

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 – Diagrama de Implantação desenvolvido no Online Visual-Paradigm .	56
Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm .	57
Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo .	64
Figura 4 – Diagrama Entidade-Relacionamento (DER) desenvolvido no brModelo	64

# Lista de quadros

Quadro 1 – Comparativo de funcionalidades entre Chalés Água de Coco e concorrentes . . . . .	16
Quadro 2 – Divisão das Funções e Responsabilidades da Equipe do Projeto . . .	20
Quadro 3 – Product Backlog - Parte 1 . . . . .	22
Quadro 4 – Product Backlog - Parte 2 . . . . .	23
Quadro 5 – Sprints Backlog - Parte 1 . . . . .	24
Quadro 6 – Sprints Backlog - Parte 2 . . . . .	25
Quadro 7 – Identificação dos Riscos do Projeto - Parte 1 . . . . .	27
Quadro 8 – Identificação dos Riscos do Projeto - Parte 2 . . . . .	28
Quadro 9 – Análise e Planejamento dos Riscos - Parte 1 . . . . .	29
Quadro 10 – Análise e Planejamento dos Riscos - Parte 2 . . . . .	30
Quadro 11 – Análise e Planejamento dos Riscos - Parte 3 . . . . .	31
Quadro 12 – Mecanismos de Monitoramento dos Riscos - Parte 1 . . . . .	31
Quadro 13 – Mecanismos de Monitoramento dos Riscos - Parte 2 . . . . .	32
Quadro 14 – Questionário Aplicado à Proprietária – Parte 1 . . . . .	34
Quadro 15 – Regras de Negócio - Parte 2 . . . . .	35
Quadro 16 – Requisitos Funcionais - Parte 1 . . . . .	36
Quadro 17 – Requisitos Funcionais - Parte 2 . . . . .	37
Quadro 18 – Requisitos Funcionais - Parte 3 . . . . .	38
Quadro 19 – Requisitos Não Funcionais - Parte 1 . . . . .	39
Quadro 20 – Requisitos Não Funcionais - Parte 2 . . . . .	40
Quadro 21 – Questionário Aplicado à Proprietária – Parte 1 . . . . .	84
Quadro 22 – Questionário Aplicado à Proprietária – Parte 2 . . . . .	85
Quadro 23 – Questionário Aplicado à Proprietária – Parte 3 . . . . .	86
Quadro 24 – Questionário Aplicado à Proprietária – Parte 3 . . . . .	87
Quadro 25 – Product Backlog - Parte 1 . . . . .	89
Quadro 26 – Product Backlog - Parte 2 . . . . .	90
Quadro 27 – Product Backlog - Parte 3 . . . . .	91
Quadro 28 – Product Backlog - Parte 4 . . . . .	92
Quadro 29 – Requisitos Funcionais - Parte 1 . . . . .	93
Quadro 30 – Requisitos Funcionais - Parte 2 . . . . .	94
Quadro 31 – Requisitos Funcionais - Parte 3 . . . . .	95
Quadro 32 – Requisitos Funcionais - Parte 4 . . . . .	96

# Lista de tabelas

Tabela 1 – Resumo dos custos estruturais . . . . .	70
Tabela 2 – Quantidade e horas trabalhadas por função . . . . .	71
Tabela 3 – Custos por função . . . . .	71
Tabela 4 – Custo Total por Categoria . . . . .	72

# Lista de abreviaturas e siglas

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



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Objetivos</b>	<b>13</b>
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	13
<b>1.2</b>	<b>Justificativa</b>	<b>13</b>
<b>1.3</b>	<b>Análise da Concorrência</b>	<b>14</b>
1.3.1	SimplesHotel	15
1.3.2	HospedaJá	15
1.3.3	Diferenciais do Chalés Água de Coco	15
1.3.4	Comparativo	16
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>17</b>
2.0.1	Histórico do Turismo e da Hospitalidade	17
2.0.2	A Gestão Hoteleira e o Impacto da Tecnologia	18
<b>3</b>	<b>GESTÃO DO PROJETO</b>	<b>19</b>
<b>3.1</b>	<b>Metodologia de Gestão e Desenvolvimento</b>	<b>19</b>
<b>3.2</b>	<b>Artefatos</b>	<b>20</b>
3.2.1	Product Backlog	20
3.2.2	Sprint Backlog	23
<b>3.3</b>	<b>Gestão de Riscos</b>	<b>25</b>
3.3.1	Identificação dos Riscos do Projeto	26
3.3.2	Análise e Planejamento dos Riscos	28
3.3.3	Monitoramento dos Riscos	31
<b>4</b>	<b>DESENVOLVIMENTO DO PROJETO</b>	<b>33</b>
<b>4.1</b>	<b>Escopo do Projeto</b>	<b>33</b>
4.1.1	Funcionalidades do Sistema	33
4.1.2	Regras de Negócio	34
4.1.3	Requisitos do Sistema	35
4.1.3.1	Requisitos Funcionais	35
4.1.3.2	Requisitos Não Funcionais	38
<b>4.2</b>	<b>Histórias de Usuário</b>	<b>41</b>
4.2.1	Descrição das Histórias de Usuário	41
<b>4.3</b>	<b>Arquitetura</b>	<b>53</b>
4.3.1	Arquitetura Lógica	54

4.3.1.1	Camada de Apresentação (Template) . . . . .	54
4.3.1.2	Camada de Controle (View) . . . . .	54
4.3.1.3	Camada de Modelo (Model) . . . . .	54
4.3.2	Banco de Dados . . . . .	54
4.3.3	Infraestrutura em Nuvem . . . . .	55
4.3.4	Preparação para Evoluções Futuras . . . . .	55
4.3.5	Diagrama da Arquitetura . . . . .	55
4.3.5.1	Diagrama de Implantação . . . . .	55
4.3.5.2	Diagrama de Componentes . . . . .	56
<b>4.4</b>	<b>Tecnologias . . . . .</b>	<b>57</b>
4.4.1	Django . . . . .	57
4.4.1.1	Front-end . . . . .	58
4.4.1.2	Back-end . . . . .	58
4.4.1.3	Banco de Dados . . . . .	58
4.4.2	Justificativa da Escolha . . . . .	58
4.4.2.1	Justificativa Técnica . . . . .	58
4.4.2.2	Infraestrutura com AWS . . . . .	59
<b>4.5</b>	<b>Ferramentas de Apoio . . . . .</b>	<b>59</b>
4.5.1	GitHub . . . . .	59
4.5.2	BRModelo . . . . .	60
4.5.3	Visual Paradigm Online . . . . .	60
4.5.4	Latex . . . . .	60
4.5.5	Google Meet . . . . .	60
4.5.6	ProjectLibre . . . . .	60
<b>4.6</b>	<b>Manutenibilidade . . . . .</b>	<b>61</b>
<b>4.7</b>	<b>Segurança, Privacidade e Legislação . . . . .</b>	<b>61</b>
4.7.1	Segurança da Aplicação . . . . .	62
4.7.2	Segurança na Comunicação . . . . .	62
4.7.3	Conformidade com a LGPD . . . . .	63
<b>4.8</b>	<b>Modelagem do Banco de Dados . . . . .</b>	<b>63</b>
4.8.1	Modelo Entidade-Relacionamento - MER . . . . .	64
4.8.2	Diagrama Entidade-Relacionamento - DER . . . . .	64
<b>4.9</b>	<b>Repositório da Aplicação . . . . .</b>	<b>64</b>
4.9.1	Definição do repositório da aplicação . . . . .	65
<b>4.10</b>	<b>Entregas . . . . .</b>	<b>67</b>
4.10.1	Desenvolvimento de um Tema - 08/04 . . . . .	67
4.10.2	Desenho da Aplicação - 29/04 . . . . .	67
4.10.3	Prova de Conceito - 20/05 . . . . .	67
4.10.4	Análise e Documentação - 10/06 . . . . .	67

4.10.5	Produto Mínimo Viável - 24/06 . . . . .	68
<b>5</b>	<b>VIABILIDADE FINANCEIRA . . . . .</b>	<b>69</b>
<b>5.1</b>	<b>Custos . . . . .</b>	<b>69</b>
5.1.1	Custo Estrutural . . . . .	69
5.1.2	Custo de Pessoal . . . . .	70
5.1.3	Custo Total . . . . .	71
<b>6</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>73</b>
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>74</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>75</b>
	<b>APÊNDICES . . . . .</b>	<b>79</b>
	<b>APÊNDICE A – DIÁRIO DE BORDO . . . . .</b>	<b>80</b>
A.1	1º SEMANA . . . . .	80
A.2	2º SEMANA . . . . .	80
A.3	3º SEMANA . . . . .	80
A.4	4º SEMANA . . . . .	81
A.5	5º SEMANA . . . . .	81
A.6	6º SEMANA . . . . .	81
A.7	7º SEMANA . . . . .	82
A.8	8º SEMANA . . . . .	82
A.9	9º SEMANA . . . . .	82
A.10	10º SEMANA . . . . .	82
A.11	11º SEMANA . . . . .	82
A.12	12º SEMANA . . . . .	83
A.13	13º SEMANA . . . . .	83
	<b>APÊNDICE B – QUESTIONÁRIO APLICADO À PROPRIETÁRIA DA POUSADA . . . . .</b>	<b>84</b>
	<b>APÊNDICE C – PRODUCT BACKLOG DETALHADO . . . . .</b>	<b>89</b>
	<b>APÊNDICE D – REQUISITOS FUNCIONAIS DETALHADOS . . . . .</b>	<b>93</b>
	<b>APÊNDICE E – RELATÓRIO DE TESTES FUNCIONAIS - MÓ- DULO DE GESTÃO DE HÓSPEDES . . . . .</b>	<b>97</b>

APÊNDICE F – RELATÓRIO DE TESTES FUNCIONAIS - MÓ- DULO DE GESTÃO DE QUARTOS . . . . .	103
--	-----

APÊNDICE G – RELATÓRIO DE TESTES FUNCIONAIS - MÓ- DULO DE GESTÃO DE RESERVAS . . . . .	109
---	-----

# 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 a partir das necessidades da Pousada Chalés Água de Coco. Atualmente, a gestão é feita em planilhas, o que dificulta a organização e aumenta a chance de erros. Assim, este projeto busca centralizar as informações em um sistema web, tornando os processos mais ágeis e confiáveis.

### 1.1.1 Objetivo Geral

Desenvolver um sistema *web* para automatizar os processos administrativos da pousada Chalés Água de Coco, atualmente realizados por meio de planilhas eletrônicas. A adoção de ferramentas digitais em pequenos empreendimentos do setor de hospitalidade tem se mostrado fundamental para aumentar a eficiência e reduzir falhas operacionais (TOTVS, 2024). 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, foram definidos os seguintes objetivos específicos:

- Implementar funcionalidades para o gerenciamento de hóspedes, reservas e acomodações;
- Desenvolver um módulo de controle financeiro que permita registrar receitas e despesas;
- Proporcionar maior organização e acessibilidade às informações administrativas;
- Reduzir a ocorrência de falhas decorrentes do uso de planilhas manuais;
- Disponibilizar relatórios gerenciais que apoiem a tomada de decisão dos gestores.

## 1.2 Justificativa

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

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.

Além da relevância prática, o desenvolvimento deste projeto apresenta também uma dimensão acadêmica. A proposta permite a aplicação prática dos conhecimentos adquiridos ao longo do curso, abrangendo áreas como análise de requisitos, modelagem de dados, programação web, computação em nuvem, experiência do usuário e segurança da informação. Segundo Turban, Pollard e Wood (TURBAN; POLLARD; WOOD, 2018), projetos que envolvem o desenvolvimento de sistemas de informação proporcionam uma visão integrada dos desafios tecnológicos e de gestão, favorecendo a formação de profissionais mais preparados para atuar no mercado.

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 *NFSe*, motor de reservas e envio de *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

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.

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

## 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.

### 2.0.1 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.

### 2.0.2 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).

Diante 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.

Integrante	Função	Papel Scrum	Responsabilidades
Guilherme Akio	Analista de QA	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	CPromover e facilitar a aplicação da teoria e das práticas do <i>Scrum</i> ; Definir e organizar a estrutura do sistema
Kelly Radchelle	Gerente de Projeto	Product Owner	Acompanhar o progresso das sprints; Facilitar e coordenar tomada de decisões e comunicação entre as partes interessadas <i>sprints</i>
Rafael Teixeira	Desenvolvedor Frontend	Time de Desenvolvimento	Entregas de funcionalidades relacionadas à experiência do usuário; Projetar e implementar interfaces da aplicação
Ricardo Carriel	Desenvolvedor Backend	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.

Quadro 2 – Divisão das Funções e Responsabilidades da Equipe do Projeto

## 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 está 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 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.

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 <i>Check-in</i> e <i>Check-out</i>	<i>Check-in</i> e <i>Check-out</i>	ALTA
12	Integrar ambientes <i>backend</i> e <i>frontend</i>	Requisito técnico	ALTA

Fonte: Elaborado pelos autores.

Quadro 3 – Product Backlog - Parte 1

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 pousada para uso da aplicação	Implantação	MÉDIA

Fonte: Elaborado pelos autores.

#### Quadro 4 – Product Backlog - Parte 2

### 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. E está registrado no [Quadro 5](#).

As *sprints* foram organizadas em períodos de quinze dias (duas semanas), intervalo em conformidade com o Guia do Scrum ([SCHWABER; SUTHERLAND, 2013](#)). Totalizando 12 semanas de trabalho. Essa definição considerou a atuação majoritariamente remota da equipe e o tempo diário individual limitado para o trabalho, circunstâncias que demandaram períodos mais extensos para a execução das tarefas. Além disso, a organização



das tarefas dentro de cada *sprint* levou em consideração as datas de entrega do MVP, da documentação técnica e do produto final do projeto.

Diante disso, os itens do *Product Backlog* foram distribuídos de forma a cumprir um progresso gradual das atividades em duas fases. Na primeira fase, as primeiras *sprints* focaram na modelagem do sistema e escolha das ferramentas e tecnologias de gestão e desenvolvimento. Enquanto, nas próximas *sprints*, foram iniciadas tarefas relacionadas ao desenvolvimento do MVP e documentação do sistema. Ao passo que, na segunda fase, a equipe focou em desenvolver as funcionalidades de gestão financeira, integração com o MVP, realização de testes e revisão da documentação. Sendo que, as atividades de desenvolvimento foram distribuídas considerando: prioridade e relações de dependência.

Primeira fase			
Sprint	Período	Objetivo	Atividades
1	25/03/2025 a 08/04/2025	Planejamento Inicial do Projeto	Definição de ferramentas, metodologias, tecnologias e ambientes de trabalho.
2	09/04/2025 a 22/04/2025	Início da Modelagem do Sistema	Levantamento de requisitos e definição do escopo do projeto
3	23/04/2025 a 06/05/2025	Finalização da Modelagem do Sistema	Configuração dos ambientes de trabalho e criação dos diagramas faltantes.
4	07/05/2025 a 20/05/2025	Início do desenvolvimento do MVP	Desenvolvimento de funcionalidades CRUD de quartos.
5	21/05/2025 a 03/06/2025	Continuidade do desenvolvimento do MVP	Desenvolvimento de funcionalidades CRUD de hóspedes e reservas; Criação de interfaces do CRUD de quartos.
6	04/06/2025 a 24/06/2025	Finalização do desenvolvimento do MVP	Revisão da Documentação e criação das interfaces CRUDs de hóspedes e reservas

Fonte: Elaborado pelos autores.

Quadro 5 – Sprints Backlog - Parte 1

Segunda fase			
Sprint	Período	Objetivo	Atividades
7	12/08/2025 a 18/08/2025	Revisão e Ajustes no MVP	Revisão e padronização do código; Criação do plano inicial de testes
8	27/08/2025 a 09/09/2025	Início do desenvolvimento do módulo financeiro	Implementação de CRUDs financeiros e cálculos avançados
9	10/09/2025 a 23/09/2025	Finalização do desenvolvimento do módulo financeiro	Criação das interfaces dos CRUDs financeiros e implementação dos relatórios
10	24/09/2025 a 07/10/2025	Revisão do código	Correção de erros e planejamento de testes de integração
11	08/10/2025 a 21/10/2025	Execução teste gerais	Realização de testes de integração e segurança
12	22/10/2025 a 07/11/2025	Deploy do sistema	Correção de bugs e implantação final do sistema

Fonte: Elaborado pelos autores.

### Quadro 6 – Sprints Backlog - Parte 2

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. 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* do GitHub. 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*.

## 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 7).

<b>Código</b>	<b>Risco</b>	<b>Afeta</b>	<b>Descrição</b>
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 dominação do assunto

Fonte: Elaborado pelos autores.

Quadro 7 – Identificação dos Riscos do Projeto - Parte 1

Código	Risco	Afeta	Descrição
R07	Vazamento de dados de hóspedes	Negócio	O sistema armazena dados pessoais e sensíveis e não possui uma estrutura de segurança fortificada
R08	Custos do projeto subestimados	Projeto e Produto	Há custos adicionais não previstos no planejamento
R09	Risco de integração de módulos (Django)	Projeto e Produto	A equipe enfrenta dificuldades em integrar as entidades do sistema decorrente de um mau planejamento de comunicação entre elas, segundo a arquitetura Django (MVT)
R10	Problemas de segurança no ambiente de produção	Produto	A implantação do sistema expõe falhas de segurança não identificados durante o desenvolvimento

Fonte: Elaborado pelos autores.

Quadro 8 – Identificação dos Riscos do Projeto - Parte 2

### 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 9](#).

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 9 – Análise e Planejamento dos Riscos - Parte 1

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; Monitorar as atividades e obstáculos nas <i>sprints</i> de desenvolvimento para prever possíveis atrasos no cronograma	Reorganizar tarefas das <i>sprints</i> de desenvolvimento, priorizando a realização de tarefas associadas as funcionalidades essenciais do sistema
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

Fonte: Elaborado pelos autores.

Quadro 10 – Análise e Planejamento dos Riscos - Parte 2

R09	Alto	Média	Realizar validações dos diagramas e o MER do sistema antes e durante a fase de desenvolvimento do sistema	Revisar os diagramas e o MER; Identificar falhas de integração entre os módulos; Corrigir as falhas
R10	Alto	Média	Definir um plano de testes que abranja os diferentes aspectos e funcionalidades do sistema	Desconectar o sistema; identificar a origem do problema de segurança; corrigir e documentar o erro

Fonte: Elaborado pelos autores.

Quadro 11 – Análise e Planejamento dos Riscos - Parte 3

### 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 12](#), no qual estão os indicadores de risco, os mecanismos que devem ser utilizados na identificação de alterações, sendo que esses elementos devem ser monitorados a cada *sprint* realizada, durante as reuniões pós *sprint*.

Risco	Indicadores	Métricas
R01	- Time <i>scrum</i> frequentemente tem dúvidas sobre o que deve ser feito; - Solicitações frequentes de adição de novas funcionalidades.	- Número de dúvidas abertas durante as reuniões de desenvolvimento.
R02	- Constante adiamento de entregas; - Tempo de desenvolvimento de tarefas maior do que o previsto;	- Porcentagem de tarefas entregues por <i>sprint</i> .
R03	- Carência de backups; - Carência de testes de restauração.	- Frequência (tempo médio) de realização de <i>backups</i> . - Frequência (tempo médio) de falhas registradas nos testes de banco de dados.

Fonte: Elaborado pelos autores.

Quadro 12 – Mecanismos de Monitoramento dos Riscos - Parte 1



Risco	Indicadores	Métricas
R04	<ul style="list-style-type: none"> <li>- Desenvolvimento das funcionalidades está mais complexo que o esperado;</li> <li>- Aumento de tarefas não previstas nas <i>sprints backlogs</i>.</li> </ul>	- Número de novas tarefas adicionadas nas <i>sprints backlogs</i> .
R05	<ul style="list-style-type: none"> <li>- Tarefas sendo remanejadas constantemente para futuras <i>sprints</i>.</li> </ul>	- Número de remanejamento de tarefas por <i>sprints</i> .
R06	<ul style="list-style-type: none"> <li>- Carência de comunicação entre os membros do time;</li> <li>- Carência de documentação e <i>code review</i> para facilitar substituições em tarefas.</li> </ul>	- Número da concentração de tarefas por integrante.
R07	<ul style="list-style-type: none"> <li>- Dados sensíveis armazenados sem criptografia;</li> <li>- Carência de testes de segurança;</li> <li>- Códigos com permissões e bibliotecas mal configurados.</li> </ul>	<ul style="list-style-type: none"> <li>- Número do uso de dados sensíveis sem criptografia.</li> <li>- Número de inconformidades com a LGPD (<i>checklist</i>).</li> </ul>
R08	<ul style="list-style-type: none"> <li>- Gastos com ferramentas para funcionalidades não previstas.</li> </ul>	-Número de recursos contratados fora do planejamento inicial.
R09	<ul style="list-style-type: none"> <li>- Módulos desenvolvidos em paralelo sem o uso de práticas de integração contínua;</li> <li>- Carência de testes de integração;</li> <li>- Modelos mal padronizados.</li> </ul>	- Número de <i>bugs</i> relacionados à comunicação entre módulos.
R10	<ul style="list-style-type: none"> <li>- Carência de mecanismos como HTTPS e <i>headers</i> de segurança;</li> <li>- Dados sensíveis expostos em variáveis de ambiente.</li> </ul>	Número de falhas identificadas durante testes de segurança.

Fonte: Elaborado pelos autores.

### Quadro 13 – Mecanismos de Monitoramento dos Riscos - Parte 2

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 da Pousada Chalés Água de Coco, disponíveis no [Quadro 14](#).

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 14 – Questionário Aplicado à Proprietária – Parte 1

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.

#### Quadro 15 – Regras de Negócio - Parte 2

### 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 16](#). O quadro completo está disponível no [Apêndice D](#).

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

Fonte: Elaborado pelos autores.

Quadro 16 – Requisitos Funcionais - Parte 1

Código	Descrição	Prioridade	Regra de Negócio Relacionada
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
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	RN014; 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

Fonte: Elaborado pelos autores.

Quadro 17 – Requisitos Funcionais - Parte 2

Código	Descrição	Prioridade	Regra de Negócio Relacionada
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
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.

### Quadro 18 – Requisitos Funcionais - Parte 3

#### 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 19](#).

Código	Módulo	Descrição
RNF01	Usabilidade	A interface do sistema deve ser intuitiva, responsiva (compatível e adaptada tanto para dispositivos desktop quanto mobile) e de fácil utilização, de modo que as tarefas essenciais da gestão da pousada sejam realizadas de forma eficiente e com mínimo esforço de aprendizado pela proprietária. Para isso, deve-se adotar os princípios de interface amigável como a priorização da simplicidade e da clareza, padrões de interface consistentes e acessíveis
RNF02	Usabilidade	O sistema deve fornecer mensagens de feedback claras, objetivas e contextualizadas para todas as ações realizadas pela usuária, como confirmação de reserva (exemplo: reserva efetuada com sucesso) ou notificações de erros (exemplo: falha ao cadastrar um quarto), garantindo uma interação segura e satisfatória
RNF03	Performance	O sistema deve apresentar um tempo de resposta baixo, com carregamento das páginas e execução de ações da proprietária entre 2 e 3 segundos, para garantir uma navegação fluida.
RNF04	Segurança	O sistema deve garantir a segurança das informações da pousada e dos hóspedes através da implementação de mecanismos robustos de autenticação e autorização, de forma a assegurar que apenas a usuária autorizada consiga acessar ou alterar dados na aplicação.

Fonte: Elaborado pelos autores.

Quadro 19 – Requisitos Não Funcionais - Parte 1



Código	Módulo	Descrição
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.
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 deploy da aplicação deve ser realizado em uma infraestrutura de nuvem (Amazon EC2), a fim de proporcionar maior estabilidade, flexibilidade à aplicação e permitir que possíveis atualizações e manutenções tenham impacto mínimo para a usuária.
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, conseqüentemente, 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 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 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 1 mostra o funcionamento da arquitetura do sistema.



## Diagrama de Implantação

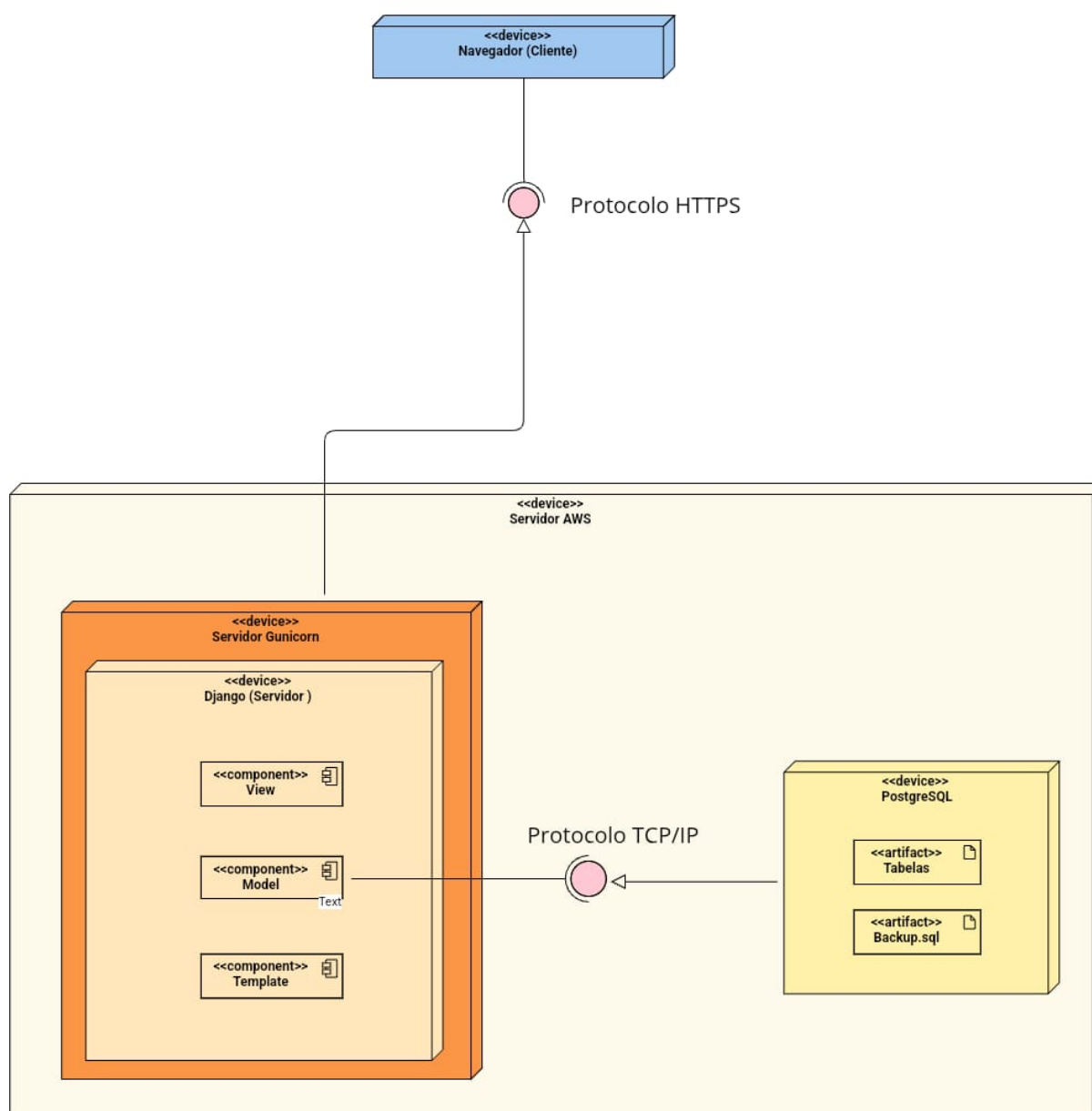


Figura 1 – Diagrama de Implantação desenvolvido no Online Visual-Paradigm

### 4.3.5.2 Diagrama de Componentes

A Figura 2 mostra o funcionamento da arquitetura do sistema.

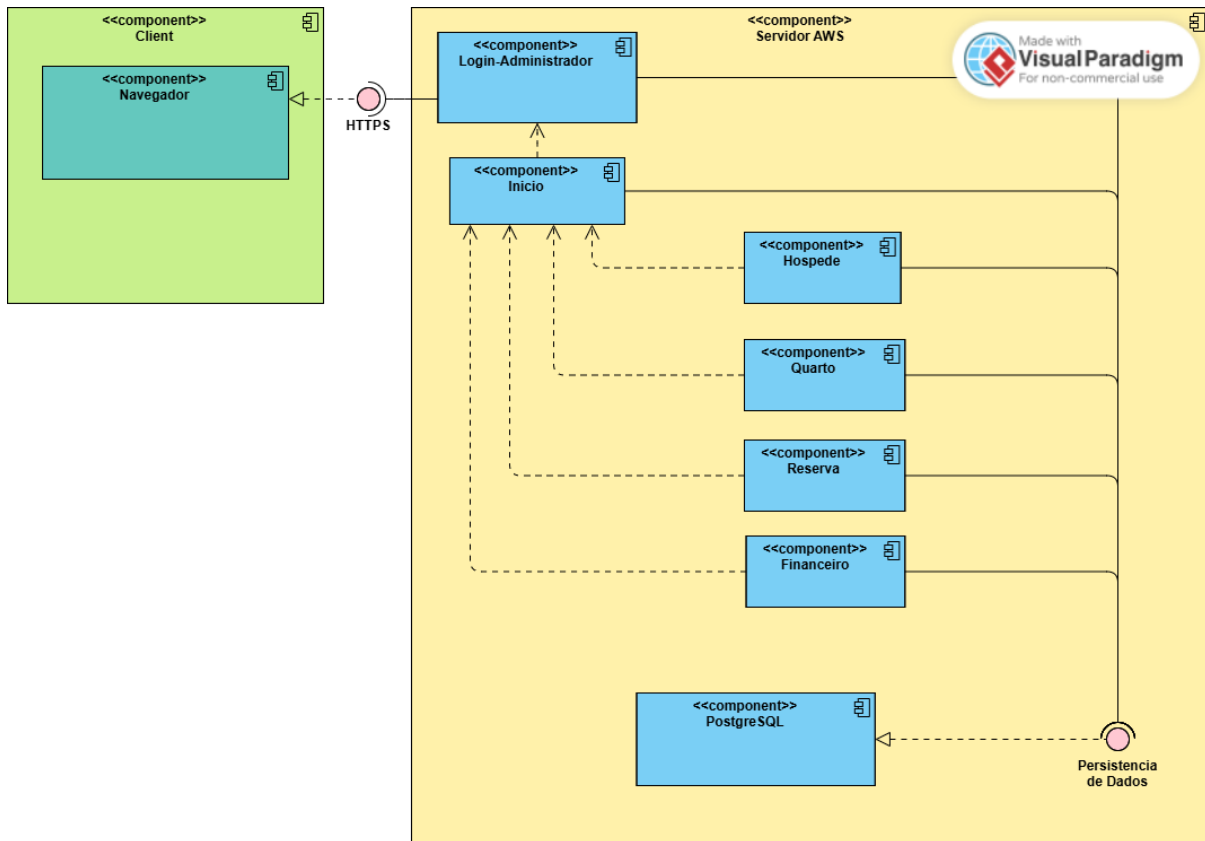


Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm

## 4.4 Tecnologias

O desenvolvimento do sistema da Pousada Chalés Água de Coco foi baseado na adoção de tecnologias modernas, acessíveis e amplamente utilizadas no mercado. A escolha dessas tecnologias teve como objetivo garantir robustez, escalabilidade, segurança e facilidade de manutenção. Nesta seção, são apresentadas as principais ferramentas e frameworks adotados, com ênfase no Django, framework principal da aplicação, além da infraestrutura de banco de dados e nuvem utilizada. Também são detalhados os motivos técnicos que justificam essas escolhas.

### 4.4.1 Django

O Django é um framework web de alto nível baseado em Python que oferece uma série de recursos que o tornam ideal para o desenvolvimento de sistemas como o nosso gerenciador de reservas de quartos (PYTHON, 2025). Uma de suas principais vantagens é a rapidez no desenvolvimento, já que ele vem com diversas funcionalidades já prontas (DJANGO, 2024).

#### 4.4.1.1 Front-end

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

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

#### 4.4.1.2 Back-end

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

#### 4.4.1.3 Banco de Dados

O Banco de Dados é o PostgreSQL, sendo um banco escalável e flexível, este SGBD pode suportar grandes volumes de dados e de usuários além de ser compatível com uma grande gama de linguagens de programação. O PostgreSQL também é uma ótima opção por ser acessível, já que sua licença é livre, assim sem custos de licenciamento e a liberdade para modificar ou implementar o código-fonte da maneira que for necessária ([POSTGRESQL, 2025](#)).

### 4.4.2 Justificativa da Escolha

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

#### 4.4.2.1 Justificativa Técnica

Entre os diferenciais do Django, destacam-se o ORM nativo, que abstrai o uso de SQL e facilita a manipulação segura dos dados; o sistema integrado de autenticação e autorização, que oferece controle de acesso granular; e a proteção nativa contra ataques

como SQL Injection, Cross-Site Scripting (XSS) e Cross-Site Request Forgery (CSRF). Esses recursos reduzem significativamente o tempo de desenvolvimento e aumentam a segurança da aplicação. Além disso, o Django é um software de código aberto, com forte comunidade ativa, documentação completa e contínua evolução. Essa característica o torna ideal para projetos acadêmicos e corporativos, permitindo a entrega de soluções confiáveis e bem estruturadas ([DJANGO, 2024](#)).

#### 4.4.2.2 Infraestrutura com AWS

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

## 4.5 Ferramentas de Apoio

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.

### 4.5.1 GitHub

O GitHub foi utilizado para controle de versão e colaboração durante o desenvolvimento do sistema. A plataforma permite armazenar e gerenciar o código-fonte, realizar revisões e integrar funcionalidades de forma eficiente. O GitHub facilitou a organização do fluxo de trabalho, o rastreamento de mudanças e a colaboração entre os membros da equipe, promovendo maior controle e transparência no ciclo de desenvolvimento ([GITHUB, 2024](#)).

### 4.5.2 BRModelo

O BRModelo foi utilizado para a modelagem lógica e relacional do banco de dados. A ferramenta oferece uma interface intuitiva para construção de diagramas entidade-relacionamento (DER), o que auxiliou na estruturação clara das tabelas, relacionamentos e chaves do sistema ([BRMODELO, 2024](#)). O uso do BRModelo contribuiu diretamente para a coerência e integridade do esquema de dados implementado no PostgreSQL.

### 4.5.3 Visual Paradigm Online

O Visual Paradigm Online foi utilizado na criação dos diagramas de Implantação e Componentes. Esta ferramenta auxiliou na documentação da arquitetura do sistema, contribuindo para uma melhor compreensão dos fluxos e interações entre os componentes ([VISUAL-PARADIGM, 2024](#)). A versão online possibilitou colaboração remota e armazenamento em nuvem, o que otimizou a produtividade da equipe.

### 4.5.4 Latex

O LaTeX foi utilizado na produção e formatação do trabalho acadêmico. Por meio de seu sistema de marcação, foi possível obter um alto nível de controle sobre a estrutura e apresentação do documento, garantindo consistência, qualidade e organização ([LATEX, 2024](#)).

### 4.5.5 Google Meet

O Google Meet foi utilizado como plataforma de comunicação e realização de encontros virtuais da equipe ao longo do desenvolvimento do projeto ([GOOGLE, 2024](#)). As reuniões periódicas possibilitaram a discussão de tarefas, alinhamento de prazos e entregas mais organizadas.

### 4.5.6 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.

## 4.6 Manutenibilidade

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

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

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

A utilização do sistema de controle de versão Git, em conjunto com a plataforma GitHub, possibilita o rastreamento detalhado de alterações, revisão de código e colaboração eficaz entre os membros da equipe ([GITHUB, 2024](#)). Isso garante maior controle sobre o histórico de desenvolvimento e facilita a identificação e resolução de falhas.

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

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

## 4.7 Segurança, Privacidade e Legislação

A segurança da informação 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](#)).

### 4.7.1 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).

### 4.7.2 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).

### 4.7.3 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.
- **Transparência:** 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.

## 4.8 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.



### 4.8.1 Modelo Entidade-Relacionamento - MER

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

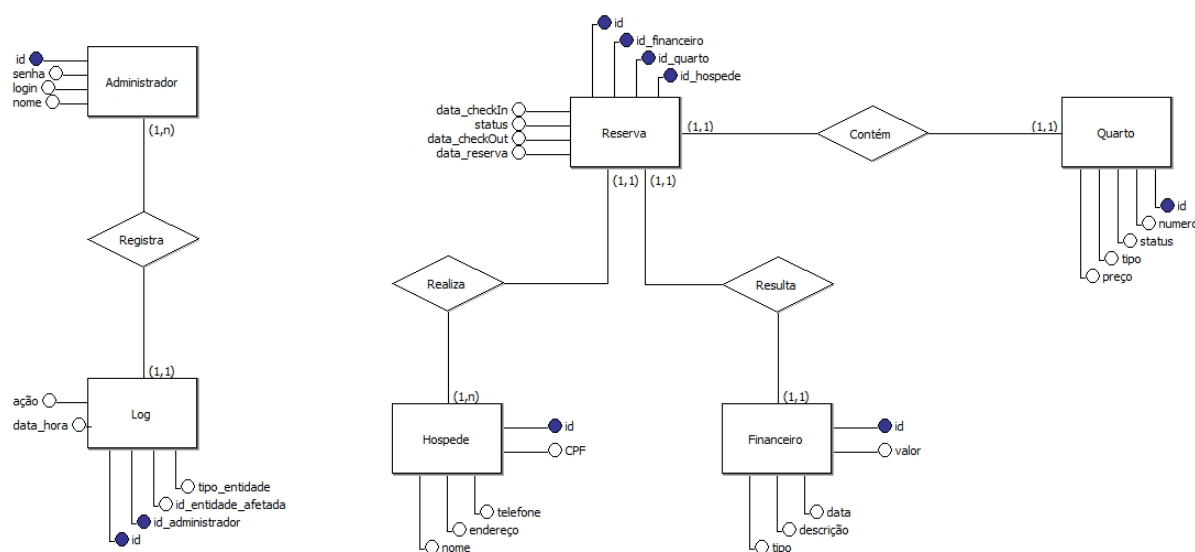


Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo

### 4.8.2 Diagrama Entidade-Relacionamento - DER

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

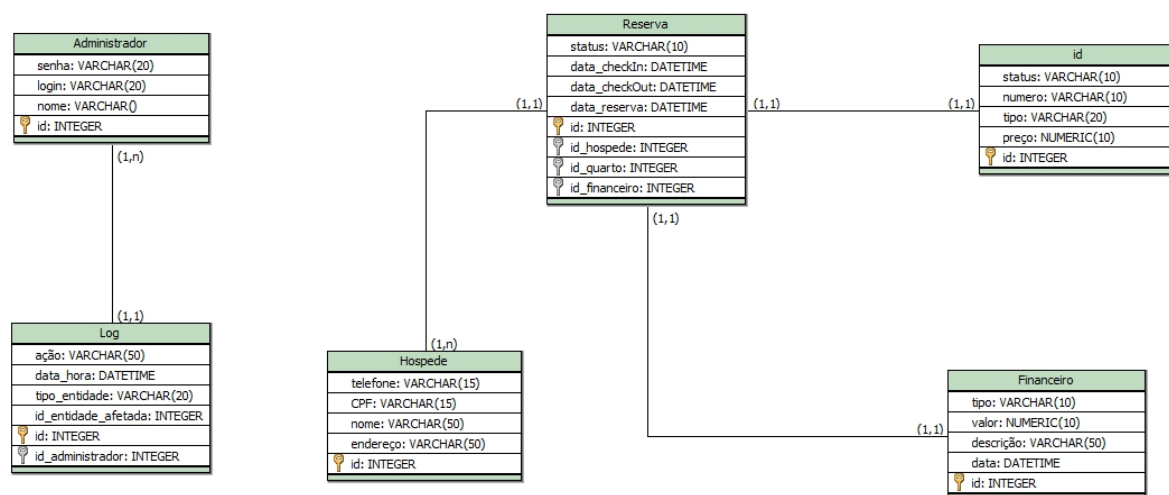


Figura 4 – Diagrama Entidade-Relacionamento (DER) desenvolvido no brModelo

## 4.9 Repositório da Aplicação

Nesta seção é apresentada a estrutura do ambiente de desenvolvimento do sistema web de gestão da Pousada Chalés Água de Coco. O código-fonte do sistema foi estruturado em aplicações modulares, arquivos de configuração geral do ambiente *Django* e arquivos

auxiliares. Essa configuração segue as boas práticas do *framework*, promovendo baixo acoplamento e alta coesão, o que garante um bom encapsulamento entre os módulos. Os códigos foram mantidos simples e otimizados para aproveitar as características dinâmicas do *Python* e as boas práticas do *framework*. Além disso, foram aplicadas boas práticas de organização e arquitetura para garantir facilidade de manutenção e evolução do sistema.

#### 4.9.1 Definição do repositório da aplicação

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

1. **/code/setup/setup:** Armazena os *scripts* de configuração do *framework* do sistema Django.
  - a) ***settings.py***: Define o diretório base do projeto (`BASE_DIR`), configura variáveis sensíveis como `SECRET_KEY`, modo de *debug*, domínios autorizados (`ALLOWED_HOSTS`), lista de aplicativos instalados (`INSTALLED_APPS`), *middlewares*, arquivos de URLs, sistema de templates e banco de dados (*PostgreSQL*).
  - b) ***urls.py***: Define as rotas URLs do projeto.
  - c) ***wsgi.py***: Arquivo padrão para *deploy* da aplicação via servidor WSGI (*Web Server Gateway Interface*). Define a *callable application* para a comunicação entre o servidor WSGI e o servidor *Django*.
  - d) ***manage.py***: Arquivo de execução dos comandos administrativos do *Django* via terminal. Define a configuração padrão, importa e executa o Django.
2. **/code/setup/apps/core:** Armazena as aplicações internas do sistema, sendo que cada aplicação representa um módulo do sistema.
  - a) ***/models***: Armazena os modelos que representam as tabelas no banco de dados:
    - i. ***hospede.py***: Define a estrutura da tabela “Hóspede”.
    - ii. ***quarto.py***: Define a estrutura da tabela “Quarto”.
    - iii. ***reserva.py***: Define a estrutura da tabela “Reserva”.
  - b) ***/forms***: Contém os formulários baseados nos modelos:
    - i. ***hospede.py***: Define os formulários para criar e editar objetos do tipo “Hóspede” na interface web.
    - ii. ***quarto.py***: Define os formulários baseados no modelo “Quarto”, usados para atividades como criar e editar quartos na interface web. Utiliza *widgets* para estilizar *inputs* e *labels*.

- iii. **reserva.py**: Define os formulários para criar e editar objetos do tipo “Reserva” na interface web.
  - c) **/urls**: Contém arquivos que os mapeiam as URLs específicas de cada módulo:
    - i. **hospede.py**: Define as rotas da aplicação “Hóspede”.
    - ii. **quarto.py**: Define as rotas da aplicação “Quarto”, como listagem de quartos e criação de um novo quarto.
    - iii. **reserva.py**: Define as rotas da aplicação “Reserva”, como listagem de quartos e criação de um novo quarto.
  - d) **/views**: Contém as funções responsáveis por processar as requisições:
    - i. **hospede.py**: Contém as *views* do módulo “Hóspede”, como `hospede_list`, `hospede_create` e `hospede_update`.
    - ii. **quarto.py**: Contém as *views* do módulo “Quarto”, como `excluir_quarto` e `tipos_quarto`.
    - iii. **reserva.py**: Contém as *views* do módulo “Reserva”, como `marcar_checkin` e `list_quarto`.
3. **/code/setup/templates**: Armazena os arquivos HTML do sistema.
- a) **/core/**:
    - i. **/hospede/**: Armazena os arquivos HTML do módulo de gestão de hóspedes.
    - ii. **/quarto/**: Armazena os arquivos HTML do módulo de gestão de quartos.
    - iii. **/reserva/**: Armazena os arquivos HTML do módulo de gestão de reservas.
  - b) **/shared/base.html**: *Template* utilizado como estrutura principal de todas as páginas do sistema, definindo um *layout* comum. Utiliza ícones do FontAwesome e é estilizado pelo arquivo *output.css*, criado com o Tailwind CSS e DaisyUI.
4. **/code/setup/static/css/output.css**: *CSS* gerado pela biblioteca Tailwind CSS com base nas configurações do sistema. Contém variáveis, estilos base e camadas para definir cores, espaçamentos, tamanhos de fonte, *padding*s, bordas e define comportamentos padrão.

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

## 4.10 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.

### 4.10.1 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.

### 4.10.2 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.

### 4.10.3 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.

### 4.10.4 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.

#### 4.10.5 Produto Mínimo Viável - 24/06

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

## 5 Viabilidade Financeira

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.

### 5.1 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.

#### 5.1.1 Custo Estrutural

Os custos estruturais englobam os ativos e serviços necessários para o desenvolvimento e implantação do sistema. Isso inclui desde os equipamentos (*hardware*) utilizados pela equipe até os serviços de hospedagem, domínios e licenças de *software* (serviços e infraestrutura).

Para análise desses custos, a [Tabela 1](#) consolida todos esses itens, distinguindo os custos efetivos no contexto acadêmico dos custos estimados para uma implementação comercial.

Tabela 1 – Resumo dos custos estruturais

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
Hospedagem em Nuvem	Serviço	1	R\$ 0,00	R\$ 0,00
Total (9 meses)	-	-	R\$ 0.00	26.234,00

Fonte: Elaborado pelos autores

A [Tabela 1](#) evidencia a disparidade entre os dois cenários aplicados no projeto. 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 operação comercial exigiria um investimento inicial de R\$ 18.250,00 para aquisição de equipamentos. Adicionalmente, haveria custos operacionais mensais recorrentes estimados em R\$ 883,70 (somando depreciação, energia, internet e hospedagem). As estimativas dos custos comerciais dos equipamentos de *hardware* consideraram valores médios do mercado de cada unidade. 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](#)).

### 5.1.2 Custo de Pessoal

O levantamento de custos de pessoal focou no cenário comercial, visto que, enquanto projeto acadêmico, a mão de obra é voluntária e resulta em um custo inicial nulo para a pousada. Diante disso, foram obtidos dados salariais em plataformas especializadas para projetar os custos associados ao sistema em ambiente comercial ([Glassdoor, 2025](#); [Indeed Brasil, 2025](#)).

A [Tabela 2](#) reúne os custos referentes a mão de obra do desenvolvimento do projeto, no contexto comercial. Nela são especificados: funções ocupadas pelos membros da equipe, seguida por quantidade de pessoas por cargo, horas trabalhadas por dia e dias trabalhados por mês(média). Enquanto a [Tabela 3](#) detalha: o preço por hora de cada profissional. A partir disso, faz-se as estimativas de custo mensal e a projeção em 4 meses e 9 meses.

Tabela 2 – Quantidade e horas trabalhadas por função

Função	Quantidade	Horas/Dia	Dias/Mês	Total de Horas/Mês
Engenheiro de Dados (DBA)	1	6	22	132
Analista de Documentação	1	6	22	132
Gerente de Projeto (PM)	1	6	22	132
Desenvolvedor Front-End	1	6	22	132
Desenvolvedor Back-End	1	6	22	132
<b>Total Mão de Obra</b>	6	36	132	792

Fonte: Elaborado pelos autores.

Tabela 3 – Custos por função

Função	Custo Hora (R\$)	Custo Mensal (R\$)	Total 4 meses (R\$)	Total 9 meses (R\$)
Analista de QA	15,86	2.093,85	8.375,40	18.844,65
Engenheiro de Software	18,75	2.475,00	9.900,00	22.275,00
Gerente de Projeto (PM)	16,00	2.112,00	8.448,00	19.008,00
Desenvolvedor Front-End	13,00	1.716,00	6.864,00	15.444,00
Desenvolvedor Back-End	13,00	1.716,00	6.864,00	15.444,00
<b>Total Mão de Obra</b>	76,61	10.112,85	46.451,7	91.015,65

Fonte: Elaborado pelos autores.

A projeção de custos de pessoal levou em consideração os períodos de 4 e 9 meses, pois foram momentos importantes no desenvolvimento. Nos primeiros 4 meses foi desenvolvido e entregue o MVP do projeto, enquanto que o período de 9 meses representa o tempo disponível para o planejamento, desenvolvimento e implantação do sistema. Permitindo, assim, uma visão mais ampla dos custos de mão de obra.

Enquanto que, os cargos representam os papéis necessários e atuados para a entrega do sistema projetado, de forma que, os valores levantados são referentes a salários de profissionais júnior. Assim como a quantidade de horas e dias trabalhados, que representam o tempo investido efetivamente pela equipe no atual projeto.

Portanto, em um cenário de implementação comercial e baseado nos dados coletados, a pousada parceira teria um investimento mínimo para mão de obra de R\$ 46.451,70 no primeiro quadrimestre e um total de R\$ 91.015,00 para todo o processo.

### 5.1.3 Custo Total

O custo total do projeto representa a somatória dos custos estruturais e de pessoal para a realização do projeto. Essa despesa integral está demonstrada na [Tabela 4](#) que une os custos por mão de obra e custos por estrutura em 3 períodos diferentes: mensal — valor referente a gastos de um único mês; 4 meses — valor referente primeiro quadrimestre



trabalhado até a entrega do MVP; e 9 meses — referente ao total de meses programado para a realização do projeto.

Tabela 4 – Custo Total por Categoria

<b>Categoria</b>	<b>Custo Mensal (R\$)</b>	<b>Custo Total (4 meses) (R\$)</b>	<b>Custo Total (9 meses) (R\$)</b>
Mão de Obra	10.112,85	46.451,7	91.015,65
Estrutura	883,70	3.534,80	7.953,30
<b>Total</b>	10.996,55	49.986,5	98.968,95

Fonte: Elaborado pelos autores.

Dessa forma, contabilizando os custos estruturais e de pessoal, a pousada teria que destinar o valor mínimo inicial de R\$ 98.968,95 para o atual projeto.

## 6 Considerações Finais

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

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

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

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

## 7 Conclusão

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

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

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

# Referências

- ALDRIGUI, M. *Meios de hospedagem*. São Paulo: Aleph, 2007. (Coleção ABC do Turismo). ISBN 978-85-7657-031-8. Citado na página 17.
- 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 70.
- AWS. *AWS - Amazon Web Services*. 2025. Disponível em: <<https://aws.amazon.com/documentation/>>. Acesso em: 7 jun. 2025. Citado 3 vezes nas páginas 55, 59 e 62.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. 3. ed. [S.l.]: Addison-Wesley, 2012. Citado na página 54.
- 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 17.
- 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 61.
- 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](https://www.researchgate.net/publication/222452921_Strategic_Use_of_Information_Technologies_in_the_Tourism_Industry)>. Citado na página 18.
- 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 33.
- DJANGO. *Django Documentation*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://docs.djangoproject.com/pt-br/4.2/>>. Citado 4 vezes nas páginas 54, 57, 59 e 62.
- DUCKETT. *HTML and CSS: Design and Build Websites*. [S.l.]: Wiley, 2011. ISBN 9781118008188. Citado na página 54.
- 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](https://www.enel.com.br/pt-saopaulo/Para_Voce/tarifa-energia-eletrica.html)>. Citado na página 70.
- 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-semester-e-fatura-108-bilhoes>>. Acesso em: 22 set. 2025. Citado na página 17.

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 12.

GITHUB. *GitHub Docs*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://docs.github.com/>>. Citado 2 vezes nas páginas 59 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 70.

GOOGLE. *Google Meet Help*. [S.l.], 2024. Acessado em 7 de junho de 2025. Disponível em: <<https://support.google.com/meet/>>. Citado na página 60.

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 18.

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 34.

IGNARRA, L. R. *Fundamentos do turismo*. 3. ed. São Paulo: Thomson Learning, 2013. Citado na página 17.

Indeed Brasil. *Salários de empresas*. 2025. Acesso em: 05 jun. 2025. Disponível em: <<https://br.indeed.com/career/salaries>>. Citado na página 70.

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 14 e 55.

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 18.

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://redeotec.mec.gov.br/images/stories/pdf/eixo\\_hosp\\_lazer/061112\\_gest\\_hot.pdf](https://redeotec.mec.gov.br/images/stories/pdf/eixo_hosp_lazer/061112_gest_hot.pdf)>. Citado na página 18.

MAURÍCIO, N. R.; RAMOS, K. C. M. d. Gestão na hotelaria. *Revista F@pciência*, 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](https://www.fap.com.br/fap-ciencia/edicao_2011/011.pdf)>. Citado na página 18.

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 12 e 14.

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 18.

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 na página 62.

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 54 e 58.

PRESSMAN, R. S. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: McGraw-Hill, 2016. Citado na página 53.

PROJECTLIBRE. *ProjectLibre*. [S.l.], 2025. Acessado em 24 de junho de 2025. Disponível em: <<https://www.projectlibre.com/>>. Citado na página 60.

PYTHON. *Python Official Documentation*. 2025. Disponível em: <<https://docs.python.org/3/>>. Acesso em: 7 jun. 2025. Citado 2 vezes nas páginas 57 e 58.

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 70.

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.

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 20 e 23.

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 35.

SOMMERVILLE, I. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019. Citado 2 vezes nas páginas 53 e 54.

SOUSA, A. J. L. D. et al. *Sistema Web Gestão Pousada*. 2025. <https://github.com/guilhermebschmidt/SistemaWebGestaoPousada.git>. Acesso em: 9 jun. 2025. Citado na página 66.

STAIR, R.; REYNOLDS, G. *Princípios de Sistemas de Informação*. 13. ed. São Paulo: Cengage Learning, 2020. Citado na página 14.

TOTVS. *Digitalizacao no setor hoteleiro: os beneficios da tecnologia*. 2024. Acessado em 29 de agosto de 2025. Disponível em: <<https://www.totvs.com/blog/gestao-hoteleira/digitalizacao-no-setor-hoteleiro/>>. Citado 2 vezes nas páginas 12 e 13.

TURBAN, E.; POLLARD, C.; WOOD, G. *Sistemas de Informação: Uma Abordagem Gerencial*. 9. ed. Porto Alegre: Bookman, 2018. Citado 2 vezes nas páginas 14 e 55.

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](https://unstats.un.org/unsd/publication/Seriesm/SeriesM_83rev1e.pdf)>. Citado na página 17.

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 17 e 18.

## Apêndices



# APÊNDICE A – Diário de Bordo

## A.1 1º SEMANA

Período: 25/03/2025 a 01/04/2025.

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

## A.2 2º SEMANA

Período: 01/04/2025 a 08/04/2025.

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

## A.3 3º SEMANA

Período: 08/04/2025 a 15/04/2025.

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

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

## A.4 4° SEMANA

Período: 15/04/2025 a 22/04/2025.

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

## A.5 5° SEMANA

Período: 22/04/2025 a 29/04/2025.

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

## A.6 6° SEMANA

Período: 29/04/2025 a 06/05/2025.

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

## A.7 7º SEMANA

Período: 06/05/2025 a 13/05/2025.

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

## A.8 8º SEMANA

Período: 13/05/2025 a 20/05/2025.

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

## A.9 9º SEMANA

Período: 20/05/2025 a 27/05/2025.

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

## A.10 10º SEMANA

Período: 27/05/2025 a 03/06/2025.

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

## A.11 11º SEMANA

Período: 03/06/2025 a 10/06/2025.

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

- Anna Julia: Entregar seus tópicos da documentação (Revisão da Literatura, Viabilidade Financeira e Tecnologias e Ferramentas) e auxiliar na revisão e finalização da Documentação.
- 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 12° 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 13° 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.

# APÊNDICE B – Questionário Aplicado à Proprietária da Pousada

## Questionário Aplicado à Proprietária – Parte 1

<b>Domínio: Reservas</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.

Fonte: Elaborado pelos autores.

Quadro 21 – Questionário Aplicado à Proprietária – Parte 1

## Questionário Aplicado à Proprietária – Parte 2

<b>Domínio: Hóspedes</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.
<b>Domínio: Check-in e Check-out</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.

Fonte: Elaborado pelos autores.

Quadro 22 – Questionário Aplicado à Proprietária – Parte 2

## Questionário Aplicado à Proprietária – Parte 3

<b>Domínio: Quartos</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.
<b>Domínio: Serviços Adicionais</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
16	A pousada oferece serviços extras?	Não.
17	Se sim, esses serviços devem ser registrados no sistema junto à reserva?	(Não há serviços extras)
<b>Domínio: Pagamentos</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.

Fonte: Elaborado pelos autores.

## Questionário Aplicado à Proprietária – Parte 4

20	É necessário gerar comprovante ou recibo no sistema?	Eu envio uma confirmação de reserva com todas as informações.
<b>Domínio: Comunicação</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.
<b>Domínio: Acesso ao Sistema</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.
<b>Domínio: Relatórios e Controle</b>		
<b>Item</b>	<b>Pergunta</b>	<b>Resposta</b>
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.





## APÊNDICE C – Product Backlog Detalhado

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.

Código	Item	Categoria	Prioridade
19	Criar interfaces para gestão de quartos	Gestão de Quartos	MÉDIA
20	Implementar funcionalidade de alteração manual do status (disponível, indisponível, em manutenção) do quarto	Gestão de Quartos	MÉDIA
21	Criar funcionalidade de cadastro de hóspedes	Gestão de Hóspedes	ALTA
22	Criar funcionalidade de exclusão do cadastro de hóspedes	Gestão de Hóspedes	MÉDIA
23	Criar funcionalidade de edição do cadastro de hóspedes	Gestão de Hóspedes	MÉDIA
24	Criar interfaces para gestão de hóspedes	Gestão de Hóspedes	MÉDIA
25	Criar funcionalidade de cadastro de reservas	Gestão de Reservas	ALTA
26	Criar funcionalidade de exclusão do cadastro de reservas	Gestão de Reservas	ALTA
27	Criar funcionalidade de edição do cadastro de reservas	Gestão de Reservas	ALTA
28	Criar interfaces para gestão de Reservas	Gestão de Reservas	ALTA
29	Criar funcionalidade para visualização do histórico de reservas por hóspede	Gestão de Reservas	MÉDIA
30	Criar lógica de validação de disponibilidade de quartos para reservas	Gestão de Reservas	ALTA
31	Implementar atualização automática do status do quarto para “ocupado” após o <i>check-in</i>	<i>Check-in</i> e <i>Check-out</i>	ALTA
32	Criar funcionalidade de atualização do <i>check-out</i>	<i>Check-in</i> e <i>Check-out</i>	ALTA
33	Integrar ambientes, <i>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.

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.

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	Registrar escolhas e mudanças de rumo	Documentação	ALTA
66	Documentar problemas ocorridos e lições aprendidas	Documentação	ALTA
67	Elaborar o plano de implantação	Implantação	MÉDIA
68	Realizar implantação do sistema	Implantação	MÉDIA
69	Revisão final da documentação técnica	Documentação	ALTA
70	Treinamento da proprietária da pousada para uso da aplicação	Implantação	MÉDIA

Fonte: Elaborado pelos autores.

Quadro 28 – Product Backlog - Parte 4

## APÊNDICE D – Requisitos Funcionais Detalhados

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 e-mail	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 29 – Requisitos Funcionais - Parte 1

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 30 – Requisitos Funcionais - Parte 2

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 <i>early 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 31 – Requisitos Funcionais - Parte 3



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.

Quadro 32 – Requisitos Funcionais - Parte 4

# APÊNDICE E – 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

---

<b>Funcionalidade</b>	Criar Hóspede
-----------------------	---------------

---

<b>Cenário de Teste</b>	Criar um novo hóspede com todos os dados válidos
-------------------------	--

---

**Passos para Execução**

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".

---

<b>Resultado Esperado</b>	O sistema deve salvar o hóspede, redirecionar para a lista e o novo hóspede deve aparecer na tabela.
---------------------------	--

---

<b>Status</b>	<b>Passou</b>
---------------	---------------

---

<b>Observações</b>	Fluxo de criação principal funcionando corretamente.
--------------------	--

---

## ID do Teste: HOS-002

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

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

## ID do Teste: HOS-004

<b>Funcionalidade</b>	Atualizar Hóspede
<b>Cenário de Teste</b>	Editar o telefone de um hóspede
<b>Passos para Execução</b>	<ol style="list-style-type: none"><li>1. Clicar em Hospede.</li><li>2. Clicar em "Listar Hóspedes".</li><li>3. Na lista de hóspedes, clicar no nome de um hóspede cadastrado.</li><li>4. Clicar em "Editar".</li><li>5. Alterar o número de telefone.</li><li>6. Clicar em "Salvar".</li></ol>
<b>Resultado Esperado</b>	O sistema deve salvar a alteração e, na lista, o novo telefone deve ser exibido.
<b>Status</b>	<b>Passou</b>
<b>Observações</b>	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"><li>1. Clicar em Hospede.</li><li>2. Clicar em "Listar Hóspedes".</li><li>3. Na lista de hóspedes, usar a barra de busca para procurar por parte do nome de um hóspede cadastrado.</li><li>4. Verificar o resultado.</li><li>5. Limpar a busca e procurar por um nome que não existe (ex: "Zzzzz").</li></ol>
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	Falhou
Observações	<b>BUG-001:</b> Após inserir um nome existente na barra de busca, o resultado foi abrir uma pagina com o erro " <b>Page not found (404)</b> ".

## ID do Teste: HOS-006

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

# APÊNDICE F – 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

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



## ID do Teste: QUA-002

<b>Funcionalidade</b>	Criar Quarto
<b>Cenário de Teste</b>	Criar um novo quarto com status "Indisponível"
<b>Passos para Execução</b>	<ol style="list-style-type: none"><li>1. Clicar em "Quartos".</li><li>2. Clicar no botão "Adicionar Quarto".</li><li>3. Preencher os campos, mas deixar a caixa "Status" desmarcada.</li><li>4. Clicar em "Adicionar".</li></ol>
<b>Resultado Esperado</b>	O quarto deve ser criado e aparecer na lista com o status "Indisponível".
<b>Status</b>	<b>Passou</b>
<b>Observações</b>	Ao clicar em "Adicionar", o sistema redirecionou para a lista de quartos, com o status certo "Indisponível".

## ID do Teste: QUA-003

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

## ID do Teste: QUA-004

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

## ID do Teste: QUA-005

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

## ID do Teste: QUA-006

<b>Funcionalidade</b>	Excluir Quarto
<b>Cenário de Teste</b>	Excluir um quarto de forma permanente
<b>Passos para Execução</b>	<ol style="list-style-type: none"><li>1. Clicar em "Quartos".</li><li>2. Clicar em "Listar Quartos".</li><li>3. Clicar em "Excluir" no quarto escolhido.</li><li>4. No modal de confirmação, clicar em "Sim, excluir".</li></ol>
<b>Resultado Esperado</b>	O quarto deve ser removido da lista permanentemente.
<b>Status</b>	Falhou
<b>Observações</b>	<b>BUG-002:</b> após clicar na confirmação de exclusão, a página é redirecionada para uma outra página contendo o erro "TemplateDoesNotExist".

# APÊNDICE G – 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

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

ID do Teste: RES-002

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

## ID do Teste: RES-003

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



ID do Teste: RES-004

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

## ID do Teste: RES-005

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

## ID do Teste: RES-006

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

ID do Teste: RES-007

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

## ID do Teste: RES-008

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