scrum; Definir e organizar a estrutura do sistema
Kelly Radchelle	Gerente de Projeto	Product Owner	Acompanhar o progresso das <i>sprints</i> ; Facilitar e coordenar tomada de decisões e comunicação entre as partes interessadas
Rafael Teixeira	Desenvolvedor Front-end	Time de Desenvolvimento	Entregas de funcionalidades relacionadas à experiência do usuário; Projetar e implementar interfaces da aplicação
Ricardo Carriel	Desenvolvedor Back-end	Time de Desenvolvimento	Entregas de funcionalidades relacionadas a lógica e integração do sistema; Implementar lógica de negócio; Configurar e administrar o servidor

Fonte: Elaborado pelos autores.

3.2 Artefatos

No Scrum, os artefatos são elementos que ajudam a equipe a consolidar a transparência no processo de desenvolvimento (SCHWABER; SUTHERLAND, 2013). Diante disso, o *time scrum* realizou o planejamento dos artefatos: *Product Backlog* e *Sprints Backlog*.

3.2.1 Product Backlog

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 (SCHWABER; SUTHERLAND, 2013). Por isso, o *product owner* do projeto elaborou o *product backlog* inicial da aplicação *web* de gestão da pousada Chalés Água de Coco (Quadro 3). Enquanto, o *product backlog* completo do projeto esta detalhado no Apêndice C.

A lista foi elaborada com base nas histórias de usuário levantadas pela equipe, considerando que essas histórias refletem fielmente as necessidades e expectativas da usuária, assegurando o caráter utilitário do *product backlog*. Além disso, foi atribuída uma classificação de prioridade a cada item, variando entre alta, média e baixa. Essa priorização definiu como de alta prioridade os itens essenciais para a entrega do *Minimum Viable Product* (MVP) do sistema; de prioridade média, aqueles necessários para a conclusão do escopo final do projeto, mas não previstos para a entrega do MVP; e de prioridade baixa, os itens importantes, porém que não comprometem o funcionamento básico nem a entrega final do sistema.

Quadro 3 – Product Backlog (Parte 1)

Código	Item	Categoria	Prioridade
1	Definir, configurar e organizar ambientes de desenvolvimento e hospedagem	Requisito Técnico	ALTA
2	Levantar e documentar requisitos funcionais e não funcionais	Modelagem de dados	ALTA
3	Criar e mapear casos de uso e histórias de usuário	Modelagem de dados	ALTA
4	Criar e documentar MER e DER	Modelagem de dados	ALTA
5	Criar e documentar diagramas de componentes e de implantação	Arquitetura	ALTA
6	Implementar funcionalidades de autenticação de usuário	Autenticação e Segurança	ALTA
7	Implementar funcionalidades de cadastro, edição e exclusão de quartos	Gestão de Quartos	ALTA
8	Implementar funcionalidades de cadastro, edição e exclusão de hóspedes	Gestão de Hóspedes	ALTA
9	Implementar funcionalidades de cadastro, edição e exclusão de reservas	Gestão de Reservas	ALTA
10	Implementar interfaces de usuário	Requisito Técnico	ALTA
11	Implementar funcionalidades de Check-in e Check-out	Check-in e Check-out	ALTA
12	Integrar ambientes back-end e front-end	Requisito técnico	ALTA

Fonte: Elaborado pelos autores.

Quadro 3 – Product Backlog (Parte 2)

Código	Item	Categoria	Prioridade
14	Implementar funcionalidades de cadastro, edição e exclusão de receitas	Gestão Financeira	MÉDIA
15	Implementar funcionalidade de registro de pagamento de reservas	Gestão Financeira e Gestão de Reservas	MÉDIA
16	Implementar funcionalidades de cadastro, edição e exclusão de despesas	Gestão Financeira	MÉDIA
17	Implementar funcionalidades de geração de relatórios	Gestão Financeira	BAIXA
18	Implementar balanço financeiro simples (receitas, despesas, saldo) por período	Gestão Financeira	BAIXA
19	Implementar filtros de visualização de receita/despesa (data, categoria)	Gestão Financeira	BAIXA
20	Implementar funcionalidades de notificação	Notificações	BAIXA
21	Implementar configurações de segurança	Autenticação e Segurança	ALTA
22	Definir, executar e documentar plano de testes	Testes	MÉDIA
23	Elaborar e executar o plano de implantação	Implantação	MÉDIA
24	Treinamento da proprietária da poussada para uso da aplicação	Implantação	MÉDIA

Fonte: Elaborado pelos autores.

3.2.2 Sprint Backlog

O *Sprint Backlog* constitui o plano de execução do projeto, derivado do *Product Backlog*, e estabelece o conjunto de itens de trabalho a ser realizado em cada *sprint*. Nesse contexto, o artefato foi desenvolvido pela equipe do projeto como ferramenta auxiliar na gestão, com o objetivo de garantir que todas as partes estivessem cientes do andamento e das etapas em desenvolvimento. O plano completo encontra-se disponível no [Apêndice D](#).

As *sprints* foram organizadas em períodos de sete dias (uma semana), intervalo estabelecido em conformidade com o Guia do Scrum ([SCHWABER; SUTHERLAND, 2013](#)), totalizando 25 semanas de trabalho. Essa definição considerou as circunstâncias acadêmicas que envolvem o projeto, as quais exigiam entregas semanais e acompanhamento

contínuo das atividades.

Dessa forma, os itens do *Product Backlog* foram distribuídos de modo a garantir um progresso gradual das atividades, organizadas em duas fases. Na primeira fase, as *sprints* iniciais concentraram-se na modelagem do sistema e na escolha das ferramentas e tecnologias de gestão e desenvolvimento. Nas *sprints* seguintes, foram iniciadas as tarefas relacionadas ao desenvolvimento do MVP e à documentação do sistema. Já na segunda fase, a equipe concentrou esforços na implementação das funcionalidades de gestão financeira, na integração com o MVP, na realização de testes e na revisão da documentação. As atividades de desenvolvimento foram distribuídas considerando critérios de prioridade e relações de dependência entre tarefas.

Para auxiliar a equipe na visualização, acompanhamento e organização das *sprints*, foram utilizadas duas ferramentas de gestão distintas. Na primeira fase do projeto, empregou-se o ProjectLibre, no qual foram cadastradas as atividades com seus respectivos prazos e responsáveis [Figura 1](#). No entanto, devido a dificuldades técnicas e a demandas externas, a equipe passou a utilizar, na segunda fase do projeto, um quadro Kanban. Essa mudança proporcionou maior flexibilidade no controle das tarefas e autonomia para cada membro da equipe gerenciar suas próprias atividades, especialmente pela familiaridade de alguns integrantes com esse modelo. O quadro foi criado no próprio repositório do projeto e está disponível na seção *projects* ([Figura 2](#)). A transição entre as ferramentas permitiu otimizar a comunicação interna e o acompanhamento do progresso das entregas, contribuindo para uma gestão mais ágil das *sprints*.

Figura 1 – QR Code documento do ProjectLibre



Fonte: Elaboração própria.

Figura 2 – QR Code quadro Kanban no GitHub



Fonte: Elaboração própria.

3.3 Gestão de Riscos

Os riscos de um projeto de *software* são elementos ou eventos de origem multifatorial que, caso ocorram, impactam negativamente o cronograma, os custos, a qualidade ou o escopo do projeto. Dessa forma, o gerenciamento de riscos é um processo iterativo fundamental para prever os riscos associados ao desenvolvimento de um projeto, visto que, promove a compreensão dos riscos com vistas à sua previsão, detecção e tratamento, de modo a não comprometer a entrega do produto final ([SOMMERVILLE, 2011](#), p. 416).

Diante disso, a equipe do projeto de desenvolvimento do sistema de gestão da Pousada Chalés Água de Coco 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*.

Para isso, a equipe elaborou um plano 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.

De forma que, 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.

3.3.1 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 da análise das características inerentes ao desenvolvimento de uma aplicação *web*, da equipe Scrum e do próprio negócio e foram registradas no [Quadro 4](#).

Quadro 4 – Identificação dos Riscos do Projeto

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
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 domínio 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.

3.3.2 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 5](#).

Quadro 5 – Análise e Planejamento dos Riscos (Parte 1)

Risco	Impacto	Probabilidade de Ocorrência	Estratégias de Prevenção	Estratégias de Contingência
R01	Baixo	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	Baixo	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.

Fonte: Elaborado pelos autores.

Quadro 5 – Análise e Planejamento dos Riscos (Parte 2)

Risco	Impacto	Probabilidade de Ocorrência	Estratégias de Prevenção	Estratégias de Contingência
R05	Médio	Alta	Definir junto a com o time de desenvolvimento o número de <i>sprints</i> viáveis para o desenvolvimento; Moniturar 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.
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.

3.3.3 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 6](#), 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*.

Quadro 6 – Mecanismos de Monitoramento dos Riscos (Parte 1)

Risco	Indicadores	Métricas
R01	<ul style="list-style-type: none"> - Time Scrum 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 <i>backups</i>; - 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>).

Fonte: Elaborado pelos autores.

Quadro 6 – Mecanismos de Monitoramento dos Riscos (Parte 2)

Risco	Indicadores	Métricas
R08	- Gastos com ferramentas para funcionalidades não previstas.	- Número de recursos contratados fora do planejamento inicial.
R09	- Módulos desenvolvidos em paralelo sem o uso de práticas de integração contínua; - Carência de testes de integração; - Modelos mal padronizados.	- Número de <i>bugs</i> relacionados à comunicação entre módulos.
R10	- Carência de mecanismos como HTTPS e <i>headers</i> de segurança; - Dados sensíveis expostos em variáveis de ambiente.	- Número de falhas identificadas durante testes de segurança.

Fonte: Elaborado pelos autores.

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.

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

O presente tópico delimita o sistema *web* de gestão desenvolvido para a Pousada Chalés Água de Coco, descrevendo suas funcionalidades, requisitos e restrições. Estes que foram levantados por meio da aplicação de um questionário à proprietária ([Apêndice B](#)) e da análise do site institucional da pousada ([COCO, 2025](#)) pela equipe responsável, com o objetivo de assegurar que o produto final atenda as necessidades da pousada de forma coerente, segura e eficiente, e apoie, no contexto de sua gestão, os processos de tomada de decisão.

4.1.1 Funcionalidades do Sistema

O escopo deste projeto define que o produto final é composto por funcionalidades organizadas em cinco módulos principais:

- Gestão de Acesso: funcionalidades de *login* e *logout* da administradora, garantindo acesso seguro e restrito ao sistema.
- Gestão Financeira: controle de pagamentos, movimentação de receitas e despesas, além da geração de relatórios financeiros básicos para suporte à administração da pousada.
- Gestão de Hóspedes: cadastro, atualização e consulta de informações dos hóspedes, incluindo histórico de hospedagem.
- Gestão de Quartos: cadastro, edição, exclusão e monitoramento da ocupação dos quartos, assegurando a atualização de disponibilidade correta em tempo real.
- Gestão de Reservas: cadastro, edição, cancelamento de reservas e gerenciamento de *check-in* e *check-out*, garantindo o controle completo do fluxo de hospedagem.

Estas funcionalidades atuam de forma integrada, através de uma interface intuitiva, desenvolvida exclusivamente para o uso de uma única usuária: a proprietária da pousada.

No entanto, com o objetivo de garantir a entrega do produto final dentro do prazo, o sistema não abrange funcionalidades que necessitam de integrações com sistemas externos, como confirmação automática de pagamentos ou processamento de reservas em plataformas externas. O que não descarta a possibilidade de que, em etapas futuras, o sistema seja atualizado para incluir integrações com serviços de mensageria, a fim de ampliar a automação dos processos de reserva da pousada.

4.1.2 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). Dessa forma, as regras de negócio permitem a compreensão da lógica de funcionamento da empresa, sendo assim, um fator essencial para o desenvolvimento de sistemas computacionais alinhados com as necessidades reais da mesma.

Neste contexto, para auxiliar a equipe na modelagem e desenvolvimento aplicação, foi realizado o levantamento das regras de negócio do sistema *web* de gestão da pousada Chalés Água de Coco, disponíveis no [Quadro 7](#).

Quadro 7 – Regras de Negócio (Parte 1)

Regra	Descrição
RN01	Todas as operações de gestão da pousada devem ser realizadas exclusivamente no sistema, com acesso restrito à proprietária.
RN02	Para efetuar uma reserva, os seguintes dados do hóspede são obrigatórios: nome completo, endereço completo, CPF, telefone e e-mail.
RN03	Um mesmo hóspede pode ter mais de uma reserva ativa.
RN04	As reservas devem ser registradas com, no mínimo, 2 dias de antecedência da data de entrada.
RN05	A confirmação de uma reserva deve ocorrer mediante o pagamento de 50 por cento do valor total da reserva.
RN06	São aceitas as formas de pagamento: Pix, dinheiro e cartão (com taxa da operadora).
RN07	Deve ser enviada a confirmação da reserva para o cliente com todos os dados necessários.
RN08	Cancelamentos e remarcações são permitidos, sujeitos a regras e taxas específicas.

Fonte: Elaborado pelos autores.

Quadro 7 – Regras de Negócio (Parte 2)

Regra	Descrição
RN09	Não há oferta de serviços adicionais vinculados à reserva.
RN10	Os horários padrão de <i>check-in</i> e <i>check-out</i> são: <i>check-in</i> das 16h às 22h; <i>check-out</i> das 8h às 14h. Em períodos de temporada os horários mudam para: <i>check-in</i> das 15h às 22h; <i>check-out</i> das 8h às 11h.
RN11	A emissão de recibos ou comprovantes após <i>check-in/out</i> não é obrigatória.
RN12	São permitidos <i>early check-in</i> e <i>late check-out</i> com cobrança de tarifas adicionais.
RN13	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.
RN14	Cada quarto tem uma capacidade máxima de pessoas.
RN15	Um quarto pode ficar indisponível para manutenção.
RN16	É permitido reservar um mesmo quarto para hóspedes diferentes em datas seguidas.

Fonte: Elaborado pelos autores.

4.1.3 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. Dessa forma, refletem as necessidades dos clientes e, consequentemente, descrevem a estrutura e a operação do sistema. Por isso, a fim de orientar a equipe de desenvolvimento na modelagem da aplicação *web* de gestão da Pousada Chalés Água de Coco, foram levantados os requisitos funcionais e não funcionais do sistema.

4.1.3.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades e serviços do sistema *web* de gestão da pousada e estão listados resumidamente no [Quadro 8](#). O quadro completo está disponível no [Apêndice E](#).

Quadro 8 – Requisitos Funcionais (Parte 1)

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF01	A aplicação deve possuir um sistema de <i>login</i> para a proprietária acessar a aplicação de forma segura, permitindo que ela altere ou recupere a sua senha de acesso sob demanda	Alta	RN01
RF02	O sistema deve permitir o cadastro de novas reservas, desde que associadas um quarto e a um período (data de <i>check-in</i> e <i>check-out</i>) disponíveis	Alta	RN04; RN14; RN15; RN16
RF03	O sistema deve exigir os dados pessoais do hóspede para que a reserva seja cadastrada: nome completo, endereço completo, data de nascimento, CPF/Passaporte, telefone e e-mail	Alta	RN02
RF04	O sistema deve impedir o registro de reservas com menos de 2 dias de antecedência da data do <i>check-in</i>	Média	RN04
RF05	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	RN05
RF06	O sistema deve permitir o registro de pagamento da reserva (forma, valor, data)	Média	RN05; RN06
RF07	A proprietária deve conseguir cancelar ou editar uma reserva já cadastrada, com possível registro do motivo	Média	RN08
RF08	A proprietária deve conseguir cadastrar mais de uma reserva no nome de um mesmo hóspede	Média	RN03
RF09	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	RN10; RN16

Fonte: Elaborado pelos autores.

Quadro 8 – Requisitos Funcionais (Parte 2)

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF10	O sistema deve permitir configurar tarifas de reserva com base nas regras de negócio da pousada	Média	RN13
RF11	O sistema deve enviar uma notificação para o hóspede via <i>e-mail</i> após a confirmação da reserva	Baixa	RN07
RF12	O sistema deve permitir que a proprietária acesse o histórico de reservas e edite ou exclua os dados de hóspedes cadastrados	Média	RN02; RN03;
RF13	A proprietária deve poder fazer o cadastro de quartos (incluindo informações como número/nome do quarto, capacidade, tipo e preço por noite) e poder editar ou excluir o cadastro posteriormente	Alta	RN04; RN15
RF14	O sistema deve permitir a visualização dos quartos disponíveis no período de tempo selecionado para a reserva	Média	RN14; RN16
RF15	O sistema deve gerar relatórios de ocupação de quartos em períodos definidos	Média	RN14; RN10; RN15; RN16
RF16	O sistema deve mudar o status do quarto automaticamente ao registrar o <i>check-in</i> ou do <i>check-out</i> e permitir que a proprietária altere manualmente o status quando necessário (ex.: disponível, ocupado, em manutenção)	Alta	RN10; RN15
RF17	O sistema deve permitir à proprietária configurar os horários padrão de <i>check-in</i> e <i>check-out</i>	Alta	RN10
RF18	O sistema deve permitir o registro de <i>early check-in</i> e <i>late check-out</i> para uma reserva, bem como a configuração de tarifas adicionais correspondentes a essas opções	Média	RN12, RN13

Fonte: Elaborado pelos autores.

Quadro 8 – Requisitos Funcionais (Parte 3)

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF19	A proprietária deve poder registrar, editar e excluir 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	RN01
RF20	A proprietária deve poder registrar, editar e excluir receitas, associando-as a uma reserva ou a outras fontes de receita, especificando a data, o valor e uma descrição da receita	Média	RN01
RF21	O sistema deve permitir a filtragem das transações por tipo (receita/despesa), data e categoria	Média	RN01
RF22	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	RN01
RF23	O sistema deve gerar relatórios de faturamento por período	Média	RN01
RF24	O sistema deve apresentar um painel (<i>dashboard</i>) com métricas-chave da pousada	Média	Requisito essencial para a gestão e visualização do negócio
RF25	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.

4.1.3.2 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. E estão listados no [Quadro 9](#).

Quadro 9 – Requisitos Não Funcionais (Parte 1)

Código	Módulo	Descrição
RNF01	Usabilidade	A interface do sistema deve ser intuitiva, responsiva (compatível e adaptada tanto para dispositivos <i>desktop</i> quanto <i>mobile</i>) 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 <i>feedback</i> 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	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.
RNF05	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.

Fonte: Elaborado pelos autores.

Quadro 9 – Requisitos Não Funcionais (Parte 2)

Código	Módulo	Descrição
RNF06	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.
RNF07	Confiabilidade	O sistema deve implementar mecanismos de tratamento de erros para que falhas e perdas de dados sejam prevenidas.
RNF08	Confiabilidade	O <i>deploy</i> 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.
RNF09	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.
RNF10	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.
RNF11	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.
RNF12	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.

4.2 Histórias 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 e-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:** RF01.

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:** RF01.

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:** RF02.

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:** RF07.

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:** RF02 e RF14.

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, data de nascimento, CPF/Passaporte, telefone e e-mail).
 - Todos os campos devem ter validação de formato.
 - O campo de data de nascimento deve impedir o cadastro de pessoas menores de 18 anos, considerando a data de registro da reserva.
 - 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:** RF03.

8. 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:** RF05, RF06 e RF18.

9. 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:** RF07.

10. 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:** RF08.

11. 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:** RF09.

12. 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:** RF15.

13. 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:** RF10.

14. 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:** RF11.

15. 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:** RF04.

16. 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:** RF12.

17. 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:** RF13.

18. 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:** RF13.

19. 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:** RF15.

20. 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:** RF16.

21. 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:** RF15.

22. 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:** RF17.

23. 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:** RF16.

24. 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:** RF16.

25. Early check-in e Late check-out:

- **Descrição:** Como proprietária, quero querer 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:** RF19.

26. 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 *early check-in* e *late check-out* aos preços das estadias.
 - O valor total da reserva deve refletir os acréscimos automaticamente.
- **Requisito Funcional Relacionado:** RF19.

27. 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:** RF20 e RF21.

28. 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:** RF20 e RF21.

29. 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:** RF22.

30. 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:** RF24.

31. 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:** RF23.

32. 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:** RF14.

33. 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:** RF26.

4.3 Arquitetura

A arquitetura do sistema desenvolvido para a Pousada Chalés Água de Coco foi planejada com base em princípios de eficiência, escalabilidade, modularidade e segurança, aspectos considerados fundamentais para sistemas de informação de uso corporativo ([PRES-SMAN, 2016; SOMMERVILLE, 2019](#)). O objetivo principal é garantir o funcionamento estável da plataforma de reservas, proporcionando aos gestores uma experiência fluida, confiável e segura, ao mesmo tempo em que facilita a manutenção e a evolução da solução no longo prazo.

A definição da arquitetura seguiu boas práticas de desenvolvimento de *software* orientado a serviços e apoiou-se em tecnologias consolidadas no mercado. A seguir, são descritos o desenho arquitetural, a organização das camadas da aplicação, o banco de dados utilizado e a infraestrutura de hospedagem.

4.3.1 Arquitetura Lógica

O sistema adota o *framework* *Django* ([DJANGO, 2024](#)), que segue o padrão de arquitetura MTV (*Model-Template-View*). Esse padrão promove a separação de responsabilidades, característica importante para manter a aplicação organizada, escalável e de fácil manutenção. Conforme defendem Bass, Clements e Kazman ([BASS; CLEMENTS; KAZMAN, 2012](#)), a modularidade arquitetural é um fator crítico para a evolução e a qualidade de sistemas de *software*.

4.3.1.1 Camada de Apresentação (Template)

Responsável pela interface com o usuário, esta camada é construída com HTML e CSS, priorizando responsividade e usabilidade ([DUCKETT, 2011](#)). Os dados exibidos ao usuário são carregados dinamicamente a partir da camada de controle, garantindo atualização em tempo real e interação consistente.

4.3.1.2 Camada de Controle (View)

Na camada de controle são implementadas as regras de negócio da aplicação. Essa camada processa as requisições recebidas, interage com os modelos de dados e retorna as informações organizadas para a interface de apresentação. Também é responsável pela autenticação, controle de acessos e segurança no fluxo de informações.

4.3.1.3 Camada de Modelo (Model)

A camada de modelo representa a estrutura de dados da aplicação. Utiliza o ORM (*Object-Relational Mapper*) do Django para abstrair as operações no banco relacional, reduzindo a complexidade da manipulação de dados e prevenindo falhas de segurança como *SQL Injection* ([POSTGRESQL, 2025](#)). Essa camada assegura integridade, consistência e eficiência na persistência de dados.

4.3.2 Banco de Dados

O banco de dados escolhido foi o *PostgreSQL*, considerado uma solução robusta, segura e compatível com aplicações de missão crítica. Sua adoção é justificada pelo suporte avançado a transações, confiabilidade e integração nativa com o Django ([POSTGRESQL, 2025](#)). Essa escolha também está alinhada às recomendações de Sommerville ([SOMMERVILLE, 2019](#)), que destaca a importância de bancos de dados maduros e estáveis em sistemas corporativos.

4.3.3 Infraestrutura em Nuvem

A aplicação está hospedada na *Amazon Web Services (AWS)*, que oferece recursos de alta disponibilidade, escalabilidade e mecanismos de segurança em conformidade com padrões internacionais ([AWS, 2025](#)). A configuração contempla:

- Suporte a conexões seguras via HTTPS, garantindo confidencialidade e integridade dos dados trafegados;
- Rotinas automáticas de *backup* para prevenir perda de informações;
- Monitoramento contínuo dos serviços, possibilitando respostas rápidas a falhas e incidentes.

O uso da computação em nuvem é reforçado por Laudon e Laudon ([LAUDON; LAUDON, 2020](#)), que destacam sua relevância para pequenos e médios negócios devido à flexibilidade e à redução de custos com infraestrutura física.

4.3.4 Preparação para Evoluções Futuras

Por fim, a arquitetura proposta foi estruturada para suportar futuras expansões, como integração com APIs externas (ex.: canais de venda e pagamentos online) e módulos avançados de relatórios gerenciais. De acordo com Turban, Pollard e Wood ([TURBAN; POLLARD; WOOD, 2018](#)), a capacidade de evolução é um atributo importante para a sustentabilidade de sistemas de informação em ambientes competitivos. Dessa forma, a solução desenvolvida não apenas atende às necessidades atuais da pousada, mas também abre caminho para a modernização contínua de seus processos administrativos.

4.3.5 Diagrama da Arquitetura

Para ilustrar visualmente a estrutura e o funcionamento da solução desenvolvida, foram elaborados diagramas de arquitetura com base nos padrões da UML (*Unified Modeling Language*). Esses diagramas auxiliam na compreensão das relações entre os componentes do sistema, da infraestrutura necessária para sua execução e da organização dos módulos internos. A seguir, são apresentados o diagrama de implantação, que detalha os elementos físicos envolvidos na execução da aplicação, e o diagrama de componentes, que representa a divisão lógica do sistema em módulos e suas interações.

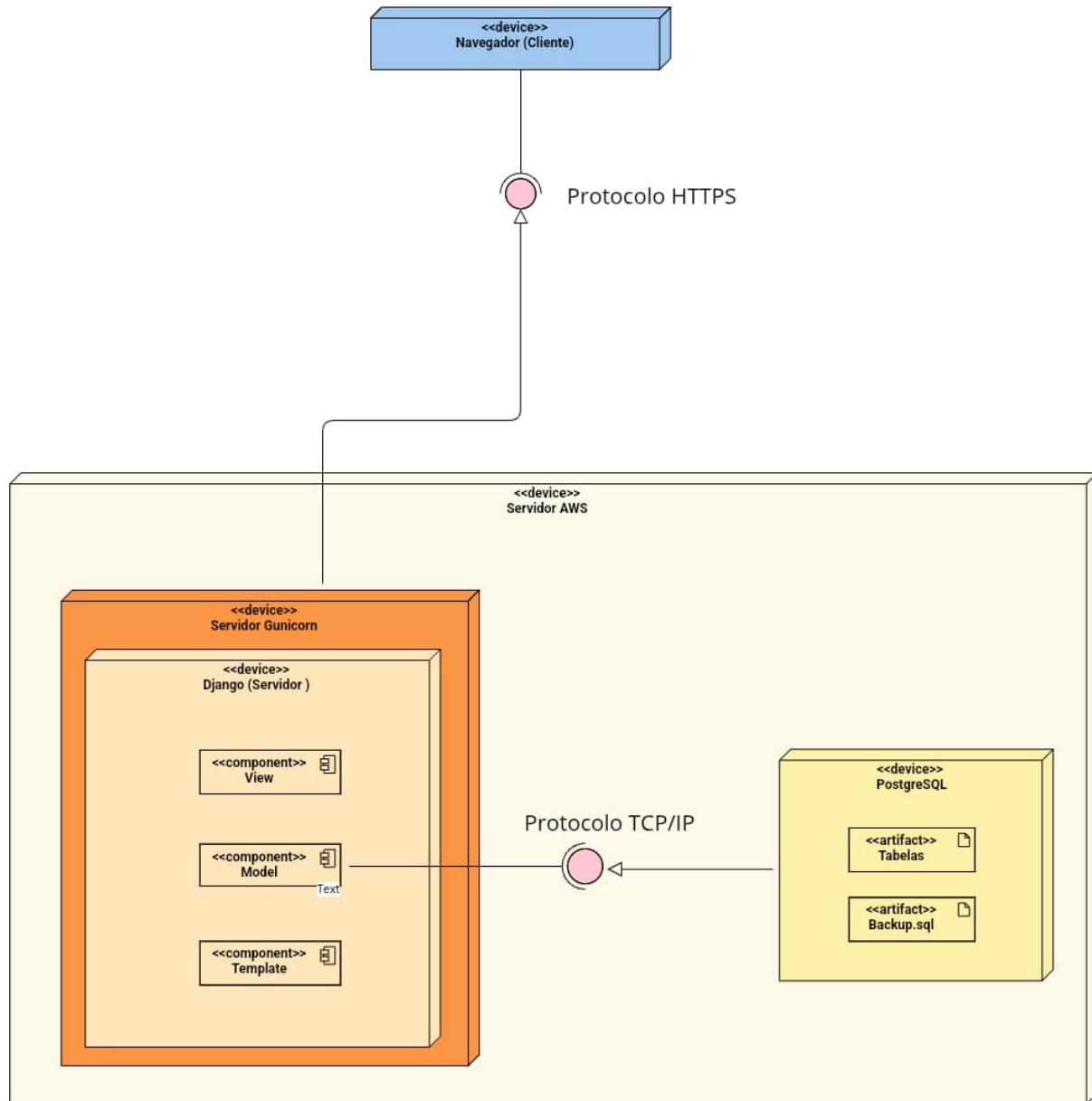
4.3.5.1 Diagrama de Implantação

A Figura 3 mostra o funcionamento da arquitetura do sistema.

Figura 3 – Diagrama de Implantação

Diagrama de Implantação

Made with
VisualParadigm
For non-commercial use

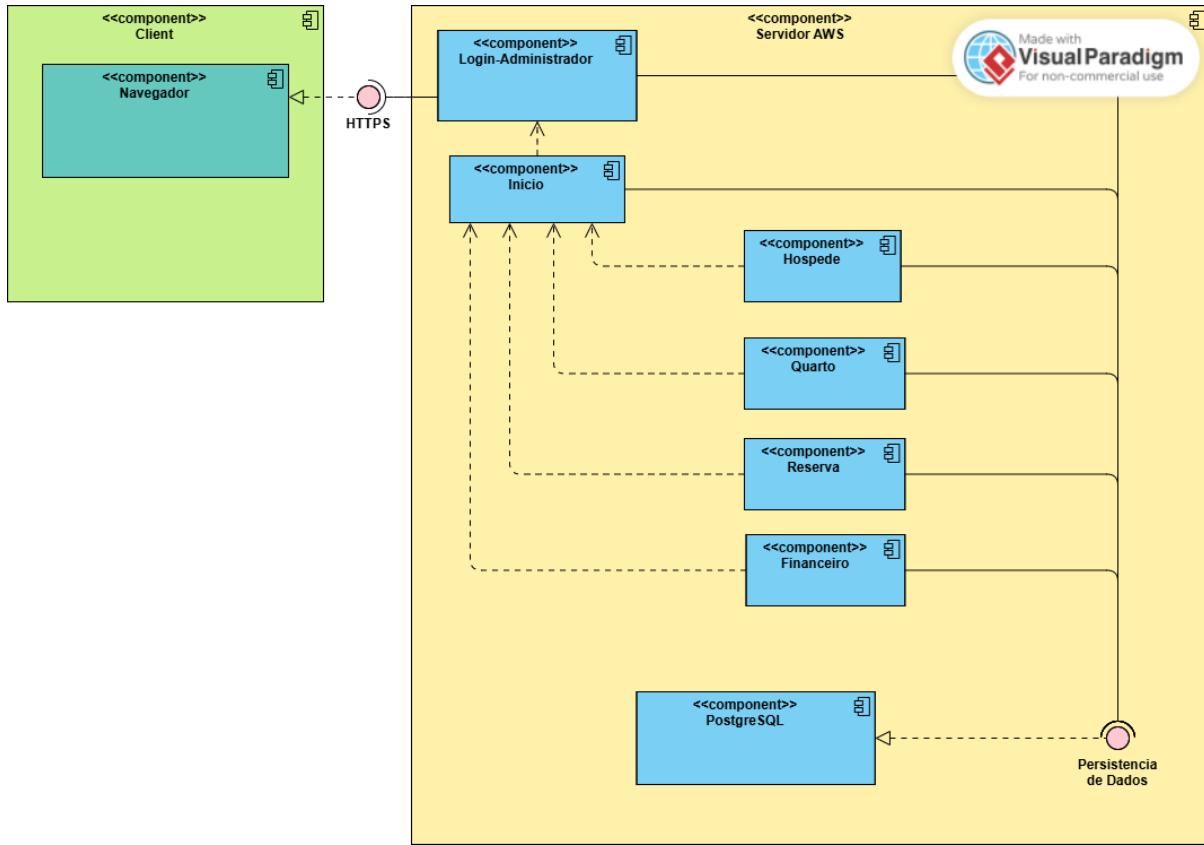


Fonte: Desenvolvido pelos autores no Online Visual-Paradigm.

4.3.5.2 Diagrama de Componentes

A Figura 4 mostra o funcionamento da arquitetura do sistema.

Figura 4 – Diagrama de Componentes



Fonte: Desenvolvido pelos autores no Online Visual-Paradigm.

4.4 Tecnologias

O desenvolvimento do sistema da Pousada Chalés Água de Coco foi pautado na utilização de tecnologias modernas, acessíveis e amplamente consolidadas no mercado. As escolhas realizadas tiveram como objetivo garantir robustez, escalabilidade, segurança, custo-benefício e facilidade de manutenção, assegurando a aderência às boas práticas de desenvolvimento *web*. Nesta seção, são apresentadas as principais ferramentas e *frameworks* empregados, destacando-se o *framework* Django, o sistema gerenciador de banco de dados PostgreSQL e a infraestrutura de nuvem provida pela Amazon Web Services (AWS).

4.4.1 Django

O Django é um *framework web* de alto nível baseado em Python, projetado para acelerar o desenvolvimento e incentivar práticas de código limpo e seguro (DJANGO, 2024; PYTHON, 2025). Sua principal vantagem é a disponibilização de diversos recursos prontos para uso, como sistema de autenticação, ORM (Object-Relational Mapper), administração nativa e proteção contra vulnerabilidades comuns (*SQL Injection*, *Cross-Site Scripting* e *Cross-Site Request Forgery*), reduzindo significativamente o tempo de desenvolvimento e

aumentando a segurança da aplicação.

4.4.1.1 Front-end

No front-end, a aplicação utiliza o padrão MTV (*Model-Template-View*), em que os *templates* HTML, associados às *views*, geram páginas dinâmicas e responsivas. A interface foi estruturada com HTML e CSS, possibilitando usabilidade e responsividade em diferentes dispositivos (DUCKETT, 2011). Esse modelo favorece a manutenção e a escalabilidade, permitindo a inclusão de novas funcionalidades, como calendário de disponibilidade, relatórios personalizados e notificações automatizadas.

4.4.1.2 Back-end

O back-end concentra as regras de negócio e a lógica da aplicação. As *views* do Django processam as requisições, acessam o banco de dados via ORM e retornam as informações aos *templates*. Além disso, o Django disponibiliza controle granular de permissões e autenticação, fundamentais para proteger informações sensíveis de hóspedes e reservas.

4.4.2 Banco de Dados

O banco de dados utilizado é o PostgreSQL, um dos sistemas gerenciadores de banco de dados relacionais (SGBDR) mais robustos e confiáveis atualmente (POSTGRESQL, 2025). Entre seus diferenciais técnicos estão: suporte ao padrão ACID, tipos de dados avançados como JSONB, replicação nativa, *triggers*, *stored procedures* e mecanismos de índices sofisticados. Sua natureza de código aberto garante liberdade de modificação, ausência de custos de licenciamento e ampla compatibilidade com diferentes linguagens e *frameworks*, sendo uma opção altamente escalável para sistemas de pequeno a grande porte.

4.4.3 Infraestrutura em Nuvem com AWS

A hospedagem do sistema foi estruturada na Amazon Web Services (AWS), provedora líder de serviços em nuvem. A escolha pela AWS fundamenta-se em sua capacidade de oferecer alta disponibilidade, escalabilidade e segurança, alinhadas ao modelo de computação em nuvem definido pelo NIST (National Institute of Standards and Technology), que recomenda elasticidade, autosserviço sob demanda, medição de uso e ampla conectividade (MELL; GRANCE, 2011; AWS, 2025).

Entre os serviços utilizados destacam-se:

- **Amazon RDS:** gerenciamento simplificado do banco de dados relacional PostgreSQL;
- **Amazon S3:** armazenamento seguro e escalável de arquivos e mídias;
- **Amazon CloudWatch:** monitoramento e geração de métricas em tempo real.

Essa infraestrutura garante redundância, políticas de *backup* e suporte a conexões seguras via HTTPS, fatores indispensáveis para um sistema que manipula dados pessoais e financeiros.

4.4.4 Boas Práticas de Desenvolvimento

Além da escolha tecnológica, adotaram-se boas práticas de engenharia de *software*, como versionamento de código com Git, integração e entregas contínuas (CI/CD), aplicação de testes automatizados e atualizações periódicas de dependências. Tais práticas favorecem a manutenibilidade, reduzem riscos de falhas em produção e facilitam a evolução da aplicação, assegurando qualidade e estabilidade ao longo de seu ciclo de vida.

4.4.5 Resumo da Escolha Tecnológica

A adoção do Django, PostgreSQL e AWS foi motivada pela combinação de fatores como robustez, segurança, escalabilidade e baixo custo de implementação. Essas tecnologias, em conjunto com boas práticas de desenvolvimento, formam a base de uma solução moderna e confiável, alinhada às necessidades da Pousada Chalés Água de Coco e coerente com os objetivos do projeto.

4.5 Estilo de Código

O desenvolvimento do sistema de gestão para a Pousada Chalés Água de Coco foi orientado pela adoção de um estilo de codificação consistente e pelo uso de padrões amplamente reconhecidos na comunidade de desenvolvimento Python e Django. Essa abordagem garante legibilidade, manutenibilidade e escalabilidade do código ao longo de todo o ciclo de vida do sistema ([ROSSUM, 2013](#); [DJANGO, 2024](#)).

4.5.1 Estrutura e Organização do Projeto

O projeto segue a arquitetura padrão do *framework* Django, caracterizada pela clara separação de responsabilidades (*Separation of Concerns*). A estrutura é modular, sendo as funcionalidades agrupadas em aplicações (*apps*) específicas, como *core* (responsável por hóspedes, quartos e reservas) e *financeiro* (voltada ao controle de títulos, categorias e

bancos). Dentro de cada *app*, o código é organizado conforme sua finalidade: `models.py` define a estrutura de dados e relacionamentos; `views.py` contém a lógica de negócios e o controle das requisições; `forms.py` é responsável pela validação e renderização de formulários; `urls.py` gerencia o roteamento de páginas; e o diretório `templates/` abriga os arquivos HTML da camada de apresentação. Além disso, foi criada uma pasta `utils/` para funções auxiliares reutilizáveis (como validação de datas e envio de e-mails), seguindo o princípio DRY (*Don't Repeat Yourself*) (DJANGO, 2024). A configuração central do projeto reside nos arquivos `settings.py` e `urls.py` localizados na pasta principal `setup/`, em conformidade com as convenções do Django.

4.5.2 Convenções de Nomenclatura

As convenções de nomenclatura seguem as diretrizes da PEP 8 (ROSSUM, 2013), garantindo padronização e clareza:

- **Classes (Models, Forms, Views baseadas em classes):** utilizam o padrão *CamelCase* (ex.: `Hospede`, `ReservaForm`, `CategoriaDespesaListView`);
- **Funções, métodos e variáveis:** seguem o padrão *snake_case* (ex.: `listar_hospedes`, `form_valid`, `data_reserva_inicio`);
- **Constantes:** utilizam letras maiúsculas e o padrão *UPPER_CASE_SNAKE_CASE* (ex.: `STATUS_CHOICES`, `DJANGO_SETTINGS_MODULE`);
- **Arquivos e módulos:** seguem o padrão *snake_case* (ex.: `test_models.py`, `conflito_datas.py`);
- **Templates:** utilizam nomes descritivos em *snake_case* ou *kebab-case* (ex.: `listar.html`, `form.html`, `list-check-in.html`).

4.5.3 Padrões de Código e Legibilidade

O código demonstra boa legibilidade, utilizando identação consistente, nomes descritivos e separação lógica de blocos. Comentários são empregados para explicar trechos mais complexos, tanto no back-end, quanto no front-end. As importações seguem boas práticas: primeiro as bibliotecas padrão do Python, depois as dependências externas e, por fim, os módulos locais, conforme as recomendações da PEP 8.

4.5.4 Considerações Finais sobre o Estilo

Em resumo, o projeto apresenta uma codificação consistente, modular e alinhada às boas práticas da comunidade Django e Python. A separação em *apps*, a organização interna dos módulos e o uso de recursos nativos do *framework* favorecem a manutenibilidade e a evolução do sistema.

4.6 Ferramentas de Apoio

Durante o desenvolvimento do sistema da Pousada Chalés Água de Coco, foram empregadas diversas ferramentas que auxiliaram em diferentes etapas do projeto, desde a modelagem de dados e a construção da arquitetura até o controle de versões, documentação e comunicação entre os membros da equipe. A escolha dessas ferramentas foi guiada por critérios como acessibilidade, confiabilidade, funcionalidades oferecidas e integração com as tecnologias adotadas. A seguir, são descritas as principais ferramentas utilizadas e suas contribuições para o sucesso do projeto.

4.6.1 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](#)).

4.6.2 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.

4.6.3 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.

4.6.4 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](#)).

4.6.5 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.

4.6.6 ProjectLibre

Para monitorar o progresso do projeto, foi utilizada a ferramenta ProjectLibre. Esta é um software *open source* de gerenciamento de projetos ([PROJECTLIBRE, 2025](#)). A ferramenta possibilitou uma visualização clara e dinâmica das estruturas das *sprints backlog*, o que contribuiu para que a equipe colocasse em prática os processos de identificação de problemas e atrasos, por meio de inspeções frequentes e da manutenção da transparência, facilidade pela geração automática do Modelo de Gantt— representa visualmente um projeto.

Na ferramenta, as tarefas do *product backlog* foram organizadas com datas de início e término, respeitando os prazos definidos para a conclusão das *sprints*. Além disso, os recursos, tanto humanos quanto de infraestrutura, foram devidamente associados a cada tarefa, o que possibilitou o monitoramento e o controle eficaz dos custos do projeto.

4.7 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.

4.8 Segurança, Privacidade e Legislação

A segurança da informação e a conformidade com legislações específicas representam pilares fundamentais no desenvolvimento de sistemas *web* que envolvem dados pessoais e financeiros. No caso do sistema de reservas da Pousada Chalés Água de Coco, essas diretrizes foram centrais na concepção e implementação da aplicação. Para tanto, foram aplicadas boas práticas de desenvolvimento seguro, funcionalidades nativas do *framework* Django e medidas alinhadas à Lei Geral de Proteção de Dados (LGPD – Lei nº 13.709/2018) ([BRASIL, 2018](#)).

4.8.1 Segurança da Aplicação

O *framework* Django oferece mecanismos nativos de defesa contra vulnerabilidades comuns em aplicações *web*, como injeção de SQL, execução remota de código, *Cross-Site Scripting* (XSS) e falsificação de requisições entre sites (CSRF) ([DJANGO, 2024](#)). Além das proteções automáticas, foram aplicadas práticas adicionais, descritas a seguir:

- **Autenticação e autorização:** O controle de acesso é baseado em permissões, restringindo páginas sensíveis apenas a usuários autenticados e devidamente autorizados.
- **Proteção contra CSRF:** Todas as requisições POST utilizam *tokens* exclusivos para prevenir ações maliciosas.
- **Escapamento automático de HTML (XSS):** O sistema de *templates* do Django neutraliza a execução de *scripts* não autorizados.
- **Armazenamento seguro de senhas:** Senhas são armazenadas utilizando algoritmos modernos de *hash*, como *PBKDF2*, reduzindo significativamente os riscos em caso de vazamento ([OWASP Foundation, 2021](#)).
- **Gerenciamento de sessões:** Utilização de identificadores criptografados, expiração automática de sessões inativas e proteção contra falsificação.

Essas medidas seguem recomendações amplamente adotadas pela comunidade de segurança, como as diretrizes da *Open Web Application Security Project* (OWASP), voltadas à mitigação das principais ameaças em sistemas *web* (OWASP Foundation, 2021).

4.8.2 Segurança na Comunicação

Todas as comunicações entre cliente e servidor ocorrem por meio do protocolo HTTPS, utilizando certificado SSL/TLS emitido por uma autoridade certificadora confiável, instalado e configurado no ambiente AWS (AWS, 2025). O uso de criptografia garante:

- **Confidencialidade:** Proteção dos dados contra interceptação não autorizada.
- **Integridade:** Garantia de que as informações não foram alteradas durante a transmissão.
- **Autenticidade:** Validação da identidade do servidor pela autoridade certificadora.

Adicionalmente, todas as requisições HTTP são redirecionadas automaticamente para HTTPS. Foram configurados cabeçalhos de segurança (como `Strict-Transport-Security` e `X-Content-Type-Options`), reforçando a proteção contra ataques de *downgrade* e manipulação de conteúdo (RESCORLA, 2018). Os *headers* de segurança do sistema foram testados através da plataforma Security Headers (HELME, 2025). O resultado foi uma nota A+, disponível na [seção F.2](#).

Para verificar a segurança da implementação, a comunicação HTTPS do sistema *web* da Pousada Chalés de Água de Coco também foi avaliada através da plataforma SSL Labs, por meio do SSL Server Test, que analisa detalhadamente a implementação do protocolo TLS/SSL pelo servidor e a proteção das conexões estabelecidas (Qualys SSL Labs, 2025). O certificado utilizado foi emitido pela autoridade certificadora gratuita Let's Encrypt (Let's Encrypt, 2025). A aplicação recebeu a nota A, indicando uma configuração segura e adequada, e a captura de tela do resultado está disponível no [Apêndice F](#).

4.8.3 Conformidade com a LGPD

A LGPD estabelece princípios fundamentais para o tratamento de dados pessoais, como a necessidade de consentimento, a finalidade específica do uso e a transparência para o titular. Em conformidade com essa legislação, o sistema adota os seguintes mecanismos:

- **Minimização de dados:** Coleta apenas informações estritamente necessárias para a operação das reservas.

- **Gestão de consentimento:** Usuários podem alterar o consentimento previamente fornecido, em conformidade com os princípios de liberdade e autonomia.
- **Direitos do titular:** Possibilidade de solicitação de acesso, correção ou exclusão de dados pessoais armazenados.
- **Transparéncia:** Informações sobre a finalidade do tratamento de dados estão disponíveis de forma clara.
- **Armazenamento seguro e backups:** Dados são protegidos por mecanismos de criptografia e submetidos a rotinas de *backup* regular em ambiente de nuvem (AWS), garantindo resiliência e prevenção contra perdas.

A adoção desses recursos reforça a responsabilidade do sistema frente à segurança, privacidade e conformidade, estabelecendo uma base sólida para a proteção das informações dos hóspedes e para a credibilidade do serviço.

4.9 Modelagem do Banco de Dados

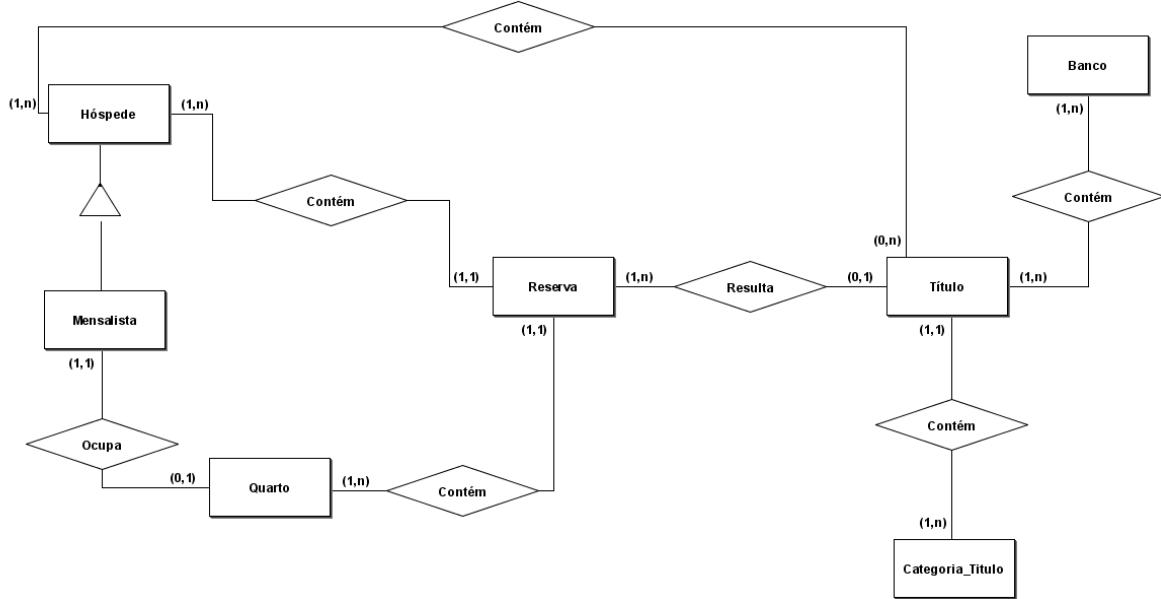
A modelagem do banco de dados é uma etapa fundamental no desenvolvimento de sistemas de informação, pois define a estrutura lógica e relacional para o armazenamento e manipulação dos dados. No sistema da Pousada Chalés Água de Coco, a modelagem foi realizada com foco na integridade dos dados, normalização e clareza nos relacionamentos entre as entidades.

Utilizando a ferramenta BRModelo, foram construídos o Modelo Entidade Relacionamento (MER) e o Diagrama Entidade Relacionamento (DER), os quais serviram como base para a implementação do banco de dados relacional no PostgreSQL. Esses diagramas ajudam a visualizar as entidades principais do sistema, como hóspedes, reservas, quartos, bem como os vínculos entre elas, garantindo coerência e consistência no projeto de dados.

4.9.1 Modelo Entidade-Relacionamento - MER

A Figura 5 mostra o Modelo Entidade-Relacionamento (MER) do sistema.

Figura 5 – Modelo Entidade-Relacionamento (MER)

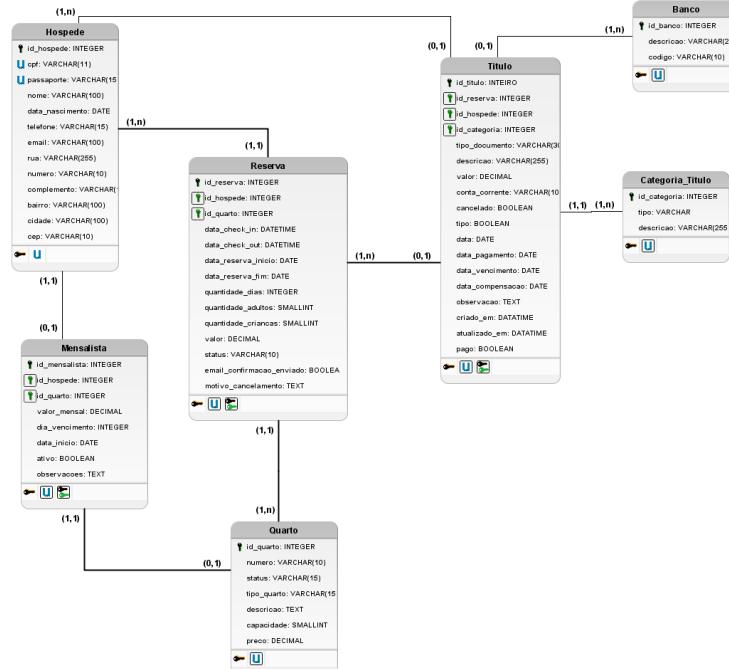


Fonte: Desenvolvido pelos autores no brModelo.

4.9.2 Diagrama Entidade-Relacionamento - DER

A Figura 6 mostra o Diagrama Entidade-Relacionamento (DER) do sistema.

Figura 6 – Diagrama Entidade-Relacionamento (DER)



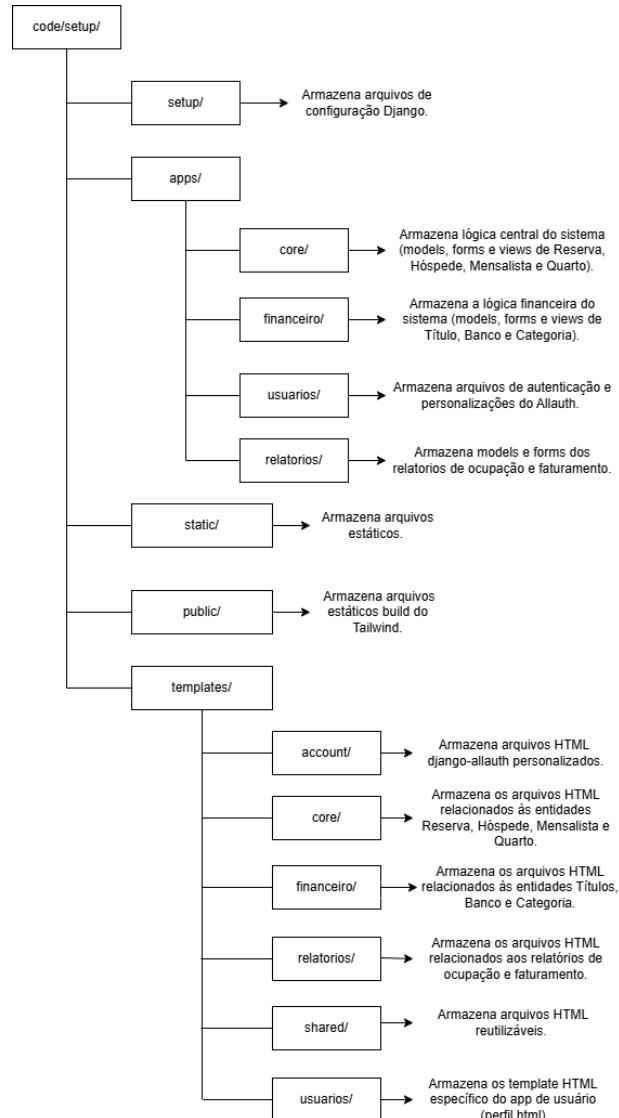
Fonte: Desenvolvido pelos autores no brModelo.

4.10 Repositório da Aplicação

Nesta seção é apresentada a estrutura do ambiente de desenvolvimento do sistema *web* de gestão da Pousada Chalés Água de Coco. Visando facilitar o gerenciamento, a colaboração entre os desenvolvedores e o controle do versionamento do código-fonte do sistema de gestão, a estrutura foi armazenada em um repositório remoto no GitHub (Figura 8).

O código-fonte do sistema foi estruturado em aplicações modulares, arquivos de configuração geral do ambiente Django e arquivos auxiliares, seguindo o padrão recomendado pelo *framework*. Essa organização, que visa facilitar o desenvolvimento, a manutenção e a segurança do sistema, está estruturada como ilustrado na Figura 7

Figura 7 – Exemplificação da Estrutura do Re却itório



Fonte: Elaboração própria.

Essa configuração segue as boas práticas do *framework*, promovendo baixo acoplamento e alta coesão. Na camada de lógica, essa coesão foi aplicada especificamente no *app core*, que organiza os arquivos *models* e *views* das entidades quarto, hóspede e reserva, as quais formam o domínio central do negócio: a gestão de reservas. Os demais *apps* representam preocupações separadas: usuários (autenticação), financeiro (contabilidade) e relatórios (camada de leitura e agregação). E, por isso, seguiram uma lógica de baixo acoplamento.

No entanto, para a camada de apresentação (*templates*), optou-se por centralizar todos os arquivos em um único diretório (`code/setup/templates/`). Essa escolha, apesar de reduzir a portabilidade individual de cada *app*, aumenta a coesão do projeto e facilita a manutenção e a organização das interfaces, simplificando o fluxo de trabalho.

Figura 8 – QR Code repositório do sistema no GitHub



Fonte: Elaboração própria.

4.11 Entregas

O desenvolvimento do sistema seguiu um cronograma baseado em entregas parciais, cada uma representando uma etapa importante na evolução do projeto. Essas entregas permitiram o acompanhamento contínuo do progresso, validação das funcionalidades e documentação formal de todas as fases do trabalho. A seguir, são descritas as principais entregas realizadas ao longo do semestre, com seus respectivos objetivos e resultados.

4.11.1 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.

4.11.2 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

4.11.3 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.

4.11.4 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.

4.11.5 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 Plano de Testes

Neste tópico é detalhado o planejamento e execução dos testes do sistema *web* de gestão da pousada.

5.1 Objetivo

O objetivo do plano de testes é validar a funcionalidade, confiabilidade e robustez do sistema, com foco principal em:

- **Autenticação e Controle de Acesso:** Garantir a robustez do ciclo de vida do usuário.
- **Confiabilidade dos Relatórios:** Assegurar a exatidão dos cálculos e totais apresentados.
- **Experiência do Usuário:** Garantir a eficiência da interface do sistema na entrega de *feedbacks* imediatos ao usuário e na navegação em grandes volumes de dados.
- **Integridade dos Dados:** Validar o ciclo de vida completo da entidade "Hóspede", a fim de garantir a aplicação correta das regras de validação no cadastro e a proteção da integridade referencial do sistema.
- **Lógica de Negócio de Reservas:** Validar o ciclo de vida completo da entidade "Reserva", desde a criação ao cancelamento, garantindo a concordância às regras financeiras e de disponibilidade.
- **Qualidade do Código-Fonte:** Identificar e mitigar preventivamente *bugs* e vulnerabilidades de segurança através da análise estática.

5.2 Escopo do Teste

Para uma cobertura abrangente do código, esse plano de testes combinou as seguintes categorias de testes:

- **Análise Estática:** Para verificação de *bugs* e vulnerabilidades.
- **Testes Unitários:** Testar cobertura das principais funções e lógicas de negócio.
- **Testes Funcionais:** Testar a experiência do usuário no navegador.

Em contrapartida, esse plano não contempla as seguintes categorias de testes:

- **Testes de Carga e Estresse:** O sistema é um *back-office*, com um número muito limitado de usuários simultâneos (cenário de uso primário de uma única usuária). Devido a este perfil de baixa concorrência, os riscos técnicos associados a gargalos de performance são considerados muito baixos. Nesse sentido, entendeu-se que o investimento de esforço para testes de carga seria desproporcional ao risco, sendo mais valioso priorizar os testes funcionais, de integridade e de segurança.
- **Testes de Acessibilidade:** O escopo do projeto foi definido com base no perfil de usuário primário (a proprietária), e este perfil não gerou requisitos formais de acessibilidade. Por esse motivo, não foram adicionados testes de validação correspondentes.

5.3 Análise Estática de Qualidade de Código

Para a realização da análise estática do código foi utilizada a ferramenta SonarQube. O objetivo foi identificar e corrigir *code smells*, *bugs* em potencial e vulnerabilidades de segurança antes do *deploy*.

- Durante a execução do SonarQube foram avaliados e corrigidos os seguintes casos:
 - **Falhas:**
 - * **Detecção de Recursão:** Sinalização de *RecursionError*.
 - * **Detecção de AttributeError:** Sinalização de falhas de importação.
 - **Vulnerabilidades:**
 - * **SQL Injection:** Verificação de uso de SQL puro de forma insegura.
 - **Qualidade do Código:**
 - * **Complexidade Cognitiva:** Sinalização de funções muito complexas que podem dificultar manutenções futuras.
 - * **Código Duplicado:** Identificação de lógicas repetidas de forma desnecessária.
 - * **Variáveis e Imports Não Utilizadas:** Sinalização da existência de variáveis e imports não utilizadas.

5.4 Testes Funcionais

Os testes funcionais deveriam testar as seguintes categorias: autenticação, segurança, quartos, hóspedes, tarifas, reservas, financeiro, integridade, relatórios, UI/UX, Backup/Restore, Performance e Fluxo E2E.

Para cada caso de teste foi definida uma pré-condição, os passos que devem seguidos durante o teste e o resultado esperado, como exemplificado no [Quadro 10](#).

Quadro 10 – Testes Funcionais: Categoria Quartos

ID	Descrição	RF	Pré-condição	Ação	Resultado Esperado
CT-Q01	Cadastro de quarto válido	RF13	Nenhum	Informar número, tipo, capacidade, preço → Salvar	Quarto cadastrado.
CT-Q02	Número duplicado	RF13	Quarto nº 101 cadastrado	Cadastrar nº 101 novamente	Erro de duplicidade.
CT-Q03	Preço inválido (negativo ou zero)	RF13	Nenhum	Informar preço inválido → Salvar	Erro de validação.
CT-Q04	Exclusão de quarto com reserva ativa	RF13	Quarto com reserva ativa	Tentar excluir	Exclusão bloqueada; mensagem informativa exibida.
CT-Q05	Ver disponibilidade por período	RF14	Quartos e reservas cadastrados	Selecionar intervalo → Buscar	Somente quartos disponíveis exibidos.

Fonte: Elaborado pelos autores.

O quadro completo com todos os testes funcionais está disponível no repositório do GitHub ([Figura 9](#)). No total foram planejados 66 testes funcionais que cobriam, diretamente, o equivalente a 80% dos requisitos funcionais do sistema (relacionados a 20 dos requisitos funcionais totais). No entanto, foi efetivamente realizado, cerca de 38% dos testes, devido atrasos no desenvolvimento de novas funcionalidades. Os testes realizados estão documentados do [Apêndice E](#) ao [Apêndice J](#).

Figura 9 – QR Code Documento de Testes Funcionais



Fonte: Elaboração própria.

5.5 Testes Unitários

Os testes unitários foram a primeira linha de defesa da qualidade do código e auxiliaram na validação da lógica interna da aplicação. Para o desenvolvimento, utilizou-se o *framework* Pytest. Seguindo a convenção padrão do Django, os testes foram organizados em pastas `tests/` localizadas dentro de cada *app* (módulo) da aplicação. Esta estrutura permitiu a criação de testes limpos, modulares e eficientes.

Como estratégia, o foco não foi desenvolver testes unitários capazes de atingir 100% de cobertura de linhas , mas sim em garantir a robustez das áreas de maior complexidade e risco de negócio. O esforço foi concentrado em:

- **Lógica de Cálculo:** Funções puras responsáveis pela geração de relatórios e KPIs (ex: balanço financeiro).
- **Funções de Validação:** A lógica complexa de validação nos `forms.py` e `models.py` (ex: regras de conflito de reserva).
- **Funções Utilitárias:** Componentes críticos de `utils.py` (ex: exportação de arquivos).

O código padrão gerado pelo Django (como o Admin, migrações, ou funções de `views.py` que apenas renderizam *templates*) foi intencionalmente considerado de baixo risco e não foi o foco principal desta camada de testes, sendo validado pelos Testes Funcionais.

A cobertura de testes foi monitorada através da integração com o SonarQube e apresentou um resultado final de, aproximadamente, 67% das linhas de código. Esta métrica é um reflexo direto da estratégia adotada: ela representa uma cobertura muito alta sobre a área de negócio complexa do sistema, principalmente das linhas de gestão de reservas, ao mesmo tempo que omite os códigos de baixo risco.

6 Viabilidade Financeira

Este capítulo apresenta o estudo de viabilidade financeira do sistema de gestão desenvolvido, contemplando custos de infraestrutura, mão de obra e manutenção. A análise considera a natureza de extensão acadêmica do projeto, caracterizado pelo trabalho de desenvolvimento voluntário, em regime *home office* e uso de equipamentos e infraestrutura próprios da equipe, apresentando os custos efetivos. Adicionalmente, também, apresenta projeções para um contexto comercial, incluindo custos indiretos. Essa abordagem tem como objetivo avaliar a factibilidade do projeto tanto para a aplicação imediata na pousada quanto para futuras possíveis expansões ou adaptações do sistema.

6.1 Custos

O levantamento dos custos relacionados ao projeto considerou os dois contextos em que ele se aplica: como extensão acadêmica e a projeção para ambiente comercial. Dentro do período de 9 meses de desenvolvimento (correspondente ao tempo total estimado de projeto). E, adotando os seguintes critérios: cotações oficiais de serviços e *softwares* disponíveis publicamente em seus sites institucionais; valores de mercado obtidos através de pesquisas salariais em plataformas especializadas; estimativas de consumo de energia elétrica baseadas em especificações técnicas dos equipamentos e tarifas locais; e cálculos de depreciação fundamentados na vida útil padrão dos equipamentos tecnológicos utilizados.

6.1.1 Custo Estrutural

Os custos estruturais englobam os ativos e serviços necessários para o ciclo de vida completo do sistema, desde o seu desenvolvimento até a sua implantação. O que inclui os equipamentos (*hardware*) utilizados pela equipe até os serviços de hospedagem, domínios e licenças de *software* (serviços e infraestrutura).

Para análise desses custos, foram construídas a [Tabela 1](#) e a [Tabela 2](#). Ambas as tabelas adotam uma abordagem de duplo contexto: o Custo Efetivo, que reflete os valores atuais como projeto acadêmico, e o Custo Estimado, que projeta os valores de mercado para uma eventual implementação comercial.

Tabela 1 – Resumo dos custos da infraestrutura de desenvolvimento

Item	Tipo	Qtd.	Custo Efetivo	Custo Estimado
Notebooks	Equipamento	5	R\$ 0,00	R\$ 17.500,00/único
Depreciação Notebooks	Custo indireto	5	R\$ 0,00	R\$ 291,65/ mês
Roteadores	Equipamento	5	R\$ 0,00	R\$ 750,00/único
Internet	Infraestrutura	5	R\$ 0,00	R\$ 574,85/ mês
Eletricidade	Infraestrutura	5	R\$ 0,00	R\$ 17,20/mês
Github	Plataforma de Desenvolvimento	1	R\$ 0,00	R\$ 0,00
Total (9 meses)	–	26	R\$ 0,00	R\$ 19.133,70

Fonte: Elaborado pelos autores

A Tabela 1 elucida os custos relacionados ao ambiente de trabalho da equipe. As estimativas dos custos comerciais dos equipamentos de *hardware* consideraram valores médios do mercado de cada unidade, avaliados através de uma pesquisa em grandes sites de vendas. Os valores mensais de infraestrutura (eletricidade e internet) foram estimados através do custo médio local de tarifas de energia (consumo por kWh) e planos mensais de internet (ANATEL, 2025; ENEL, 2025). Enquanto, o cálculo de depreciação linear dos 5 notebooks considerou vida útil de 5 anos com uma taxa de 20% ao ano para cada (Receita Federal do Brasil, 2017).

Tabela 2 – Resumo dos custos da infraestrutura de implantação e software

Item	Tipo	Custo Efetivo	Custo Estimado
AWS	Hospedagem em Nuvem	R\$ 0,00	R\$ 56,80/mês
Certificado SSL	Certificação	R\$ 0,00	R\$ 0,00
Nginx	Proxy Reverso	R\$ 0,00	R\$ 0,00
Gunicorn	Servidor de Aplicação	R\$ 0,00	R\$ 0,00
Django	Framework Web	R\$ 0,00	R\$ 0,00
PostgreSQL	Banco de Dados	R\$ 0,00	R\$ 0,00
Total (9 meses)	–	R\$ 0,00	R\$ 511,20

Fonte: Elaborado pelos autores

A Tabela 2 detalha os componentes de software e serviços em nuvem para a operação do sistema. No contexto acadêmico, o custo efetivo foi zerado pela utilização do nível gratuito (*Free Tier*) da AWS, que possui duração de 12 meses para serviços essenciais.

Para o cenário comercial, o Custo Estimado é composto essencialmente pelo serviço de hospedagem em nuvem, em específico do servidor EC2 com serviços adicionais de armazenamento e de envio de *e-mail*, projetado para um custo mensal de aproximadamente R\$ 56,80. Enquanto, a estratégia de negócio se apoia no uso de tecnologias de código aberto (Nginx, Gunicorn, Django, PostgreSQL) e certificados de segurança gratuitos da Let's Encrypt, que mantêm os custos de licenciamento zerados em ambos os cenários.

A análise comparativa das tabelas evidenciam a viabilidade do projeto nos dois cenários. No contexto acadêmico, o custo estrutural direto foi nulo (R\$ 0,00), viabilizado pela utilização de recursos preexistentes da equipe, planos gratuitos de serviços e *softwares* de código aberto.

Em contrapartida, uma transição para um modelo comercial exigiria um planejamento financeiro robusto. A operação demandaria um investimento inicial de R\$ 18.250,00 para aquisição de equipamentos. Adicionalmente, haveria custos operacionais mensais recorrentes estimados em R\$ 940,50 para a manutenção do sistema. Totalizando em um investimento de R\$ 26.572,30.

6.1.2 Custo de Pessoal

A análise do custo de pessoal visa quantificar o valor de mercado da mão de obra investida no desenvolvimento e implantação do sistema. Nesse sentido, o levantamento foca na projeção para um cenário comercial, uma vez que a execução do projeto no âmbito acadêmico se deu de forma voluntária, representando um custo efetivo nulo para a poussada. Para realizar tal projeção, a metodologia adotada baseou-se em dados salariais consultados em plataformas especializadas, a fim de estimar os custos de mão de obra em um ambiente de mercado ([Glassdoor, 2025](#); [Indeed Brasil, 2025](#)).

A estrutura de cálculo para o custo de pessoal é apresentada em duas fases. Primeiramente, o esforço de mão de obra foi dimensionado na [Tabela 3](#), que especifica os parâmetros de trabalho, como as funções da equipe, a quantidade de profissionais por cargo e a dedicação em horas diárias e dias por mês. Em seguida, a [Tabela 4](#) apresenta o valor da hora técnica para cada função e, a partir do cruzamento desses dados, consolida as estimativas de custo mensal, bem como as projeções de custo total.

Tabela 3 – Quantidade e horas trabalhadas por função

Função	Quantidade	Horas/Dia	Dias/Mês	Total de Horas/Mês
Analista de Testes	1	3	22	66
Desenvolvedor Back-End	1	3	22	66
Desenvolvedor Front-End	1	3	22	66
Engenheiro de Software	1	3	22	66
Gerente de Projeto (PM)	1	3	22	66

Fonte: Elaborado pelos autores.

Tabela 4 – Custos por função

Função	Custo Hora (R\$)	Custo Mensal (R\$)	Total 4 meses (R\$)	Total 9 meses (R\$)
Analista de Testes	13,63	899,58	3.598,32	8.096,22
Desenvolvedor Back-End	23,50	1.551,00	6.204,00	13.959,00
Desenvolvedor Front-End	22,35	1.475,10	5.900,40	13.275,90
Engenheiro de Software	26,13	1.724,58	6.898,32	15.521,22
Gerente de Projeto (PM)	17,42	1.149,72	4.598,88	10.347,48
Total Mão de Obra	103,03	6.799,98	27.199,92	61.199,82

Fonte: Elaborado pelos autores.

A projeção dos custos de pessoal foi dividida em dois intervalos temporais: 4 e 9 meses. O período de quatro meses corresponde ao marco de entrega da primeira versão funcional do sistema (MVP), enquanto o de nove meses abrange o ciclo completo de planejamento, desenvolvimento e implantação do projeto.

As condições para os cálculos consideram que os cargos listados refletem as funções essenciais desempenhadas pela equipe, com valores salariais de referência correspondentes ao nível júnior de mercado. Adicionalmente, a carga horária e os dias de trabalho especificados representam o tempo efetivamente investido pela equipe na execução do projeto. Dessa forma, a projeção para um cenário de implementação comercial indica que o investimento em mão de obra para a pousada parceira seria de R\$ 27.199,92 no primeiro quadrimestre e um total de R\$ 61.199,82 para a execução e conclusão completa do projeto em nove meses.

6.1.3 Custo Total

O custo total do projeto representa a somatória dos custos estruturais e de pessoal para a realização do projeto. A despesa integral recorrente está demonstrada na Tabela 5 que une os custos por mão de obra e estrutura em 3 períodos distintos:

- Mensal: valor estimado para um único mês de operação.
- 4 meses: Corresponde ao valor acumulado no primeiro quadrimestre, resultando na entrega do MVP.
- 9 meses: Representa o custo projetado para o ciclo de vida completo do projeto.

Tabela 5 – Custo Recorrente Total por Categoria

Categoria	Custo Mensal (R\$)	Custo Total (4 meses) (R\$)	Custo Total (9 meses) (R\$)
Mão de Obra	6.799,98	R\$ 27.199,92	R\$ 61.199,82
Estrutura	940,50	3.762,80	8.464,50
Total	7.740,48	30.962,72	69.664,32

Fonte: Elaborado pelos autores.

Dessa forma, Ao consolidar o investimento fixo em equipamentos com os custos recorrentes de estrutura e pessoal, o custo total estimado para a pousada parceira realizar o projeto ao longo de 9 meses é de R\$ 87.914,32.

6.2 Cenários de Comercialização

Este tópico propõe cenários hipotéticos de viabilidade comercial a longo prazo, considerando que a pousada parceira arcaria com os custos de desenvolvimento e manutenção futura do sistema. A arquitetura atual foi desenvolvida exclusivamente para o uso interno e administrativo da pousada, de forma que, trata-se de um software proprietário exclusivo.

Neste contexto, a primeira solução comercial analisada seria a aquisição da propriedade intelectual do sistema pela pousada, amortizando o custo total do projeto.

6.2.1 Cenário A: Aquisição da Propriedade Intelectual

Este cenário prevê o pagamento gradual do custo total de desenvolvimento do projeto, que foi de R\$87.914,32. Para fins de análise, vamos considerar um período de quatro anos (48 meses) para quitação do ativo. A escolha deste prazo fundamenta-se no conceito contábil de vida útil finita de um ativo intangível, conforme estabelecido no CPC 04 (R1) ([Comitê de Pronunciamentos Contábeis, 2010](#)). Já que os sistemas que possuem vida útil limitada por fatores como a dependência de frameworks de terceiros, a rápida evolução do mercado e a obsolescência tecnológica.

Além do custo de desenvolvimento, o cenário de aquisição também exigiria que a pousada arcasse com os custos operacionais contínuos de manutenção (corretiva, preventiva e adaptativa) e suporte, compostos por:

- **Mão de Obra Técnica:** Estimada em 2 horas mensais (1h de manutenção, 1h de suporte) com valor médio de R\$ 20,60/hora ([Tabela 4](#)), totalizando R\$ 41,20 por mês.
- **Hospedagem em Nuvem (AWS):** Estimada em R\$ 56,80 mensais ([Tabela 2](#)).

Dessa forma, o custo operacional mensal seria de aproximadamente R\$98,00.

Nesse cenário de aquisição, portanto, o custo total do projeto (R\$87.914,32) seria amortizado em 48 parcelas mensais de R\$1.831,55. Adicionalmente, seria necessário arcar com o custo operacional de R\$98,00, totalizando um custo mensal de R\$1.929,55 durante quatro anos.

6.2.2 Cenário B: Licenciamento de Uso e Manutenção

Neste cenário, a empresa parceira pagaria um custo mensal para o licenciamento de uso e manutenção do software, enquanto a equipe do projeto deteria a propriedade intelectual do sistema. Isso permitiria à equipe fazer ajustes e melhorias no sistema, a fim de torná-lo um Software como Serviço (SaaS) e comercializar o sistema para outras pousadas de pequeno e médio porte que ainda utilizam planilhas e papel na sua gestão, gerando receita e lucro em cima da aplicação projetada e reavendo, a longo prazo, o custo de estrutura e pessoal investidos inicialmente. Nesse arranjo, a pousada parceira continuaria atuando como o principal estudo de caso e parceiro de *design* do projeto, auxiliando na arquitetura, validação e *marketing* do sistema.

Em troca dessa parceria estratégica, ela continuaria utilizando o sistema em condições especiais, pagando uma mensalidade de licenciamento que cobre apenas os custos operacionais (hospedagem e manutenção). Ela teria, então, a opção de migrar para a plataforma SaaS completa quando esta estiver finalizada.

Adicionalmente, ter a pousada como o principal estudo de caso e parceiro de *design* ajudaria não só no desenvolvimento de um PMS, no modelo SaaS, robusto (por ter sido construído com base na realidade concreta de um gestor autônomo), como também nas campanhas de *marketing*. Inclusive, a fim de viabilizar a adaptação do sistema, na captação de investidores, usando-a na exemplificação e metrificação do projeto.

Para lançar a primeira versão do sistema, a equipe reutilizaria todo o código já desenvolvido, fazendo ajustes na arquitetura em nuvem e adaptações no banco de dados e na estrutura do código, para atender diferentes regras de negócio. Com o foco em um

público alvo constituído por pousadas com as mesmas características da pousada parceira: com 10 a 20 acomodações e gestão altamente manual. E com um lançamento que não prevê um produto com todas as funcionalidades presentes em um PMS tradicional, mas sim uma aplicação focada em simplicidade e usabilidade, constituindo-se como um serviço de rápida adoção e baixo custo que resolve o problema mais imediato do público-alvo.

Para além disso, o desenvolvimento de um modelo SaaS permite custos previsíveis e escalabilidade, dado que é um modelo de disponibilização de *software* em nuvem, o que permite o acesso dos usuários à aplicação via internet a partir de um pagamento periódico (Correia, 2024). Além de ser um modelo que independe de estruturas de operação e manutenção robustas para o usuário final, que consegue ter acesso a infraestrutura de *software* a um baixo custo, o que facilita a adesão de ferramentas tecnológicas para empresas de diferentes portes (CORREIA; BARRETO; ALVES, 2024). Sendo assim, conclui-se que este é um sistema financeiramente sustentável tanto para os desenvolvedores e investidores quanto para os futuros clientes.

6.2.3 Análise de Viabilidade dos Cenários

Para comprovar a viabilidade diante dos cenários propostos, foi feita a análise das respostas fornecidas pela proprietária [seção B.2](#) para medir os benefícios causados pela adoção do sistema. Foi utilizado como custo de mão de obra da proprietária o valor de R\$35,88/hora, baseado no salário médio de um empreendedor formalizado (SEBRAE, 2024).

1. **Otimização da Administração de Reservas:** Embora a gestora veja o envio da confirmação como uma tarefa imediata, ela também confirmou um fluxo manual de múltiplas etapas e diversos registros para se organizar. Neste contexto, estabeleceu-se uma premissa conservadora de 4 minutos economizados por reserva (considerando que o processo manual leva de 8 a 10 minutos e as funcionalidades atuais do sistema).
 - Cálculo: $(4 \text{ min}/\text{reserva} \times 15 \text{ reservas/mês}) = 60 \text{ min/mês}$
 - **Total Economizado:** R\$ 35,88 por mês
2. **Otimização da Gestão de Mensalistas:** Atualmente, a proprietária precisa gerir, paralelamente, hóspedes mensalistas com diferentes datas de vencimento. O que demanda esforço organizacional. O sistema irá auxiliar a gestora nesse processo de organização diminuindo a energia empregada em pelo menos 1 minuto por mensalista.
 - Premissa: Economia de 1 minuto por mensalista
 - Cálculo: $(1 \text{ min}/\text{mensalista} \times 10 \text{ mensalistas}) = 10 \text{ min/mês}$
 - **Total Economizado:** R\$ 5,98 por mês

3. Auxílio no Fechamento Financeiro: A gestora confirmou que a atividade de fechamento financeiro mensal é uma carga de trabalho pesada. Neste contexto, estima-se pelo menos 3h de trabalho mensal empregado nas tarefas relacionadas. E, para fins de análise, supõe-se que o sistema, com seus relatórios e filtros, economiza de 2h das 3h estimadas para a conferência manual de receitas e despesas.

- Cálculo: (2 horas economizadas/mês × R\$ 35,88/hora)
- **Total Economizado:** R\$ 71,76 por mês

O total de benefícios financeiros quantificáveis ($R\$ 35,88 + R\$ 5,98 + R\$ 71,76$) é de R\$ 113,62 por mês.

Esta análise comprova que o Cenário A, com um custo de R\$ 1.929,55, é inviável. O benefício de R\$ 113,62 não é suficiente para justificar o investimento. Mesmo com outros benefícios indiretos decorrentes da transição para uma gestão baseada em dados (relatórios) e não em achismos, seria um cenário de difícil manutenção para pousada. Isso porque, até em um cenário de aumento em 5 reservas diárias, decorrentes de estratégias de marketing, totalizando em 20 reservas mensais, cada uma com valor médio de R\$ 210,00, o valor restante mensal a ser pago para aquisição do sistema demandaria uma cessão de aproximadamente 40.20% da receita de cada reserva concluída. Um cenário inviável ao que tange a saúde financeira da pousada.

Em contrapartida, um cenário de licenciamento de uso através do estabelecimento de uma parceria comercial (Cenário B) é altamente viável. O custo da licença de R\$ 98,00 é facilmente coberto pelos benefícios quantificáveis de R\$ 113,62, gerando um retorno sobre o investimento (ROI) positivo imediato de R\$ 15,62 por mês. Este valor não inclui o benefício estratégico dos relatórios, que permitiria o aumento do retorno em até R\$ 210,00, no caso da geração de uma reserva a mais por mês.

7 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.

8 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

ALDRIGUI, M. *Meios de hospedagem*. São Paulo: Aleph, 2007. (Coleção ABC do Turismo). ISBN 978-85-7657-031-8. Citado na página 19.

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 74.

AWS. *AWS - Amazon Web Services*. 2025. Disponível em: <<https://aws.amazon.com/documentation/>>. Acesso em: 7 jun. 2025. Citado 3 vezes nas páginas 54, 57 e 63.

BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. 3. ed. [S.l.]: Addison-Wesley, 2012. Citado na página 53.

BRASIL. *Portaria nº 100, de 16 de junho de 2011*. Brasília, DF: [s.n.], 2011. Institui o Sistema Brasileiro de Classificação de Meios de Hospedagem (SBClass), estabelece os critérios de classificação destes, cria o Conselho Técnico Nacional de Classificação de Meios de Hospedagem (CTClass) e dá outras providências. Diário Oficial da União, 21 jun. 2011. Acesso em: 22 set. 2025. Citado na página 19.

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 62.

BRMODELO. *BRModelo Web*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://www.brmodeloweb.com/>>. Citado na página 60.

BUHALIS, D. Strategic use of information technologies in the tourism industry. *Tourism Management*, v. 19, n. 5, p. 409–421, 1998. ISSN 0261-5177. Disponível em: <https://www.researchgate.net/publication/222452921_Strategic_Use_of_Information_Technologies_in_the_Tourism_Industry>. Citado na página 20.

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 na página 32.

Comitê de Pronunciamentos Contábeis. *CPC 04 (R1)- Ativo Intangível*. 2010. <https://conteudo.cvm.gov.br/export/sites/cvm/menu/regulados/normascontabeis/cpc/CPC_04_R1_rev_12.pdf>. Acesso em: 29 out. 2025. Citado na página 77.

CORREIA, G. d. S.; BARRETO, G. S. d. L.; ALVES, N. d. M. *Crescimento e expansão no uso de software como serviço (SaaS): estratégias e obstáculos para empresas de tecnologia*. Centro Universitário Mário Pontes Jucá (UMJ), 2024. e149021 p. Recebido: 20/12/2023; Aceito: 11/01/2024; Publicado on-line: 12/01/2024. Disponível em: <<https://revistajrg.com/index.php/jrg/article/view/902>>. Citado na página 79.

DJANGO. *Django Documentation*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://docs.djangoproject.com/pt-br/4.2/>>. Citado 5 vezes nas páginas 53, 56, 58, 59 e 62.

DUCKETT. *HTML and CSS: Design and Build Websites*. [S.l.]: Wiley, 2011. ISBN 9781118008188. Citado 2 vezes nas páginas 53 e 57.

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 74.

FECOMERCIO. *Turismo nacional cresce 6,9% no primeiro semestre e fatura 108 bilhões*. 2025. Fecomercio SP. Disponível em: <<https://www.fecomercio.com.br/noticia/turismo-nacional-cresce-6-9-no-primeiro-semestre-e-fatura-108-bilhoes>>. Acesso em: 22 set. 2025. Citado na página 19.

FGV - Fundação Getúlio Vargas. *Estudo revela que 66% das Micro e Pequenas Empresas estão nos níveis iniciais de maturidade digital*. 2022. Disponível em: <<https://portal.fgv.br/noticias/estudo-revela-66-micro-e-pequenas-empresas-estao-niveis-iniciais-maturidade-digital>>. Citado na página 16.

GESTÃO PRO. *Os Riscos de Nao Usar um Sistema de Gestao em Pequenas Empresas*. 2025. Acessado em 29 de agosto de 2025. Disponível em: <<https://gestaopro.com.br/blog/gestao/os-riscos-de-nao-usar-um-sistema-de-gestao-em-pequenas-empresas>>. Citado na página 14.

GITHUB. *GitHub Docs*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://docs.github.com/>>. Citado 2 vezes nas páginas 60 e 61.

Glassdoor. *Salarios de empresas*. 2025. Disponível em: <<https://www.glassdoor.com.br/Salarios/index.htm>>. Acesso em: 05 jun. 2025. Citado na página 75.

GOOGLE. *Google Meet Help*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://support.google.com/meet/>>. Citado na página 61.

HELME, S. *SecurityHeaders.com — HTTP Security Headers Scanner*. 2025. Acesso em 14 nov. 2025. Disponível em: <<https://securityheaders.com/?q=app.pousadachalesaguadecoco.com.br&followRedirects=on>>. Citado na página 63.

HOFFMANN, R. C.; OLIVEIRA, P. S. M. d.; ZEFERINO, R. Z. A utilização estratégica dos sistemas de informações gerenciais no ramo hoteleiro da cidade de ponta grossa - paraná. *Revista de Engenharia e Tecnologia*, v. 4, n. 1, p. 18–33, Abril 2012. ISSN 2176-7270. Acesso em: 17 jun. 2025. Disponível em: <<https://revistas.uepg.br/index.php/ret/article/view/11290>>. Citado na página 20.

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 33.

IGNARRA, L. R. *Fundamentos do turismo*. 3. ed. São Paulo: Thomson Learning, 2013. Citado na página 19.

Indeed Brasil. *Salários de empresas*. 2025. Acesso em: 05 jun. 2025. Disponível em: <<https://br.indeed.com/career/salaries>>. Citado na página 75.

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 60.

LAUDON, K. C.; LAUDON, J. P. *Sistemas de Informação Gerenciais*. 16. ed. São Paulo: Pearson, 2020. Citado 2 vezes nas páginas [16](#) e [54](#).

Let's Encrypt. *Let's Encrypt*. 2025. <<https://letsencrypt.org/>>. Acesso em: 12 set. 2025. Citado na página [63](#).

LUNARDI, G. L.; DOLCI, P. C.; MAçADA, A. C. G. Adoção de tecnologia de informação e seu impacto no desempenho organizacional: um estudo realizado com micro e pequenas empresas. *Revista de Administração*, v. 45, n. 1, p. 5–17, 2010. Acesso em: 21 set. 2025. Disponível em: <<https://revistas.usp.br/rausp/article/view/44489>>. Citado na página [20](#).

MARTINS, C. A. d. M. G.; GONDIM, L. R. *Gestão hoteleira*. Manaus: Centro de Educação Tecnológica do Amazonas, 2011. 38 p. Acesso em: 15 jun. 2025. Disponível em: <https://redeetec.mec.gov.br/images/stories/pdf/eixo_hosp_lazer/061112_gest_hot.pdf>. Citado na página [20](#).

MAURÍCIO, N. R.; RAMOS, K. C. M. d. Gestão na hotelaria. *Revista F@ciência*, Apucarana – PR, v. 8, n. 11, p. 99–113, 2011. ISSN 1984-2333. Acesso em: 17 jun. 2025. Disponível em: <https://www.fap.com.br/fap-ciencia/edicao_2011/011.pdf>. Citado na página [20](#).

MELL, P.; GRANCE, T. *The NIST Definition of Cloud Computing*. [S.l.], 2011. Acesso em: 25 set. 2025. Disponível em: <<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>>. Citado na página [57](#).

MINISTERIO DO TURISMO. *Micro e pequenas empresas representam 89 dos empreendimentos do setor de Turismo*. [S.l.], 2025. Acessado em 29 de agosto de 2025. Disponível em: <<https://www.gov.br/turismo/pt-br/assuntos/noticias/micro-e-pequenas-empresas-representam-89-dos-empreendimentos-do-setor-de-turismo>>. Citado 2 vezes nas páginas [14](#) e [16](#).

MOURA, V. *Descubra diferentes tipos de hospedagens pelo país*. 2023. Assessoria de Comunicação do Ministério do Turismo. Acesso em: 22 set. 2025. Disponível em: <<https://www.gov.br/turismo/pt-br/assuntos/noticias/descubra-diferentes-tipos-de-hospedagens-pelo-pais>>. Citado na página [20](#).

OWASP Foundation. *OWASP Top 10: The Ten Most Critical Web Application Security Risks*. [S.l.], 2021. Acessado em: 20 set. 2025. Disponível em: <<https://owasp.org/Top10/>>. Citado 2 vezes nas páginas [62](#) e [63](#).

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 [53](#) e [57](#).

PRESSMAN, R. S. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: McGraw-Hill, 2016. Citado na página [52](#).

PROJECTLIBRE. *ProjectLibre*. [S.l.], 2025. Acessado em 24 de junho de 2025. Disponível em: <<https://www.projectlibre.com/>>. Citado na página [61](#).

PYTHON. *Python Official Documentation*. 2025. Disponível em: <<https://docs.python.org/3/>>. Acesso em: 7 jun. 2025. Citado na página [56](#).

Qualys SSL Labs. *SSL Server Test — análise do host app.pousadachalesaguadecoco.com.br*. 2025. Disponível em: <<https://www.ssllabs.com/ssltest/analyze.html?d=app.pousadachalesaguadecoco.com.br>>. Citado na página 63.

Receita Federal do Brasil. *Consulta Externa — Norma nº 81268*. 2017. <<https://normasinternet2.receita.fazenda.gov.br/#/consulta/externa/81268>>. Acessado em: 30/10/2025. Citado na página 74.

RESCORLA, E. *The Transport Layer Security (TLS) Protocol Version 1.3*. 2018. RFC 8446. Acessado em: 20 set. 2025. Disponível em: <<https://www.rfc-editor.org/rfc/rfc8446.html>>. Citado na página 63.

ROSSUM, G. V. *PEP 8 – Style Guide for Python Code*. 2013. <<https://peps.python.org/pep-0008/>>. Acesso em: 20 out. 2025. Citado 2 vezes nas páginas 58 e 59.

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 2 vezes nas páginas 22 e 24.

SEBRAE. *Maioria das micro e pequenas empresas ainda opta pela gestão manual*. 2020. Disponível em: <<https://www.segs.com.br/seguros/268289-maioria-das-micro-e-pequenas-empresas-ainda-opta-pela-gestao-manual>>. Citado na página 16.

SEBRAE. *Empresas formais têm rendimento médio 3 vezes maior que negócios sem CNPJ*. 2024. Disponível em: <<https://agenciasebrae.com.br/economia-e-politica/empresas-formais-tem-rendimento-medio-3-vezes-maior-que-negocios-sem-cnpj/>>. Citado na página 79.

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 2 vezes nas páginas 26 e 34.

SOMMERVILLE, I. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019. Citado 2 vezes nas páginas 52 e 53.

STAIR, R.; REYNOLDS, G. *Princípios de Sistemas de Informação*. 13. ed. São Paulo: Cengage Learning, 2020. Citado na página 16.

TOTVS. *Digitalização no setor hoteleiro: os benefícios da tecnologia*. 2024. Acessado em 29 de agosto de 2025. Disponível em: <<https://www.totvs.com/blog/gestao-hoteleira/digitalizacao-no-setor-hoteleiro/>>. Citado na página 14.

TURBAN, E.; POLLARD, C.; WOOD, G. *Sistemas de Informação: Uma Abordagem Gerencial*. 9. ed. Porto Alegre: Bookman, 2018. Citado na página 54.

United Nations. *International Recommendations for Tourism Statistics 2008 (Revised 1)*. New York, 2010. United Nations. Studies in Methods, Series M No. 83/Rev.1. P. 21. Acesso em: 22 set. 2025. Disponível em: <https://unstats.un.org/unsd/publication/SeriesM/SeriesM_83rev1e.pdf>. Citado na página 19.

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 60.

ZANELLA, A.; ANGELONI, M. T. Pousadas - uma alternativa criativa de hospedagem. *Turismo - Visão e Ação*, Universidade do Vale do Itajaí, Camboriú, Brasil, v. 8, n. 2, p. 253–271, mai–jul 2006. Citado 2 vezes nas páginas 19 e 20.

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, entendo 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:
 - Guilherme Akio: Analista de Testes.
 - Guilherme Schmidt: Engenheiro 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 4º 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 5º SEMANA

Período: 22/04/2025 a 29/04/2025.

Essa semana foram desenvolvidas atividades ligadas ao desenvolvimento da prova de conceito. Foram elas:

- Kelly Radchelle: Documentação dos Casos de Uso.
- Guilherme Schmidt: Documentação dos Diagramas de Caso de Uso.
- Reunião via Google Meet na qual foi discutido o desenho da aplicação com todos os membros da equipe.
- Guilherme Bittencourt: Criação do Diagrama de Componentes.
- Kelly Radchelle: Criação do Diagrama de Implantação.
- Guilherme Schmidt, Guilherme Akio, Rafael Teixeira e Ricardo Carriel: Apresentação do Desenho da Aplicação.

A.6 6º 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.

- Guilherme Schmidt: Alterações no diagrama de componentes.
- Kelly Radchelle: Alterações no diagrama de implantação.
- Guilherme Schmidt: Criação do MER.
- Guilherme Schmidt: Criação do repositório Git para versionamento da aplicação.
- Kelly Radchelle e Ricardo Carriel: Levantamento das regras de negócio e requisitos com a proprietária.

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:

- Guilherme Akio: Início do desenvolvimento do backend e frontend no Django: views, models e templates.
- Ricardo Carriel: Início da configuração do ambiente de hospedagem.

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.

- Ricardo Carriel: Atualizações no código. .
- Kelly Radchelle: Desenvolvimento da Documentação.
- Guilherme Schmidt: Desenvolvimento da Documentação.

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.

Nesta semana o foco da equipe se manteve em desenvolver e revisar os tópicos da documentação. Responsabilidades de cada integrante:

- Guilherme Akio: Documentação do código e das funcionalidades desenvolvidas por si.
- Guilherme Schmidt: Entregar seus tópicos da documentação (Introdução, Manutenibilidade, Segurança e Privacidade e Arquitetura do Sistema) e auxiliar na revisão e desenvolvimento da Documentação.
- Kelly Radchelle: Entregar seus tópicos da documentação (Gestão do Projeto, Escopo do Projeto e Histórias de Usuário) e auxiliar na revisão e desenvolvimento da documentação final.
- Rafael Teixeira: Documentação do código e das funcionalidades desenvolvidas por si.

A.12 12º SEMANA

Período: 10/06/2025 a 17/06/2025.

Nesta semana a equipe corrigiu e revisou os tópicos da documentação e realizou alterações na estrutura do código para corrigir erros e facilitar o entendimento do mesmo. Além de, corrigir erros recorrentes no ProjectLibre.

A.13 13º SEMANA

Período: 17/06/2025 a 24/06/2025.

Durante essa semana a documentação final consolidada foi revisada, para aplicar aspectos faltantes das normas abnt. Desenvolveu as funcionalidades restantes relacionadas ao módulo de reserva e check-in/check-out. E iniciou os preparativos para a apresentação final do MVP.

A.14 Período: 13/08/2025 a 26/08/2025.

Durante esse período de retorno as atividades do projeto, fizemos a revisão dos requisitos e regras de negócio finais do sistema, com o objetivo de planejar as sprints da segunda fase do projeto. Levantamos funcionalidades essenciais e que ficaram pendentes da primeira fase. Definimos a nova ferramenta para acompanhamento das atividades (Github - Project) das sprints e como seriam divididas as issues/tasks.

A.15 14º SEMANA

Período: 27/08/2025 a 02/09/2025. Durante a primeira sprint da segunda fase do projeto, foram planejadas atividades de revisão e retomada de funcionalidades pendentes do sistema e planejamento do plano de testes:

- Rafael Teixeira: Adicionar campo de motivo de cancelamento de reserva.
- Rafael Teixeira: Criar função de editar cadastro de quartos .
- Guilherme Schimdt: Revisar e Corrigir tópico Introdução.
- Guilherme Schimdt: Revisar e corrigir o tópico objetivo.
- Kelly Radchelle: Revisar e ajustar product backlog.
- Kelly Radchelle: Revisar e atualizar o escopo do projeto.

- Ricardo Carriel: Revisar e padronizar código.
- Guilherme Akio: Criar estrutura do plano de teste.

A.16 15° SEMANA

Período: 03/09/2025 a 10/09/2025. Nesta semana foi iniciado o desenvolvimento e implementação de novas funcionalidades relacionadas ao módulo financeiro. As atividades foram distribuídas:

- Rafael Teixeira: Criar tabela financeiro_título.
- Ricardo Carriel: Criar app Financeiro com configurações iniciais.
- Kelly Radchelle: Revisar e atualizar tópico histórias de usuário.
- Kelly Radchelle: Atualizar tópico de Sprint Backlog.
- Guilherme Schimdt: Revisar e Corrigir Justificativa.
- Guilherme Schimdt: Revisar e corrigir tópico de Análise da concorrência.
- Guilherme Akio: Testar módulo de gestão de hóspedes.

A.17 16° SEMANA

Período: 11/09/2025 a 17/09/2025. Durante a semana foi planejado realizar novas implementações no módulo financeiro e de autenticação. As atividades foram distribuídas:

- Rafael Teixeira: Criar tabela financeiro_categoria.
- Guilherme Schimdt: Revisar e corrigir tópico de Modelagem do banco de dados.
- Kelly Radchelle: Revisar e corrigir tópico de gestão do projeto.
- Kelly Radchelle: Implementar login/logout.
- Ricardo Carriel: Criar tabela financeiro_banco.
- Guilherme Akio: Realizar testes no módulo gestão de reservas.

Além disso, foram revistos os requisitos funcionais finais com a empresa parceira.

A.18 17° SEMANA

Período: 18/09/2025 a 24/09/2025. Para a sprint dessa semana foram planejadas as tarefas:

- Rafael Teixeira: Criar views de CRUD para Títulos.
- Rafael Teixeira: Ajuste - Adicionar campo de motivo de cancelamento de reserva.
- Guilherme Akio: Realizar testes no módulo gestão de quartos.
- Ricardo Carriel: Implementar integração com módulo de reservas.
- Guilherme Schimdt: Revisar e corrigir tópico de Segurança .
- Guilherme Schimdt: Realizar testes funcionais módulo de gestão de hóspedes.
- Kelly Radchelle: Revisar e Corrigir Revisão de Literatura.
- Kelly Radchelle: Criar telas de autenticação.

A.19 18° SEMANA

Período: 25/09/2025 a 01/10/2025. Para a sprint dessa semana foram planejadas as tarefas:

- Rafael Teixeira: Criar filtros para lista de reservas.
- Rafael Teixeira: Implementar cadastro de categorias de despesa.
- Kelly Radchelle: Criar função de envio de e-mail de confirmação de reserva.
- Guilherme Schimdt: Realizar testes funcionais módulo de gestão de quartos.
- Guilherme Akio: Realizar ajustes módulo hóspedes.
- Guilherme Schimdt: Revisar e corrigir a Arquitetura do projeto.
- Guilherme Schimdt: Realizar testes funcionais módulo de gestão de reservas.
- Ricardo Carriel: Criar funcionalidade de baixa do pagamento.
- Ricardo Carriel: Implementar regra de pagamento parcial para confirmação da reserva.
- Kelly Radchelle: Revisar e corrigir tópico de Viabilidade financeira .

A.20 19° SEMANA

Período: 02/10/2025 a 08/10/2025. Para a sprint dessa semana foram planejadas as tarefas:

- Guilherme Akio: Testes de funcionais de autenticação.
- Guilherme Akio: Criar filtros por categoria para títulos.
- Rafael Teixeira: Criar view de balanço financeiro.
- Rafael Teixeira: Criar pacotes de reservas.
- Guilherme Schimdt: Revisar e corrigir tópico de Tecnologias e ferramentas de apoio utilizadas.
- Guilherme Schimdt: Realizar testes unitários do CRUD de títulos.
- Ricardo Carriel: Criar funcionalidade de adição de tarifas à reserva.
- Ricardo Carriel: Criar templates de título.
- Kelly Radchelle: Realizar ajustes no módulo quarto.
- Kelly Radchelle: Realizar ajustes módulo de reservas.

E envio do documento de requisitos funcionais finais para a empresa parceira e para o professor orientador.

A.21 20° SEMANA

Período: 09/10/2025 a 15/10/2025. Para a sprint dessa semana foram estipuladas as tarefas:

- Rafael Teixeira: Criar view do painel de métricas.
- Ricardo Carriel: Criar templates para tarifas.
- Guilherme Akio: Criar view relatórios: financeiro.
- Guilherme Schimdt: Revisão e atualização dos diagramas.
- Kelly Radchelle: Criar view relatórios: ocupação.

A.22 21° SEMANA

Período: 16/10/ 2025 a 22/10/2025. Para a sprint dessa semana foram estipuladas as tarefas:

- Rafael Teixeira: Criar template: painel de métricas.
- Guilherme Schimdt: Adicionar tópico de estilo de código na documentação.
- Kelly Radchelle: Criar template relatório de ocupação.
- Guilherme Akio: Criar template relatórios financeiros.
- Ricardo Carriel: Configurar envio de email no servidor AWS.

A.23 22° SEMANA

Período: 23/10/2025 a 29/10/2025. Para a sprint dessa semana foram determinadas as tarefas:

- Kelly Radchelle: Ajustar e padronizar templates.
- Guilherme Schimdt: Realizar testes funcionais: títulos.
- Guilherme Akio: Adicionar lógica de disponibilidade de quartos.
- Ricardo Carriel: Adicionar função de early check-in e late check-out.
- Rafael Teixeira: Criar funcionalidade de notificação.

A.24 23° SEMANA

Período: 30/10/2025 a 5/11/2025. Essa semana foi separada para realização de novos testes e para finalização de issues não finalizadas, a fim de conseguir fazer a entrega do produto final (prevista para o dia 12/11). Foram elas:

- Guilherme Schimdt: Revisar e Realizar testes unitários. (Responsável:)
- Kelly Radchelle: Revisar e corrigir tópico de Repositório da Aplicação.
- Guilherme Akio: Finalizar issues sob responsabilidade própria em atraso.
- Rafael Teixeira: Finalizar issues sob responsabilidade própria em atraso.
- Ricardo Carriel: Finalizar issues sob responsabilidade própria em atraso.

A.25 24° SEMANA

Período: 30/10/2025 a 11/5/2025. Durante essa semana foi dado continuidade a tarefas não finalizadas e novos testes unitários. Além da revisão final da documentação.

APÊNDICE B – Questionário Aplicado à Proprietária da Pousada

B.1 Questionário 1: Levantamento de Requisitos

O [Quadro 11](#) foi aplicado à proprietária da pousada parceira para a análise e levantamento dos requisitos do sistema.

Quadro 11 – Questionário Aplicado à Proprietária 1 (Parte 1)

Domínio: Reservas		
Item	Pergunta	Resposta
1	Como os clientes normalmente fazem uma reserva hoje (telefone, WhatsApp, pessoalmente)?	Geralmente, por WhatsApp.
2	Há um prazo mínimo ou máximo para fazer uma reserva?	Preferencialmente, antecipadamente. Nas plataformas coloco 2 dias de antecedência.
3	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.
4	É permitido cancelar uma reserva? Até quantas horas antes do check-in? Há cobrança de taxa?	Sim! Temos políticas de cancelamentos e remarcações.
5	Um hóspede pode fazer mais de uma reserva ativa ao mesmo tempo?	Sim.
Domínio: Hóspedes		
Item	Pergunta	Resposta
6	Quais dados do hóspede são obrigatórios para fazer uma reserva?	Nome completo, endereço completo, CPF, telefone, e-mail.
7	É comum ter reservas feitas por um responsável em nome de outros hóspedes?	Sim.
8	Há um limite de pessoas por quarto? Como isso é controlado?	No check-in.

Fonte: Elaborado pelos autores.

Quadro 11 – Questionário Aplicado à Proprietária 1 (Parte 2)

Domínio: Check-in e Check-out		
Item	Pergunta	Resposta
9	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 das 8h até às 14h. Depende de se há entrada de outro hóspede em seguida.
10	O check-in pode ser feito antes do horário? E o check-out após o horário?	Depende, se houver saída de hóspede anterior ou entrada em seguida.
11	Quem realiza o check-in e check-out? Você ou os funcionários?	Eu ou sozinhos, com orientações minhas.
12	Há necessidade de gerar comprovante ou recibo após check-in ou check-out?	Não.
13	A pousada possui quantos quartos? Como eles são classificados?	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
14	Há períodos em que quartos são bloqueados para manutenção ?	Sim
15	Um mesmo quarto pode ser reservado para diferentes hóspedes em dias seguidos?	Sim.
Domínio: Serviços Adicionais		
Item	Pergunta	Resposta
16	A pousada oferece serviços extras?	Não.
17	Se sim, esses serviços devem ser registrados no sistema junto à reserva?	(Não há serviços extras)
Domínio: Pagamentos		
Item	Pergunta	Resposta
18	Quais formas de pagamento são aceitas (Pix, cartão, dinheiro)?	As três formas, porém no cartão tem taxa da operadora.
19	Os pagamentos são feitos no check-in, no check-out ou antecipadamente?	Metade na reserva e o restante na chegada.
20	É necessário gerar comprovante ou recibo no sistema?	Eu envio uma confirmação de reserva com todas as informações.

Fonte: Elaborado pelos autores.

Quadro 11 – Questionário Aplicado à Proprietária 1 (Parte 3)

Domínio: Comunicação		
Item	Pergunta	Resposta
21	Você gostaria que o sistema enviasse confirmação automática de reserva por WhatsApp ou e-mail?	Sim.
22	Há interesse em receber alertas automáticos de check-in, check-out ou cancelamento?	Sim.
Domínio: Acesso ao Sistema		
Item	Pergunta	Resposta
23	Somente você vai usar o sistema ou os funcionários também?	Somente eu.
24	Deseja que cada pessoa tenha um tipo de acesso diferente?	(Somente a proprietária vai usar o sistema).
25	O sistema será usado no computador, celular ou ambos?	Em ambos.
Domínio: Relatórios e Controle		
Item	Pergunta	Resposta
26	Quais relatórios são mais importantes no dia a dia?	Ocupação diária, reservas da semana, totais de pagamento, período, ...
27	É importante ter um histórico de cada hóspede e das reservas anteriores?	Sim.

Fonte: Elaborado pelos autores.

B.2 Questionário 2: Viabilidade Financeira

O [Quadro 12](#) foi aplicado à proprietária da pousada parceira para a análise da viabilidade financeira do projeto

Quadro 12 – Questionário Aplicado à Proprietária 2

Item	Pergunta	Resposta
1	Pensando em uma reserva de diária (do primeiro contato até enviar a confirmação), quanto tempo você leva em média para fazer tudo?	Tudo depende do cliente, às vezes a demora ocorre pela indecisão ou espera de outros orçamentos por parte dele. Assim que o hóspede efetua o pagamento eu já envio a confirmação
2	E quantas reservas de diária você gerencia por mês, mais ou menos?	Depende também da época do ano e do tempo (clima), por ser na praia. Não são muitos pq tenho poucos disponíveis devido aos mensalistas, mas esse mês umas 15 diárias.
3	Em média, quantos hóspedes mensalistas você costuma ter?	No momento estou com 10 mensalistas.
4	Todo mês, quanto tempo você gasta para gerenciar o pagamento de cada mensalista?	Os dias não são geralmente os mesmos, por isso não leva tanto tempo.
5	No fim do mês, quantas horas você leva para fazer o fechamento financeiro?	No fim do mês fico bem atarefada por que deixo para emitir as NF, fazer fechamento financeiro, administrar RH, pagamentos, etc.
6	Pensando em erros de agendamento (overbooking): Isso já aconteceu no último ano por ser no controle manual?	Como sou bem organizada e só eu cuido disso, não aconteceu comigo pq faço vários registros.
7	Hoje, se eu te pedisse a taxa de ocupação de 6 meses atrás, seria fácil saber?	Seria fácil porque estávamos na baixa temporada e a taxa de ocupação é baixa, em média uns 5%.

Fonte: Elaborado pelos autores.

APÊNDICE C – Product Backlog Detalhado

Quadro 13 – Product Backlog Detalhado (Parte 1)

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 funcionalidades 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 13 – Product Backlog Detalhado (Parte 2)

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 e Check-out</i>	ALTA
32	Criar funcionalidade de atualização do <i>check-out</i>	<i>Check-in e Check-out</i>	ALTA
33	Integrar ambientes, <i>backend</i> e <i>frontend</i>	Requisito técnico	ALTA
34	Documentar a configuração e o código desenvolvido	Documentação	ALTA
35	Criar funcionalidade de cadastro de receitas	Gestão Financeira	BAIXA

Fonte: Elaborado pelos autores.

Quadro 13 – Product Backlog Detalhado (Parte 3)

Código	Item	Categoria	Prioridade
36	Criar funcionalidade para registrar comprovação de pagamento de reservas	Gestão Financeira/Gestão de Reservas	BAIXA
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

Fonte: Elaborado pelos autores.

Quadro 13 – Product Backlog Detalhado (Parte 4)

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

Fonte: Elaborado pelos autores.

APÊNDICE D – Sprint Backlog Detalhado

Quadro 14 – Sprint Backlog Detalhado (Parte 1)

Sprint	Período	Objetivo	Atividades
1	26/03/2025 a 02/04/2025	Fundação e Alinhamento do Projeto	Organização do tema Definição da metodologia de trabalho; Definição ferramentas de gestão; Definição das plataformas e tecnologias de desenvolvimento
2	03/04/2025 a 09/04/2025	Detalhamento Técnico e Análise de Viabilidade	Definição dos frameworks e bibliotecas; Definição das ferramentas de desenvolvimento e hospedagem/publicação; Levantamento de custos
3	10/04/2025 a 16/04/2025	Estruturação do Escopo	Levantamento dos requisitos; Mapeamento dos casos de uso; Desenvolver Product Backlog Inicial
4	17/04/2025 a 23/04/2025	Estruturação da Arquitetura e do MVP	Definição do escopo do MVP; Organizar repositório e fluxo GitHub; Criação do Diagrama de Componentes; Criação do Diagrama de Implantação
5	24/04/2025 a 30/04/2025	Avanço na Estruturação da Arquitetura	Criar do modelo relacional (MER); Criar do diagrama entidade-relacionamento (DER); Configurar ambiente e servidor local; Configurar Banco de Dados (PostgreSQL)
6	01/05/2025 a 07/05/2025	Implementação da Estrutura Base e Módulos Iniciais	Estrutura inicial do Backend: Criar App Quarto; Criar App Hóspede; Criar App Reserva; Criar Usuário Admin. Estrutura inicial do frontend: configurações tailwind e daisiyu
7	08/05/2025 a 14/05/2025	Desenvolvimento do Cadastro de Quartos	Revisão dos Requisitos e Diagramas; Criar funcionalidade de cadastro de quartos: Criar Model Quarto; Executar migrations; Criar formulário de cadastro; Criar view de criação (QuartoViewCreate); Mapear rota (URL) para view criada

Fonte: Elaborado pelos autores.

Quadro 14 – Sprint Backlog Detalhado (Parte 2)

Sprint	Período	Objetivo	Atividades
8	15/05/2025 a 21/05/2025	Implementação do CRUD de Quartos	Criar interfaces para gestão de quartos; Criar template do formulário de cadastro. Criar funcionalidades de edição e exclusão do cadastro de quartos: Criar view de edição (QuartoViewUpdate); Criar view de exclusão (QuartoViewDelete); Criar template de edição de quarto; Criar template de exclusão de quarto; Mapear rotas (URL) para views criadas. Criar funcionalidade de listagem de quartos: Criar view de listagem (QuartoViewList); Criar template de listagem de quartos; Mapear rota (URL) para view criada.
9	22/05/2025 a 28/05/2025	Implementação do CRUD de Hóspedes	Criar funcionalidade de cadastro de hóspedes: Criar Model Hóspede; Executar migrations; Criar formulário de cadastro; Criar view de criação (HospedeViewCreate); Mapear rota (URL) para view criada. Criar funcionalidades de edição e exclusão do cadastro de hóspedes: Criar view de edição (HospedeViewUpdate); Criar view de exclusão (HospedeViewDelete); Mapear rotas (URL) para views criadas
10	29/05/2025 a 04/06/2025	Estruturação do Módulo de Reservas e Conclusão da Gestão de Hóspedes	Criar interfaces para gestão de hóspedes: Criar template do formulário de cadastro de hóspede; Criar template de edição de hóspede; Criar template de exclusão de hóspede; Criar template de listagem de hóspede. Criar funcionalidade de cadastro de reservas: Criar Model Reserva; Executar migrations; Criar formulário de cadastro; Criar view de criação ; Mapear rota (URL) para view criada. Criar funcionalidades de edição e exclusão do cadastro de reservas: Criar view de edição; Criar view de exclusão; Mapear rotas (URL) para views criadas; Consolidar toda a documentação técnica e de requisitos

Fonte: Elaborado pelos autores.

Quadro 14 – Sprint Backlog Detalhado (Parte 3)

Sprint	Período	Objetivo	Atividades
11	05/06/2025 a 11/06/2025	Operacionalização do Módulo de Reservas	Finalizar Documentação Técnica e Gerar PDF; Criar interfaces para gestão de Reservas: Criar template do formulário de cadastro de reserva; Criar template de edição de reserva; Criar template de exclusão de reserva; Criar template de listagem de reserva. Criar funcionalidades de check-in e check-out: Adicionar função de check-in na view Reserva; Adicionar função de check-out na view Reserva; Adicionar função de registrar check-in/check-out no template de listagem de reservas; Criar templates para check-in do dia (hoje); Criar templates para check-out do dia (hoje)
12	12/06/2025 a 18/06/2025	Preparação e Lançamento do MVP	Revisar formatação e ortografia da documentação; Revisar e fazer ajustes no código: Revisar alinhamento com requisitos do projeto; Ajustar possíveis bugs; Entrega do MVP
13	19/06/2025 a 25/06/2025	Apresentação do MVP	Preparo do material da apresentação do MVP
14	27/08/2025 a 02/09/2025	Consolidação da Qualidade e Evolução do MVP	Ajuste e adição de funcionalidades ao MVP: Adicionar campo de motivo de cancelamento de reserva; Adicionar campo no template de listagem de reservas; Editar função de edição do cadastro de quartos; Adicionar ajustes de adequação da função com o escopo. Revisar e padronizar código; Ajustar código de acordo com as boas práticas do Django; Criar estrutura do plano de testes; Definir o que será testado e como; Documentar escopo de testes. Documentação: Revisar e corrigir tópico Introdução; Revisar e corrigir o tópico objetivo; Revisar e ajustar <i>product backlog</i> ; Revisar e atualizar o escopo do projeto

Fonte: Elaborado pelos autores.

Quadro 14 – Sprint Backlog Detalhado (Parte 4)

Sprint	Período	Objetivo	Atividades
15	03/09/2025 a 10/09/2025	Estruturação do Módulo Financeiro e Implementação de Testes Unitários	Criar app Financeiro com configurações iniciais: Executar python manage.py startapp quartos e adicioná-lo ao INSTALLED_APPS Criar tabela financeiro_título : Criar model Título; Executar migrations; Realizar testes unitários no módulo de gestão de hóspedes. Testar funções CRUD do app Hóspede. Documentação: Revisar histórias de usuário (garantir clareza, critérios de aceite iniciais e alinhamento com backlog); Atualizar sprint backlog de acordo com ajustes feitos; Revisar e corrigir a Análise da concorrência; Revisar e corrigir a justificativa
16	11/09/2025 a 17/09/2025	Implementação do Sistema de Autenticação e Evolução do Módulo Financeiro	Configurar autenticação Django: Importar biblioteca django allauth; Ajustar mapeamento de rotas de autenticação; Ajustar configurações de acesso (autorização). Criar tabela financeiro_categoria: Criar model Categoria. Executar migrations. Criar tabela financeiro_banco: Criar model Banco; Executar migrations. Realizar testes no módulo de gestão de Reservas: Testar funções CRUD do app Hóspede. Documentação: Revisar e corrigir tópico de Modelagem do banco de dados; Revisar e corrigir tópico de gestão do projeto

Fonte: Elaborado pelos autores.

Quadro 14 – Sprint Backlog Detalhado (Parte 5)

Sprint	Período	Objetivo	Atividades
17	18/09/2025 a 24/09/2025	Implementação da Gestão de Títulos Financeiros e Validação dos Módulos Core	<p>Desenvolver funcionalidade CRUD de Títulos: Criar view de títulos financeiros no app Título; -Criar função de criação títulos financeiros ; Criar função de edição títulos financeiros; Criar função de listar títulos financeiros; Criar formulário para criação de títulos financeiros.</p> <p>Implementar integração com módulo de reservas: Adicionar lógica de criação de um título de receita relacionado ao registro de uma nova reserva.</p> <p>Ajuste: Adicionar campo de motivo de cancelamento de reserva.</p> <p>Personalizar templates padrões de login, alteração de senha e logout do django allauth.</p> <p>Realizar testes unitários módulo de gestão de quartos: Testar funções CRUD do app Quarto.</p> <p>Realizar testes funcionais módulo de gestão de hóspedes: Testar funções CRUD do app Hóspede.</p> <p>Documentação: Revisar e corrigir tópico de Segurança; Revisar e corrigir tópico de Revisão de Literatura</p>

Fonte: Elaborado pelos autores.

Quadro 14 – Sprint Backlog Detalhado (Parte 6)

Sprint	Período	Objetivo	Atividades
18	25/09/2025 a 01/10/2025	Automação do Fluxo de Reservas e Validação de Regras de Negócio	Criar função de envio de e-mail de confirmação de reserva: Adicionar configurações de envio de email do django; Adicionar campo de envio no model de Reserva; Adicionar lógica de envio na view Reserva; Adicionar função no template de listagem de reservas. Implementar regra de pagamento parcial para confirmação da reserva: Adicionar lógica de criação de 2 títulos relacionados a uma reserva; Adicionar função de troca de status da reserva quando pagamento do sinal for confirmado. Criar funcionalidade de baixa do pagamento: Adicionar lógica de confirmação de pagamento do título na view Título. Realizar ajustes módulo hóspedes: Sistema não deve permitir o cadastro de hóspedes menores de idade ou com data de nascimento no ano vigente ou posterior; A usuária deve conseguir acessar um histórico do hóspedes e visualizar todas as reservas já cadastradas no cpf : criar função na view hóspede e template; Arrumar erro de busca por nome do hóspede; Adicionar atributo: endereço. Criar filtros para lista de reservas: Adicionar filtros no template de listagem de reservas: por período, quarto, por status (canceladas). Implementar cadastro de categorias de despesa. Realizar testes funcionais módulo de gestão de quartos: Testar funções CRUD do app Hóspede; Realizar testes funcionais módulo de gestão de reservas: Testar funções CRUD do app Hóspede. Documentação: Revisar e corrigir tópico de Arquitetura do Projeto; Revisar e corrigir tópico de Viabilidade Financeira

Fonte: Elaborado pelos autores.

Quadro 14 – Sprint Backlog Detalhado (Parte 7)

Sprint	Período	Objetivo	Atividades
19	02/10/2025 - 08/10/2025	Operacionalização do Módulo Financeiro e Sofisticação das Reservas	<p>Criar lógica da funcionalidade: balanço financeiro. Criar filtros por categoria para títulos: Adicionar filtros no template de listagem de títulos financeiros: tipo (despesa/receita), tipo de documento, tipo conta corrente.</p> <p>Criar template CRUD de título: Criar templates de títulos; Criar template de listagem de títulos: adicionar filtros, função de edição e exclusão; Criar template do formulário para criação de títulos financeiros.</p> <p>Módulo de reservas-criar funcionalidade de adição de tarifas: Criar Model Tarifa; Executar Migrations; Criar view de Tarifa com funções de criação, edição e exclusão.</p> <p>Módulo de reservas: Criar pacotes de reservas (mensalista). Realizar ajustes módulo quartos. Realizar ajustes módulo reservas.</p> <p>Realizar testes unitários de títulos: cadastro e edição. Realizar testes unitários autenticação.</p> <p>Documentação: revisar e corrigir tópico de Tecnologias e ferramentas de apoio utilizadas</p>
20	09/10/2025 - 15/10/2025	Implementação de Relatórios e Dashboard	<p>Criar view relatórios: ocupação (temporada, mensal, quarto). Criar view relatórios: financeiro (tipo:receitas/despesas, categoria, período, faturamento). Criar view: painel de métricas.</p> <p>Realizar testes unitários: baixa de pagamento</p> <p>Realizar testes funcionais de títulos: cadastro e edição</p> <p>Documentação: Revisão e atualização dos diagramas (MER, DER, Casos de Uso)</p>

Fonte: Elaborado pelos autores.

Quadro 14 – Sprint Backlog Detalhado (Parte 8)

Sprint	Período	Objetivo	Atividades
21	16/10/ 2025 a 22/10/2025	Instrumentação do Sistema com Relatórios e Notificações Operacionais	<p>Configurar envio de email no servidor AWS: Adicionar configuração SMTP no servidor; Atualizar configurações no settings. py de envio de email.</p> <p>Criar template: painel de métricas. Criar template relatórios: ocupação (temporada, mensal, quarto). Criar template relatórios: financeiro (tipo:receitas/despesas, categoria, período).</p> <p>Implementar exportação de relatórios financeiro/ocupação em PDF/Excel.</p> <p>Criar funcionalidade notificação do módulo reservas (nova reserva, reserva pendente). Criar funcionalidade notificação do módulo quartos (limpeza pós check-out).</p> <p>Realizar testes unitários: cálculos balanço financeiro. Realizar testes unitários: funções CRUD de Tarifas.</p> <p>Documentação: Revisão e atualização dos diagramas (Implantação, Componentes)</p>
22	23/10/2025 a 29/10/2025	Validação Completa e Homologação do Sistema	<p>Realizar testes funcionais completos (finanças, relatórios). Realizar testes de envio de email (Integração AWS).</p> <p>Realizar testes de integração/backup com PostgreSQL. Documentação: revisar e corrigir tópico de Repositório da aplicação; Ajustes parciais dos tópicos revisados anteriormente</p>
23	30/10/2025 a 5/11/2025	Testes Finais	<p>Documentar testes finais. Ajustes finais do código. Realizar testes de headers de segurança . Realizar testes de SSL .</p> <p>Documentação: ajustes finais.</p>
24	30/10/2025 a 11/5/2025	Preparação para Implantação Final	Ajustes finais de bugs. Revisão de documentação completa. Deploy final.
25	12/11/2025 a 22/11/2025	Apresentação do Produto Final	Preparar apresentação final do projeto

Fonte: Elaborado pelos autores.

APÊNDICE E – Requisitos Funcionais

Detalhados

Quadro 15 – Requisitos Funcionais (Parte 1)

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	RN01
RF02	O sistema deve permitir que a proprietária altere sua senha de acesso	Alta	RN01
RF03	O sistema deve permitir que a proprietária recupere sua senha via <i>e-mail</i>	Alta	RN01
RF04	O sistema deve permitir o cadastro de novas reservas, desde que associadas um quarto e a um período (data de check-in e check-out) disponíveis (data de <i>check-in</i> e <i>check-out</i>)	Alta	RN04; RN14; RN15; RN16
RF05	O sistema deve permitir a edição de reservas já cadastradas	Alta	RN08
RF06	O sistema deve impedir o cadastro de reservas que não estejam associadas a quartos disponíveis	Alta	RN10; RN16
RF07	O sistema deve exigir os dados pessoais do hóspede para que a reserva seja cadastrada: nome completo, endereço completo, CPF/Passaporte, telefone e <i>e-mail</i>	Alta	RN02
RF08	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	RN05; RN07

Fonte: Elaborado pelos autores.

Quadro 15 – Requisitos Funcionais (Parte 2)

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF09	O sistema deve permitir o registro de pagamento da reserva (forma, valor, data)	Média	RN05;RN06
RF10	A proprietária deve conseguir cancelar ou alterar uma reserva, com possível registro do motivo	Média	RN08
RF11	A proprietária deve conseguir cadastrar mais de uma reserva no nome de um mesmo hóspede	Média	RN03
RF12	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	RN10; RN16
RF13	A proprietária deve conseguir acessar o histórico de reservas de um hóspede	Média	RN02; RN03
RF14	O sistema deve permitir configurar tarifas de reserva com base nas regras de negócio da pousada	Média	RN13
RF15	O sistema deve enviar uma notificação para o hóspede via <i>e-mail</i> após a confirmação da reserva	Baixa	RN07
RF16	O sistema deve impedir o registro de reservas com menos de 2 dias de antecedência da data do <i>check-in</i>	Média	RN04
RF17	O sistema deve permitir a edição dos dados de hóspedes já cadastrados	Média	RN02
RF18	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	RN14; RN15

Fonte: Elaborado pelos autores.

Quadro 15 – Requisitos Funcionais (Parte 3)

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF19	A proprietária deve conseguir editar as informações dos quartos já cadastrados	Alta	RN14; RN15
RF20	O sistema deve permitir a visualização dos quartos disponíveis no período de tempo selecionado para a reserva	Média	RN14; RN16
RF21	A proprietária deve conseguir mudar o status de um quarto (ex: disponível, ocupado, em manutenção) manualmente, se necessário	Alta	RN09
RF22	O sistema deve gerar relatórios de ocupação de quartos em períodos definidos	Média	RN03; RN14; RN07; RN15; RN16
RF23	O sistema deve mudar o status do quarto quando for realizado o registro do <i>check-out</i>	Alta	RN10
RF24	O sistema deve mudar o status do quarto de reservado para ocupado no registro do <i>check-in</i>	Alta	RN10
RF25	O sistema deve permitir à proprietária configurar os horários padrão de <i>check-in</i> e <i>check-out</i>	Alta	RN10
RF26	O sistema deve permitir a geração de recibos ou comprovantes de pagamento e estadia, sob demanda	Média	RN11
RF27	O sistema deve permitir o registro de <i>early check-in</i> e <i>late check-out</i> para uma reserva	Média	RN12
RF28	O sistema deve permitir a configuração de tarifas adicionais para textitearly <i>check-in</i> e <i>late check-out</i>	Média	RN12; RN13
RF29	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	RN01

Fonte: Elaborado pelos autores.

Quadro 15 – Requisitos Funcionais (Parte 4)

Código	Descrição	Prioridade	Regra de Negócio Relacionada
RF30	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	RN01
RF31	O sistema deve permitir a edição das transações financeiras registradas (receitas e despesas)	Média	RN01
RF32	O sistema deve permitir a exclusão das transações financeiras registradas	Média	RN01
RF33	O sistema deve permitir que a proprietária visualize todas as transações financeiras (receitas e despesas) em um determinado período	Média	RN01
RF34	O sistema deve permitir a filtragem das transações por tipo (receita/despesa), data e categoria	Média	RN01
RF35	O sistema deve gerar relatórios financeiros detalhados automaticamente por período e por categoria	Média	RN01
RF36	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	RN01
RF37	O sistema deve gerar relatórios de faturamento por período	Média	RN01
RF38	O sistema deve apresentar um painel (dashboard) com métricas-chave da pousada	Média	Requisito essencial para a gestão e visualização do negócio
RF39	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.

APÊNDICE F – Certificados de Segurança

F.1 Certificado SSL

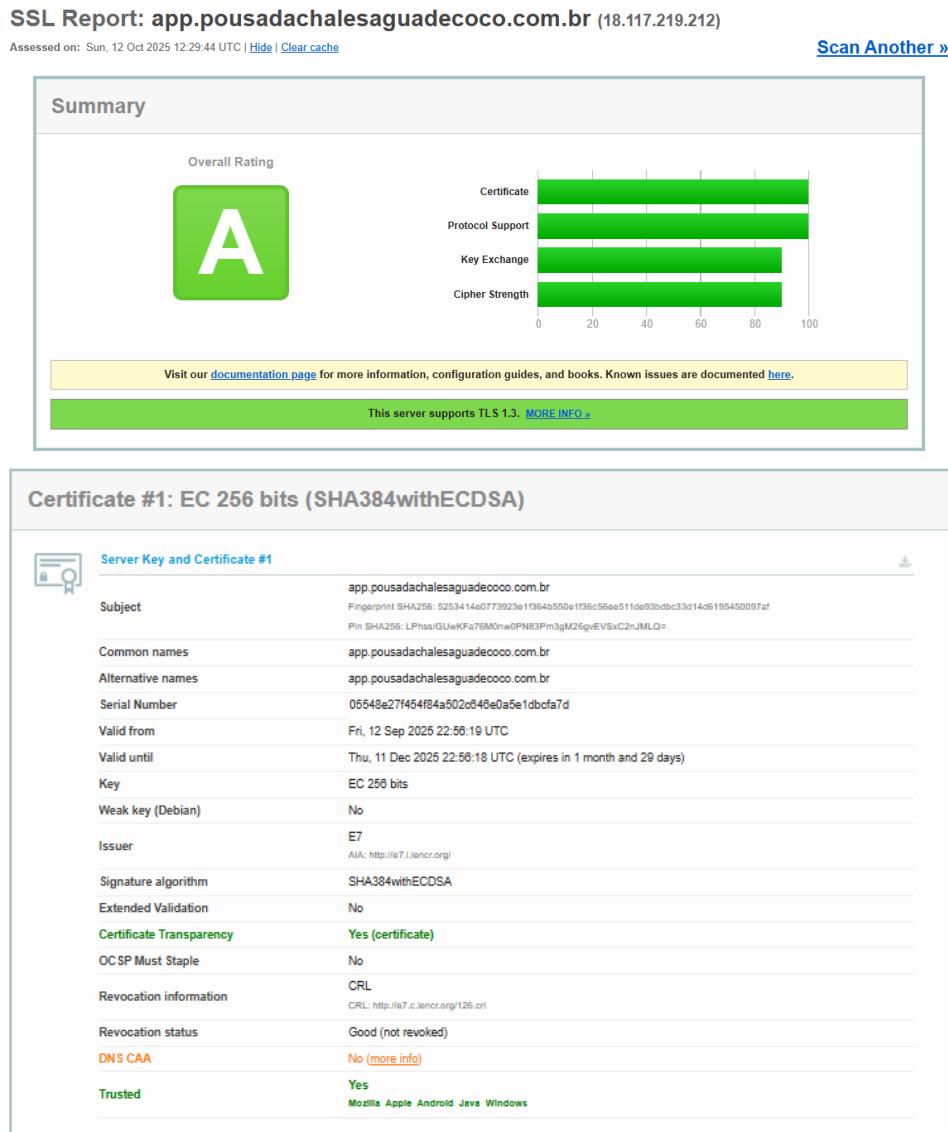


Figura 10 – Resultado do teste SSL Server Test

Fonte: Elaboração própria.

F.2 Certificado Headers de Segurança

The screenshot shows the results of a security header audit for the website <https://app.pousadachalesaiguadeccoco.com.br/accounts/login/>. The report is from November 14, 2025, at 18:11:33 UTC. The overall grade is A+. The audit found several issues:

- Warnings:** X-Frame-Options (Duplicate header), X-Content-Type-Options (Duplicate header).
- Raw Headers:** HTTP/1.1, 200 OK, Server: nginx.

Checkmarks indicate the presence of secure headers: X-Frame-Options, X-Content-Type-Options, Referer-Policy, Strict-Transport-Security, Permissions-Policy, Content-Security-Policy.

Figura 11 – Resultado do teste de Headers de Segurança

Fonte: Elaboração própria.

APÊNDICE G – Relatório de Testes Funcionais - Módulo de Gestão de Hóspedes

- **Data da Execução:** 22 de setembro de 2025
- **Testador:** Guilherme Bittencourt Schmidt

ID do Teste: HOS-001

Funcionalidade	Criar Hóspede
Cenário de Teste	Criar um novo hóspede com todos os dados válidos
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em Hospede.2. Clicar em "Adicionar Hóspede".3. Preencher todos os campos com dados válidos (CPF, nome completo, telefone, email e data de nascimento.).4. Clicar em "Adicionar".
Resultado Esperado	O sistema deve salvar o hóspede, redirecionar para a lista e o novo hóspede deve aparecer na tabela.
Status	Passou
Observações	Fluxo de criação principal funcionando corretamente.

ID do Teste: HOS-002

Funcionalidade	Criar Hóspede
Cenário de Teste	Tentar criar hóspede com campo obrigatório (nome) vazio
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em Hospede.2. Clicar em "Adicionar Hóspede".3. Preencher todos os campos, exceto o "Nome Completo".4. Clicar em "Adicionar".
Resultado Esperado	O sistema deve exibir uma mensagem de erro clara abaixo do campo "Nome Completo". O hóspede não deve ser criado.
Status	Passou
Observações	O hóspede não foi criado (correto), e a pagina exibiu uma mensagem de erro abaixo do campo Nome Completo, com a seguinte frase "Preencha este campo".

ID do Teste: HOS-003

Funcionalidade	Visualizar Hóspede
Cenário de Teste	Visualizar detalhes de um hóspede existente
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em Hospede.2. Clicar em "Listar Hóspedes".3. Na lista de hóspedes, clicar no nome de um hóspede cadastrado.
Resultado Esperado	A página de detalhes do hóspede deve ser exibida, mostrando todas as informações corretas (CPF, nome, telefone, email e data de nascimento.).
Status	Passou
Observações	A página de detalhes do hóspede carrega corretamente.

ID do Teste: HOS-004

Funcionalidade	Atualizar Hóspede
Cenário de Teste	Editar o telefone de um hóspede
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em Hospede.2. Clicar em "Listar Hóspedes".3. Na lista de hóspedes, clicar no nome de um hóspede cadastrado.4. Clicar em "Editar".5. Alterar o número de telefone.6. Clicar em "Salvar".
Resultado Esperado	O sistema deve salvar a alteração e, na lista, o novo telefone deve ser exibido.
Status	Passou
Observações	A alteração foi realizada corretamente.

ID do Teste: HOS-005

Funcionalidade	Buscar Hóspede
Cenário de Teste	Buscar por um hóspede existente e um inexistente
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em Hospede.2. Clicar em "Listar Hóspedes".3. Na lista de hóspedes, usar a barra de busca para procurar por parte do nome de um hóspede cadastrado.4. Verificar o resultado.5. Limpar a busca e procurar por um nome que não existe (ex: "Zzzzz").
Resultado Esperado	Na primeira busca, a lista deve ser filtrada corretamente. Na segunda, a lista deve ficar vazia e exibir uma mensagem de "Nenhum hóspede encontrado".
Status	Passou
Observações	Busca efetuada com sucesso.

ID do Teste: HOS-006

Funcionalidade	Excluir Hóspede
Cenário de Teste	Excluir um hóspede de forma permanente
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em Hospede.2. Clicar em "Listar Hóspedes".3. Clicar no botão "Excluir"de um hóspede na lista.4. O modal de confirmação deve aparecer.5. Clicar no botão "Excluir"dentro do modal.
Resultado Esperado	O hóspede deve ser removido da lista e do banco de dados.
Status	Passou
Observações	O modal de confirmação funcionou corretamente, prevenindo exclusão acidental.

APÊNDICE H – Relatório de Testes Funcionais - Módulo de Gestão de Quartos

- **Data da Execução:** 29 de setembro de 2025
- **Testador:** Guilherme Bittencourt Schmidt

ID do Teste: QUA-001

Funcionalidade	Criar Quarto
Cenário de Teste	Criar um novo quarto com status "Disponível"
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Quartos".2. Clicar no botão "Adicionar Quarto".3. Preencher todos os campos.4. Clicar em "Adicionar".
Resultado Esperado	O sistema deve salvar o quarto e redirecionar para a lista. O novo quarto deve aparecer com o status "Disponível".
Status	Passou
Observações	Ao clicar em "Adicionar", o sistema redirecionou para a lista de quartos, com o status certo.

ID do Teste: QUA-002

Funcionalidade	Criar Quarto
Cenário de Teste	Tentar salvar sem preencher campos obrigatórios
Passos para Execução	
	<ol style="list-style-type: none"> 1. Clicar em "Quartos". 2. Clicar no botão "Adicionar Quarto". 3. Deixar algum campo em branco e clicar em "Adicionar".
<hr/>	
Resultado Esperado	Deve exibir uma mensagem abaixo do campo, pedindo para preencher o campo.
<hr/>	
Status	Passou
<hr/>	
Observações	A validação de campos obrigatórios está funcionando. O a mensagem foi exibida corretamente.

ID do Teste: QUA-003

Funcionalidade	Visualizar Quartos
Cenário de Teste	Listagem de quartos com status diferentes
Passos para Execução	
	<ol style="list-style-type: none"> 1. Clicar em "Quartos". 2. Clicar em "Listar Quartos". 3. Garantir que existam quartos com status "Disponível" e "Indisponível".
<hr/>	
Resultado Esperado	A tabela deve exibir todos os quartos. A coluna "Status" deve mostrar "Disponível" e "Indisponível" para os quartos.
<hr/>	
Status	Passou
<hr/>	
Observações	Aparece todos os quartos e todos com seus respectivos status corretos.

ID do Teste: QUA-004

Funcionalidade	Atualizar Quarto
Cenário de Teste	Editar o preço de um quarto existente
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Quartos".2. Clicar em "Listar Quartos".3. Clicar em "Editar" no quarto escolhido.4. Alterar um campo do quarto.5. Clicar em "Salvar Alterações".
Resultado Esperado	O sistema salva a alteração, redireciona para a lista e o novo preço é exibido na tabela.
Status	Passou
Observações	O fluxo de edição está funcionando normalmente.

ID do Teste: QUA-005

Funcionalidade	Excluir Quarto
Cenário de Teste	Excluir um quarto de forma permanente
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Quartos".2. Clicar em "Listar Quartos".3. Clicar em "Excluir" no quarto escolhido.4. No modal de confirmação, clicar em "Sim, excluir".
<hr/>	
Resultado Esperado	O quarto deve ser removido da lista permanentemente.
Status	Passou
Observações	Exclusão feita com sucesso.

APÊNDICE I – Relatório de Testes Funcionais - Módulo de Gestão de Reservas

- **Data da Execução:** 29 de setembro de 2025
- **Testador:** Guilherme Bittencourt Schmidt

ID do Teste: RES-001

Funcionalidade	Criar Reserva
Cenário de Teste	Criar uma nova reserva com dados válidos
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Listar Reservas".3. Clicar em "Adicionar Reserva".4. Selecionar hóspede e quarto.5. Escolher data de início e fim.6. Clicar em "Reservar".
Resultado Esperado	O sistema salva e redireciona para a lista. A nova reserva nela.
Status	Passou
Observações	Tudo ocorreu perfeitamente.

ID do Teste: RES-002

Funcionalidade	Criar Reserva
Cenário de Teste	Tentar criar com data de início no passado
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Listar Reservas".3. Clicar em "Adicionar Reserva".4. Selecionar hóspede e quarto.5. Escolher data de início e fim antigas.6. Clicar em "Reservar".
Resultado Esperado	O sistema exibe uma mensagem de erro abaixo do campo de data. A reserva não é criada.
Status	Passou
Observações	Mensagem de erro exibida corretamente.

ID do Teste: RES-003

Funcionalidade	Criar Reserva
Cenário de Teste	Tentar criar com data de fim antes do início
Passos para Execução	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Listar Reservas".3. Clicar em "Adicionar Reserva".4. Selecionar hóspede e quarto.5. Escolher data de início e fim mais antiga que a data de início.6. Clicar em "Reservar".
Resultado Esperado	O sistema exibe mensagem de erro abaixo do campo de data de fim. A reserva não é criada.
Status	Passou
Observações	Aparece a mensagem de erro corretamente.

ID do Teste: RES-004

Funcionalidade	Visualizar Reservas
Cenário de Teste	Filtrar check-ins para hoje
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Listar Reservas".3. Criar/editar uma reserva com data de início do dia atual.4. Acessar a página "Check-ins de Hoje".
Resultado Esperado	Apenas a reserva com data de início para hoje deve aparecer na lista.
Status	Passou
Observações	Filtro funcionou como esperado.

ID do Teste: RES-005

Funcionalidade	Visualizar Reservas
Cenário de Teste	Filtrar check-outs para hoje
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Listar Reservas".3. Criar/editar uma reserva com data de fim do dia atual.4. Acessar a página "Check-outs de Hoje".
Resultado Esperado	Apenas a reserva com data de fim para hoje deve aparecer na lista.
Status	Passou
Observações	Filtro funcionou como esperado.

ID do Teste: RES-006

Funcionalidade	Ações
Cenário de Teste	Marcar Check-in de uma reserva
Passos para Execução	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Check-ins de Hoje".3. Clicar no botão "Check-in" de uma reserva.4. Confirmar a ação.
Resultado Esperado	O botão muda para "Check-in marcado" e o campo de check-in na lista principal é preenchido com a data/hora.
Status	Passou
Observações	Ação de check-in e atualização ocorreram com sucesso.

ID do Teste: RES-007

Funcionalidade	Ações
Cenário de Teste	Marcar Check-out de uma reserva
Passos para Execução	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Check-outs de Hoje".3. Clicar no botão "Check-out" de uma reserva.4. Confirmar a ação.
Resultado Esperado	O botão muda de estado e o campo de check-out na lista principal é preenchido.
Status	Passou
Observações	Funcionalidade OK.

ID do Teste: RES-008

Funcionalidade	Excluir Reserva
Cenário de Teste	Excluir uma reserva existente
Passos para Execução	
	<ol style="list-style-type: none">1. Clicar em "Reservas".2. Clicar em "Listar Reservas".3. Na lista principal, clicar para cancelar uma reserva.4. Confirmar o cancelamento no modal/página de confirmação.
Resultado Esperado	A reserva é removida da lista de forma permanente.
Status	Passou
Observações	Modal de confirmação funcionou, porém a reserva não é removida da lista, fica como "Cancelada".

APÊNDICE J – Relatório de Testes

Funcionais - Financeiro

- **Data da Execução:** 29 de outubro de 2025
- **Testador:** Guilherme Bittencourt Schmidt

ID do Teste: CT-FN01

Funcionalidade	Registro de Transação (RF19)
Cenário de Teste	Registrar uma nova despesa manual (Título de Saída)
Passos para Execução	
	<ol style="list-style-type: none">1. Acessar "Títulos Financeiros" e clicar em "Adicionar Título".2. Preencher: Descrição (ex: "Material de Limpeza"), Valor (ex: 150,00), Data Vencimento.3. Selecionar a Categoria (ex: "Limpeza").4. No campo "Tipo", selecionar a opção "Saída".5. Clicar em "Adicionar Título".
Resultado Esperado	O sistema salva e redireciona para a lista. O novo título de Saída (Despesa) deve aparecer na tabela.
Status	Passou
Observações	Fluxo principal de registro de despesas funcionando.

ID do Teste: CT-FN02

Funcionalidade	Edição de Transação (RF19)
Cenário de Teste	Editar o valor e a categoria de uma despesa existente
Passos para Execução	
	<ol style="list-style-type: none">1. Na lista de Títulos, localizar a despesa criada no CT-FN01.2. Clicar no botão "Editar".3. Alterar o Valor para "175,00" e a Categoria para "Manutenção".4. Clicar em "Salvar Alterações".
Resultado Esperado	A transação deve ser atualizada na lista, exibindo os novos valores de Categoria e Valor.
Status	Passou
Observações	A edição de títulos está funcionando como esperado.

ID do Teste: CT-FN03

Funcionalidade	Cancelamento de Transação (RF19)
Cenário de Teste	Cancelar uma despesa (em vez de excluir)
Passos para Execução	
	<ol style="list-style-type: none">1. Na lista de Títulos, localizar a despesa do teste CT-FN02.2. Clicar no botão "Editar".3. No formulário, encontrar o campo "Cancelado?".4. Mudar a seleção do rádio de "Não" para "Sim".5. Clicar em "Salvar Alterações".
Resultado Esperado	A transação deve ser salva. Na lista de títulos, o filtro "Cancelado=Não"(padrão) deve esconder este título. Ao filtrar por "Cancelado=Sim", ele deve aparecer.
Status	Passou
Observações	BUG-003: Funcionando.

ID do Teste: CT-FN04

Funcionalidade	Registro de Receita (RF20)
Cenário de Teste	Registrar receita vinculada a uma reserva existente
Passos para Execução	
	<ol style="list-style-type: none">1. Acessar "Adicionar Título".2. Preencher: Descrição (ex: "Pagamento Reserva 101"), Valor (ex: 400,00).3. No campo "Reserva", selecionar uma reserva da lista.4. No campo "Tipo", selecionar a opção "Entrada".5. Clicar em "Adicionar Título".
Resultado Esperado	A receita (Entrada) deve ser registrada. Na lista de títulos, a coluna "Hóspede" deve exibir o nome do hóspede associado àquela reserva.
Status	Passou
Observações	A vinculação entre Título e Reserva está funcionando.

ID do Teste: CT-FN05

Funcionalidade	Filtros de Transações (RF21)
Cenário de Teste	Filtrar transações por tipo, data e categoria
Passos para Execução	
	<ol style="list-style-type: none">1. Garantir que a lista contenha a Receita (CT-FN04) e a Despesa (CT-FN02).2. Usar o filtro "Tipo", selecionar "Saída" e clicar em "Filtrar". (Verificar se a Receita some).3. Limpar o filtro. Usar o filtro "Vencimento" para uma data no passado. (Verificar se os títulos de hoje somem).4. Tentar localizar o filtro de "Categoria" na tela.
Resultado Esperado	Os filtros de tipo e data devem funcionar. O filtro de categoria também deve estar disponível e funcional.
Status	Passou
Observações	Funcionou.

ID do Teste: CT-FN06

Funcionalidade	Pagamento de Título (RF20)
Cenário de Teste	Marcar título de receita como pago e verificar status da reserva
Passos para Execução	
	<ol style="list-style-type: none">1. Localizar a Receita (Entrada) criada no CT-FN04, que está vinculada a uma reserva.2. Verificar se a "Situação" do título é "Aberto" ou "Vencido".3. Clicar no botão "Marcar como Pago".4. Confirmar a ação no pop-up do navegador.
Resultado Esperado	A lista de Títulos deve ser recarregada. A "Situação" do título deve mudar para "Pago".
Status	Passou
Observações	Pagamento e confirmação de reserva funcionando.