

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

Pousada Chalés Água de Coco

São Paulo - SP - Brasil

2025

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

Pousada Chalés Água de Coco

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

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

Orientador: Marcelo Tavares de Santana

São Paulo - SP - Brasil

2025

Resumo

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

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

Abstract

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

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

Lista de ilustrações

Figura 1 – Diagrama de Implantação desenvolvido no Online Visual-Paradigm .	34
Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm .	35
Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo .	40
Figura 4 – Diagrama Entidade-Relacionamento (DER) desenvolvido no brModelo	41
Figura 5 – Comparação dos custos e receitas acumuladas no cenário otimista . .	44
Figura 6 – Comparação dos custos e receitas acumuladas no cenário pessimista .	45
Figura 7 – Comparação dos custos e receitas acumuladas no cenário realista . .	46

Lista de quadros

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

Lista de tabelas

Tabela 1 – Função dos Integrantes da Equipe	23
Tabela 2 – Resumo dos custos de Equipamentos e Serviços	42
Tabela 3 – Resumo dos custos de Infraestrutura	43
Tabela 4 – Quantidade e horas trabalhadas por função	43
Tabela 5 – Custos por função	43
Tabela 6 – Custo Total por Categoria	44

Lista de abreviaturas e siglas

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

Sumário

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

4.4.1.2	Back-end	36
4.4.1.3	Banco de Dados	36
4.4.2	Justificativa da Escolha	36
4.4.2.1	Justificativa Técnica	36
4.4.2.2	Infraestrutura com AWS	37
4.5	Ferramentas de Apoio	37
4.5.1	GitHub	37
4.5.2	BRModelo	37
4.5.3	Visual Paradigm Online	37
4.5.4	Latex	38
4.5.5	Google Meet	38
4.6	Manutenibilidade	38
4.7	Segurança, Privacidade e Legislação	39
4.7.1	Segurança da Aplicação	39
4.7.2	Segurança na Comunicação	39
4.7.2.1	Conformidade com a LGPD	39
4.8	Modelagem do Banco de Dados	40
4.8.1	Modelo Entidade-Relacionamento - MER	40
4.8.2	Diagrama Entidade-Relacionamento - DER	40
4.9	Entregas	41
5	VIABILIDADE FINANCEIRA	42
5.1	Custos	42
5.1.1	Custo Estrutural	42
5.1.2	Custo Mão de Obra	43
5.1.3	Custo Total	44
5.2	Cenários	44
5.2.1	Cenário Otimista	44
5.2.2	Cenário Pessimista	45
5.2.3	Cenário Realista	45
6	CONSIDERAÇÕES FINAIS	47
7	CONCLUSÃO	48
	REFERÊNCIAS	49

	APÊNDICES	51
	APÊNDICE A – DIÁRIO DE BORDO	52
A.1	1° SEMANA	52
A.2	2° SEMANA	52
A.3	3° SEMANA	52
A.4	4° SEMANA	53
A.5	5° SEMANA	53
A.6	6° SEMANA	53
A.7	7° SEMANA	54
A.8	8° SEMANA	54
A.9	9° SEMANA	54
A.10	10° SEMANA	54
A.11	11° SEMANA	54
A.12	12° SEMANA	55
A.13	13° SEMANA	55
	APÊNDICE B –	56
	ANEXOS	57
	ANEXO A –	58

1 Introdução

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

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

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

1.1 Objetivo

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

1.2 Justificativa

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

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

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

1.3 Análise da Concorrência

1.3.1 SimpleHotel

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

1.3.2 HospedaJá

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

1.3.3 Comparativo

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

Fonte: Elaborado pelos Autores.

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

2 Revisão de Literatura

2.1

3 Gestão do Projeto

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

3.1 Organização da Equipe

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

3.1.1 Funções e Responsabilidades

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

3.2 Metodologias de gestão e desenvolvimento

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

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

Fonte: Elaborado pelos autores.

Quadro 2: Função e Responsabilidades da Equipe do Projeto

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

3.2.1 Time Scrum

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

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

Tabela 1 – Função dos Integrantes da Equipe

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

Fonte: Elaborado pelos autores.

3.3 Artefatos

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

3.3.1 Product Backlog

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

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

Fonte: Autor.

Quadro 3: Product backlog

4 Desenvolvimento do Projeto

4.1 Escopo do Projeto

4.1.1 Requisitos Funcionais

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

Fonte: Elaborado pelos autores.

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

Fonte: Elaborado pelos autores.

Quadro 5: Requisitos Funcionais - Parte 2

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

Fonte: Elaborado pelos autores.

Quadro 6: Requisitos Funcionais - Parte 3

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

Fonte: Elaborado pelos autores.

Quadro 7: Requisitos Funcionais - Parte 4

4.1.2 Requisitos Não Funcionais

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

Fonte: Elaborado pelos autores.

Quadro 8: Requisitos Não Funcionais - Parte 1

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

Fonte: Elaborado pelos autores.

Quadro 9: Requisitos Não Funcionais - Parte 2

4.1.3 Regras de Negócio

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

Fonte: Elaborado pelos autores.

Quadro 10: Regras de Negócio

4.2 História de Usuário

4.2.1 Descrição

4.3 Arquitetura

4.3.1 Desenho da Arquitetura

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

4.3.1.1 Estrutura da Solução

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

4.3.1.2 Banco de Dados e Infraestrutura

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

4.3.2 Diagrama da Arquitetura

4.3.2.1 Diagrama de Implantação

A Figura 1 mostra o funcionamento da arquitetura do sistema.

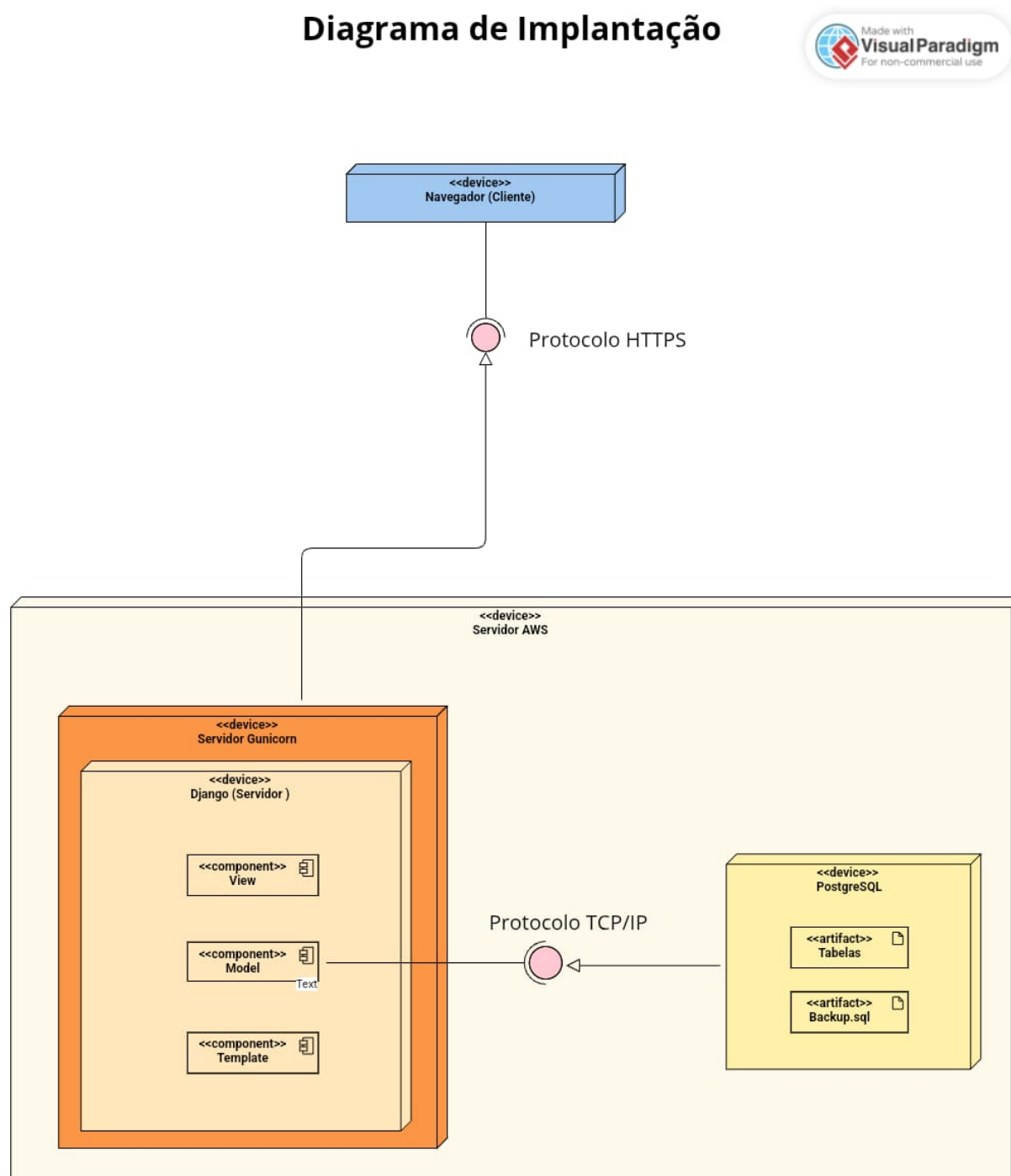


Figura 1 – Diagrama de Implantação desenvolvido no Online Visual-Paradigm

4.3.2.2 Diagrama de Componentes

A Figura 2 mostra o funcionamento da arquitetura do sistema.

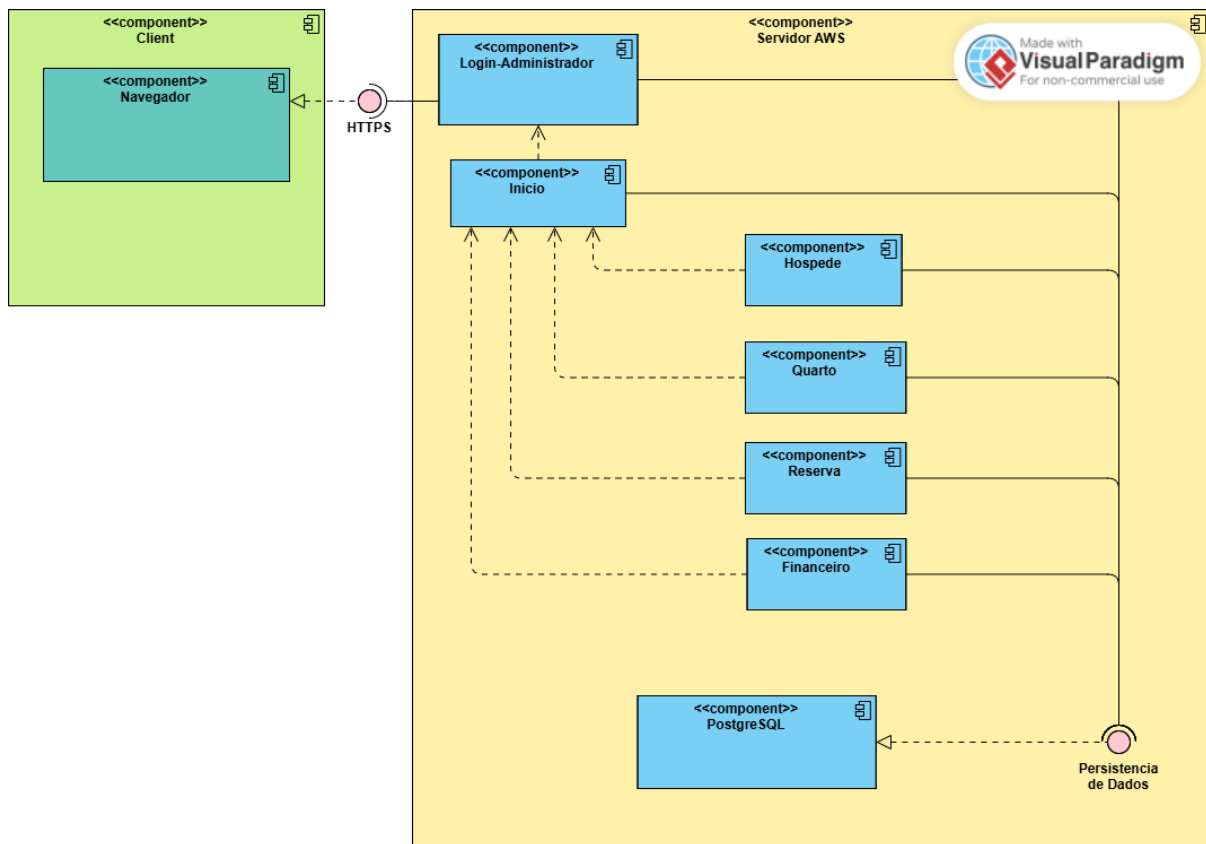


Figura 2 – Diagrama de Componentes desenvolvido no Online Visual-Paradigm

4.4 Tecnologias

4.4.1 Django

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

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.

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.

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.

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. Os recursos de computação elástica, gerenciamento de banco de dados, balanceamento de carga, armazenamento e backup são fundamentais para a operação de um sistema que lida com informações sensíveis de clientes e reservas. A integração entre Django e AWS ocorre de forma transparente, possibilitando o uso de serviços como S3 (armazenamento de mídia), RDS (gerenciamento de banco de dados relacional) e CloudWatch (monitoramento), ampliando o potencial da aplicação e assegurando a continuidade do serviço com mínimo risco de falhas.

4.5 Ferramentas de Apoio

4.5.1 GitHub

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

4.5.2 BRModelo

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

4.5.3 Visual Paradigm Online

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

4.5.4 Latex

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

4.5.5 Google Meet

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

4.6 Manutenibilidade

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

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

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

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

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

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

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

4.7 Segurança, Privacidade e Legislação

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

4.7.1 Segurança da Aplicação

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

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

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

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

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

4.7.2 Segurança na Comunicação

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

4.7.2.1 Conformidade com a LGPD

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

Solicitação de remoção de dados pessoais;

Alteração de consentimento previamente fornecido;

Acesso claro às finalidades do tratamento dos dados.

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

4.8 Modelagem do Banco de Dados

4.8.1 Modelo Entidade-Relacionamento - MER

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

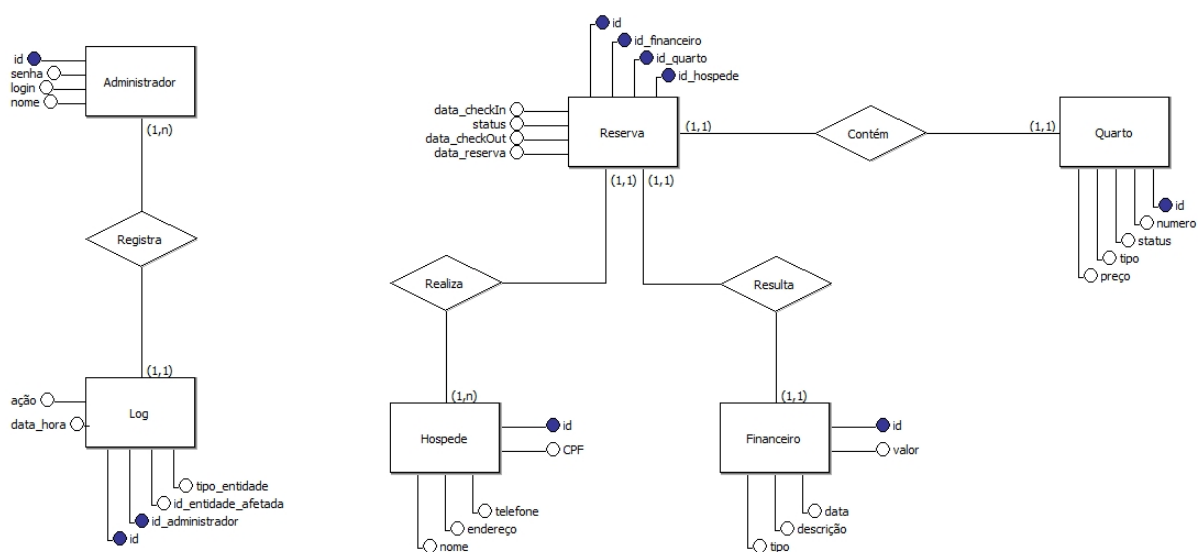


Figura 3 – Modelo Entidade-Relacionamento (MER) desenvolvido no brModelo

4.8.2 Diagrama Entidade-Relacionamento - DER

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

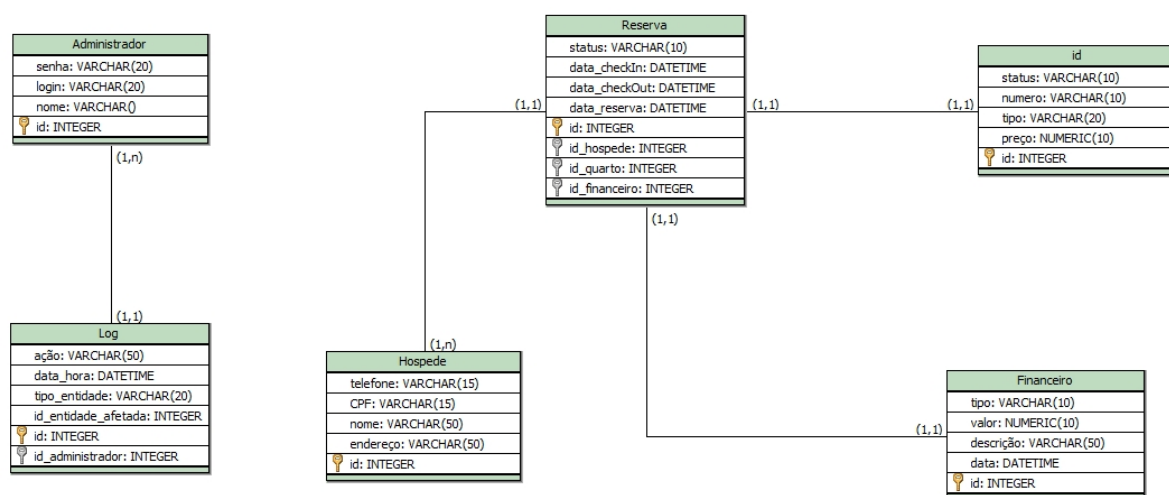


Figura 4 – Diagrama Entidade-Relacionamento (DER) desenvolvido no brModelo

4.9 Entregas

5 Viabilidade Financeira

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

5.1 Custos

5.1.1 Custo Estrutural

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

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

Tabela 2 – Resumo dos custos de Equipamentos e Serviços

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

Fonte: Elaborado pelos autores.

Tabela 3 – Resumo dos custos de Infraestrutura

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

Fonte: Elaborado pelos autores.

5.1.2 Custo Mão de Obra

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

Tabela 4 – Quantidade e horas trabalhadas por função

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

Fonte: Elaborado pelos autores.

Tabela 5 – Custos por função

Função	Custo Hora (R\$)	Custo Mensal (R\$)	Total 4 meses (R\$)	Total 9 meses (R\$)
Analista de Cronograma (PMO)	16,00	2.112,00	8.448,00	19.008,00
Engenheiro de Dados (DBA)	13,00	1.716,00	6.864,00	15.444,00
Analista de Documentação	15,00	1.980,00	7.920,00	17.820,00
Gerente de Projeto (PM)	16,00	2.112,00	8.448,00	19.008,00
Desenvolvedor Front-End	13,00	1.716,00	6.864,00	15.444,00
Desenvolvedor Back-End	13,00	1.716,00	6.864,00	15.444,00
Total Mão de Obra	86,00	11.352,00	45.408,00	102.168,00

Fonte: Elaborado pelos autores.

5.1.3 Custo Total

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

Tabela 6 – Custo Total por Categoria

Categoria	Custo Mensal (R\$)	Custo Total (4 meses) (R\$)	Custo Total (9 meses) (R\$)
Mão de Obra	11.352,00	45.408,00	102.168,00
Estrutura	118,41	473,64	1.065,69
Total	11.470,41	45.881,64	103.233,69

Fonte: Elaborado pelos autores.

5.2 Cenários

5.2.1 Cenário Otimista

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

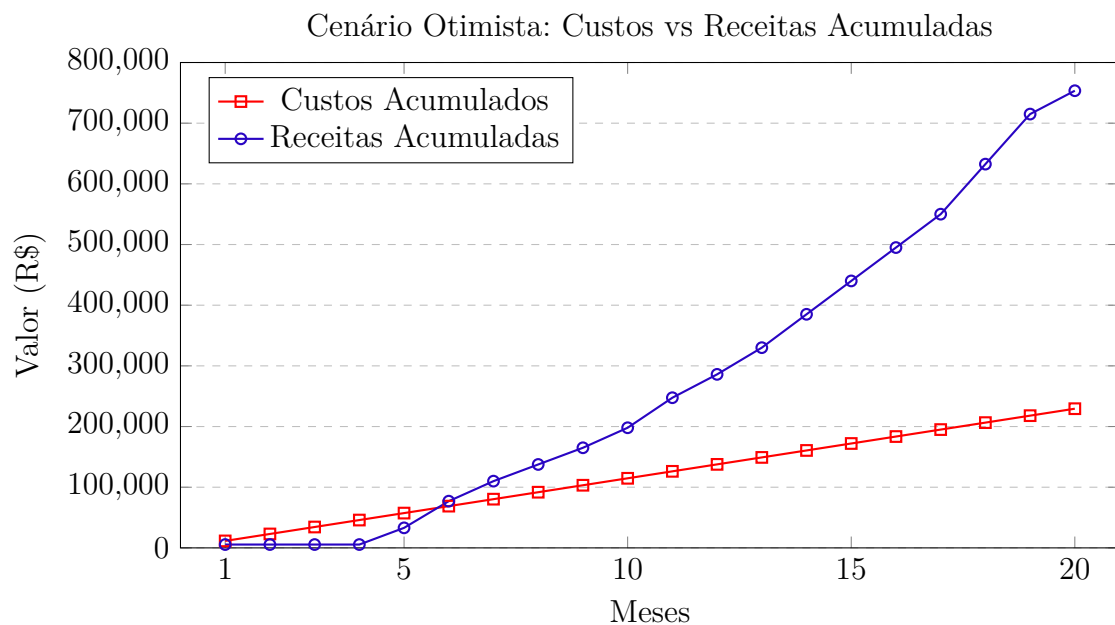


Figura 5 – Comparação dos custos e receitas acumuladas no cenário otimista

5.2.2 Cenário Pessimista

Na imagem a seguir podemos observar a projeção construída pela equipe de um cenário pessimista, levando em consideração os custos já apresentados anteriormente de mão de obra e estrutura. Podemos observar que de acordo com essa projeção, o ponto de equilíbrio será alcançado por volta de 19 meses.

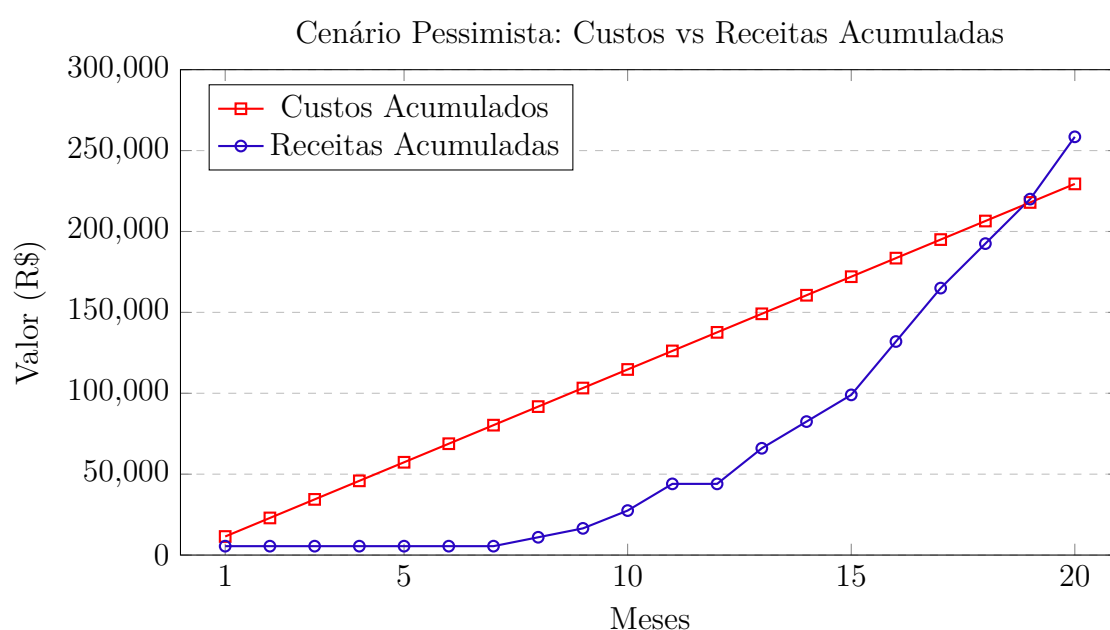


Figura 6 – Comparação dos custos e receitas acumuladas no cenário pessimista

5.2.3 Cenário Realista

Na imagem a seguir podemos observar a projeção construída pela equipe de um cenário realista, levando em consideração os custos já apresentados anteriormente de mão de obra e estrutura. Podemos observar que de acordo com essa projeção, o ponto de equilíbrio será alcançado por volta de 14 meses.

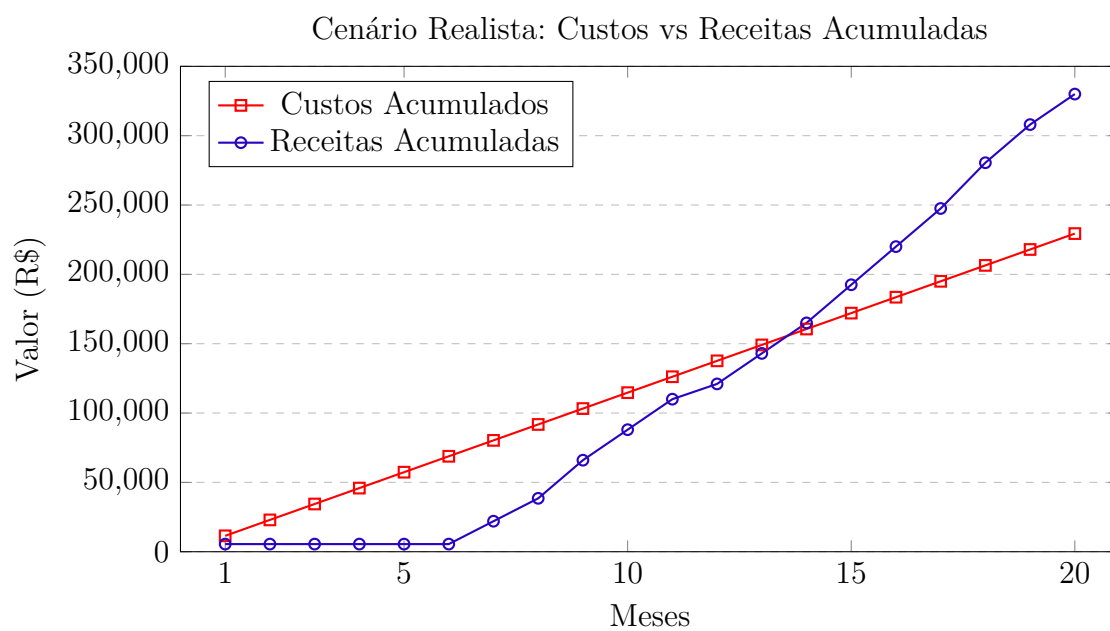


Figura 7 – Comparação dos custos e receitas acumuladas no cenário realista

6 Considerações Finais

7 Conclusão

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

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

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

Referências

Apêndices

APÊNDICE A – Diário de Bordo

A.1 1º SEMANA

Período: 25/03/2025 a 01/04/2025.

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

A.2 2º SEMANA

Período: 01/04/2025 a 08/04/2025.

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

A.3 3º SEMANA

Período: 08/04/2025 a 15/04/2025.

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

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

A.4 4° SEMANA

Período: 15/04/2025 a 22/04/2025.

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

A.5 5° SEMANA

Período: 22/04/2025 a 29/04/2025.

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

A.6 6° SEMANA

Período: 29/04/2025 a 06/05/2025.

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

A.7 7º SEMANA

Período: 06/05/2025 a 13/05/2025.

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

A.8 8º SEMANA

Período: 13/05/2025 a 20/05/2025.

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

A.9 9º SEMANA

Período: 20/05/2025 a 27/05/2025.

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

A.10 10º SEMANA

Período: 27/05/2025 a 03/06/2025.

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

A.11 11º SEMANA

Período: 03/06/2025 a 10/06/2025.

A.12 12° SEMANA

Período: 10/06/2025 a 17/06/2025.

A.13 13° SEMANA

Período: 17/06/2025 a 24/06/2025.

APÊNDICE B –

Anexos

ANEXO A –