# **Projeto Brisa - Agenda Virtual**

**Nome do Projeto:** Agenda VTA

**Autores:**

Augusto Azambuya Moretto da Silva (Back-end)

Lucas Ramos Alves (Communicator)

Mateus Franceschet Pereira (Front-end)

Patrick Vargas Santos (Full-Stack)

Roger Luiz do Nascimento Vesely (Scrum Master)

**Instituição Parceira:** Clínica Médica Veterinária Vet Assistance

**Professora** **Orientadora:** Caroline Pacheco da Rosa

### 2. Termo de Abertura do Projeto (TAP)

Nosso projeto será desenvolvido para a clínica médica veterinária "Vet Assistance", representada pelo nosso cliente, Sr. Tiago Siqueira Pereira. Atualmente a clínica gerencia os agendamentos das quatro salas de atendimento por meio de planilhas do Excel, um processo manual realizado exclusivamente pela equipe de recepção. Este método, trazido pelo cliente apresenta algumas falhas, como duplicação de horários e erros de agendamento. O bloqueio de salas especiais, ou visualização de espaços ocupados, fica inviabilizado utilizando a ferramenta atual.

Diante deste cenário, o grupo propõe o desenvolvimento de uma plataforma de agendamento digital e centralizada. A relevância desta solução está na sua capacidade de mitigar as falhas do processo atual, otimizando a gestão dos horários e dos profissionais nas diferentes salas. A plataforma oferecerá uma interface simples e de acesso restrito para a equipe da clínica, equipe de recepcionistas, setor administrativo e grupo de veterinários, na página de visualização de salas, teremos opções como: Agendamento e Bloqueio de Horários, com Visualização por Salas, sendo essa visualização o grande diferencial deste produto.

O sistema ainda contará com duas páginas diferentes de registros de usuários e relatórios da clínica e pacientes, uma página administrativa com restrições de acesso, e outra com informações importantes para a manutenção da clínica por meio da equipe de recepcionistas. Com isso, busca-se aumentar a confiabilidade dos agendamentos.

O Objetivo Principal do nosso sistema é o Agendamento e Visualização por Salas, isto deve ser claro a todo momento, pois acrescentar duas páginas de relatórios e registros de usuários aumenta consideravelmente o nível de dificuldade deste projeto. Mesmo sendo uma dificuldade calculada, esta escolha do grupo aumentará os riscos e quantidade de código a ser produzido.

## 2.1 A Plataforma

## O sistema Agenda VTA será uma aplicação web responsiva, acessível diretamente pelo navegador em computadores, sem necessidade de instalação local. Sua arquitetura será baseada em frameworks modernos de desenvolvimento, garantindo performance, segurança e escalabilidade.

## Interface e Usabilidade: A plataforma contará com um design simples e intuitivo, voltado para a rotina de recepcionistas e gestores. Todas as páginas terão acesso controlado por login e senha, com perfis diferentes de usuário (administrador, recepcionista, veterinário). As operações apresentarão feedback visual imediato para garantir segurança e clareza durante o uso.

## Agendamento e Visualização: O sistema permitirá o cadastro de horários com bloqueios automáticos, evitando conflitos. O grande diferencial será a visualização da agenda por salas em tempo real, com filtros por profissional, status e período. Essa visualização poderá ser feita em diferentes modos: diário, semanal e mensal.

## Relatórios e Gestão: Serão disponibilizados relatórios segmentados:

## Administrativo: informações gerenciais para apoio na tomada de decisão.

## Operacional: relatórios de pacientes, atendimentos e histórico de uso das salas.

## Segurança e Confiabilidade:

## Autenticação de usuários com senhas criptografadas.

## Controle de permissões por perfil.

## Backups automáticos periódicos dos dados.

## Com essa abordagem, a plataforma busca oferecer um ambiente centralizado, seguro e eficiente, reduzindo erros e otimizando o tempo da equipe da clínica.

### 

### 2.2 Objetivos do Projeto

#### O projeto Agenda VTA tem como finalidade principal otimizar o processo de agendamento da Clínica Veterinária Vet Assistance, trazendo mais eficiência, confiabilidade e organização ao fluxo de atendimentos.

#### **Objetivos Principais:**

#### **Agenda com Visualização por Salas:** Permitir que os usuários visualizem em tempo real a ocupação das salas de atendimento, diferenciando horários disponíveis, agendados e bloqueados, de forma clara e intuitiva.

#### **Dashboard com Relatórios e Controle de Usuários:** Disponibilizar painéis administrativos que auxiliem na gestão da clínica, apresentando relatórios de uso das salas, métricas de agendamentos e ferramentas de controle de usuários e permissões.

#### **Objetivos Secundários:**

#### **Sistema de Backup Automático:** Garantir a segurança das informações por meio de cópias diárias automáticas.

#### **Interface com Atalhos de Uso Frequente:** Facilitar a navegação do usuário com atalhos de teclado e ícones de acesso rápido às funções mais utilizadas.

#### **Sistema de Bloqueio Massivo:** Possibilitar o bloqueio de múltiplos horários ou de uma sala inteira em casos de manutenção ou indisponibilidade temporária.

#### **Atalhos do Teclado Facilitadores:** Oferecer comandos rápidos para agilizar ações recorrentes, reduzindo o tempo de operação.

#### **Resultados Esperados:**

#### Redução de falhas e conflitos de agendamento.

#### Aumento da produtividade da recepção.

#### Melhor organização visual da agenda.

#### Maior confiabilidade e satisfação da equipe usuária.

### 

### 2.3 - Stakeholders Envolvidos

### Cliente: Tiago Siqueira Pereira

* **Papel:** É o principal patrocinador e proponente do projeto. Ele define os requisitos de alto nível, valida as entregas e é o principal beneficiário da solução.
* **Interesse:** Obter um sistema funcional que otimize a gestão da sua clínica, reduza erros manuais, melhore a comunicação com os seus clientes e profissionalize o seu negócio. O sucesso do projeto impacta diretamente a eficiência e a lucratividade da sua empresa.

**Orientadora: Caroline Pacheco da Rosa**

* **Papel:** Acompanhar o progresso do projeto, fornecer orientação técnica e de gestão, avaliar as entregas e garantir que o projeto atende aos padrões académicos e objetivos da disciplina (utilizando metodologias ágeis).
* **Interesse:** Garantir que a equipa aplique corretamente os conceitos de gestão de projetos e desenvolvimento de software, que o projeto seja concluído com qualidade e que os alunos alcancem os resultados de aprendizagem esperados.

**Equipe do Projeto: Lucas, Roger, Patrick, Mateus, Augusto**

* **Papel:** Responsável por planejar, desenvolver, testar e entregar o sistema "Agenda VTA" de acordo com os requisitos definidos. Cada membro tem uma função específica que será trazida no tópico abaixo. (3.5 - Equipe do projeto) .
* **Interesse:** Concluir o projeto com sucesso para cumprir os objetivos académicos, adquirir experiência prática em desenvolvimento de software, trabalhar em equipe e construir um portfólio.

**Equipe da clínica/Usuários: Recepcionistas, Veterinários, Gestores.**

* **Papel:** São as pessoas que irão operar o sistema "Agenda Vet" no dia a dia para gerir agendamentos, registar clientes e animais, e comunicar com os donos dos animais.
* **Interesse:** Ter uma ferramenta fácil de usar, estável e eficiente que simplifique as suas tarefas diárias, reduza a carga de trabalho manual e minimize a possibilidade de erros. A sua aceitação e uso determinam o sucesso prático da solução.

#### 

## 

### 2.4 Equipe do Projeto e Responsabilidades

## A equipe do projeto Agenda VTA foi organizada de acordo com as habilidades e funções necessárias para garantir o sucesso do desenvolvimento. Cada integrante possui atribuições específicas que orientam sua atuação durante todas as etapas do projeto.

## 

| Patrick Vargas  Líder de Projeto e Contato com o Cliente | |
| --- | --- |
|  | * Coordenar a equipe e distribuir tarefas. * Ser o principal ponto de contato com o cliente, garantindo que as necessidades sejam compreendidas e atendidas. * Acompanhar o progresso do time e mediar decisões estratégicas. * Validar entregas antes do envio ao orientador e cliente. |

## 

| Roger VeselyDocumentação e Organização | |
| --- | --- |
|  | Produzir e manter atualizada a documentação oficial do projeto (TAP, EAP, relatórios e planos de gerenciamento).Consolidar versões e registros de alterações no documento do projeto.Apoiar a equipe na organização de informações e padronização de arquivos.Garantir clareza e consistência entre entregas técnicas e documentais. |

## 

## 

| Augusto AzambuyaResponsável Técnico de Back-End | |
| --- | --- |
|  | Implementar e manter a lógica de negócio e serviços do sistema.Projetar e gerenciar a integração com o banco de dados.Garantir a segurança e confiabilidade da aplicação no lado do servidor.Apoiar os demais integrantes na solução de problemas técnicos. |

## 

| Mateus FranceschetResponsável Técnico de Front-End | |
| --- | --- |
|  | Desenvolver a interface do usuário a partir dos protótipos definidos.Assegurar responsividade, usabilidade e consistência visual.Integrar o front-end com as funcionalidades de back-end.Trabalhar em conjunto com o time para alinhar experiência do usuário às jornadas definidas. |

| Lucas RamosComunicação e Apresentação | |
| --- | --- |
|  | Organizar a comunicação do grupo com a orientadora e stakeholders.Apoiar na mediação de reuniões e registros de decisões.Ser o responsável pela apresentação oficial do projeto nas bancas parciais e finais.Auxiliar na elaboração de materiais de apresentação e relatórios de acompanhamento. |

## 3 - Requisitos Funcionais e Não Funcionais

# **3.1 Tabela de Requisitos Funcionais - MVP Sistema Agenda VTA**

| ID | Módulo | Requisito | Descrição | Prioridade |
| --- | --- | --- | --- | --- |
| RF1.1 | Autenticação | Login do usuário | O sistema deve permitir login com email/usuário e senha | Alta |
| RF1.2 | Autenticação | Recuperação de senha | O sistema deve permitir recuperação de senha via email | Média |
| RF1.3 | Autenticação | Lembrar credenciais | O sistema deve oferecer opção 'Lembrar de mim' | Baixa |
| RF2.1 | Dashboard | Visão geral do sistema | O sistema deve exibir dashboard com estatísticas rápidas | Alta |
| RF2.2 | Dashboard | Agenda do dia | O sistema deve mostrar agendamentos do dia atual | Alta |
| RF2.3 | Dashboard | Notificações | O sistema deve exibir notificações e lembretes | Média |
| RF2.4 | Dashboard | Ações rápidas | O sistema deve fornecer atalhos para funções principais | Média |

| ID | Módulo | Requisito | Descrição | Prioridade |
| --- | --- | --- | --- | --- |
| RF3.1 | Agenda | Visualização por Salas | O sistema deve permitir visualização da agenda por Salas | Alta |
| RF3.2 | Agenda | Visualização por dia/semana | O sistema deve permitir visualização da agenda por dia ou semana | Alta |
| RF3.3 | Agenda | Navegação de períodos | O sistema deve permitir navegar entre datas (anterior/próxima/hoje) | Alta |
| RF3.4 | Agenda | Filtros de visualização | O sistema deve permitir filtrar por sala, profissional e status | Alta |
| RF3.5 | Agenda | Legenda de cores | O sistema deve usar cores para diferenciar status (livre/ocupado/bloqueado) | Média |
| RF3.6 | Agenda | Detalhes de agendamentos | O sistema deve exibir detalhes ao clicar/passar mouse sobre agendamentos | Alta |
| RF4.1 | Agendamentos | Criar novo agendamento | O sistema deve permitir criar novos agendamentos | Alta |
| RF4.2 | Agendamentos | Listar agendamentos | O sistema deve listar todos os agendamentos com filtros | Alta |
| RF4.3 | Agendamentos | Buscar agendamentos | O sistema deve permitir busca por cliente, pet ou serviço | Alta |
| RF4.4 | Agendamentos | Editar agendamento | O sistema deve permitir alterar agendamentos existentes | Alta |
| RF4.5 | Agendamentos | Cancelar agendamento | O sistema deve permitir cancelar agendamentos | Alta |
| RF4.6 | Agendamentos | Excluir agendamento | O sistema deve permitir excluir agendamentos (Com mensagem de confirmação) | Média |
| RF4.7 | Agendamentos | Filtros rápidos | O sistema deve oferecer filtros por data e status predefinidos | Média |

| ID | Módulo | Requisito | Descrição | Prioridade |
| --- | --- | --- | --- | --- |
| RF7.1 | Salas | Listar salas | O sistema deve listar todas as salas disponíveis | Alta |
| RF7.2 | Salas | Cadastrar sala | O sistema deve permitir cadastro de novas salas | Média |
| RF7.3 | Salas | Editar sala | O sistema deve permitir alteração de dados da sala | Média |
| RF7.4 | Salas | Bloquear sala | O sistema deve permitir bloqueio temporário de salas | Alta |
| RF7.5 | Salas | Status em tempo real | O sistema deve exibir status atual das salas | Alta |
| RF7.6 | Salas | Estatísticas de ocupação | O sistema deve mostrar métricas de ocupação | Média |
| RF8.1 | Usuários | Cadastrar usuário | O sistema deve permitir cadastro de novos usuários | Alta |
| RF8.2 | Usuários | Listar usuários | O sistema deve listar todos os usuários do sistema | Alta |
| RF8.3 | Usuários | Editar usuário | O sistema deve permitir alteração de dados do usuário | Alta |
| RF8.4 | Usuários | Gerenciar permissões | O sistema deve permitir definir funções e permissões | Alta |
| RF8.5 | Usuários | Ativar/Inativar usuário | O sistema deve permitir controlar status de usuários | Alta |
| RF8.6 | Usuários | Buscar usuários | O sistema deve permitir busca por nome, email ou função | Média |

### 

| ID | Módulo | Requisito | Descrição | Prioridade |
| --- | --- | --- | --- | --- |
| RF9.1 | Relatórios | Gerar relatórios gerenciais | O sistema deve gerar relatórios com KPIs do período | Média |
| RF9.2 | Relatórios | Filtrar por período | O sistema deve permitir filtrar relatórios por data | Média |
| RF9.3 | Relatórios | Exportar dados | O sistema deve permitir exportação em CSV | Baixa |
| RF9.4 | Relatórios | Gráficos de ocupação | O sistema deve exibir gráficos de agendamentos por sala | Baixa |
| RF10.1 | Geral | Controle de sessão | O sistema deve gerenciar sessões de usuário e logout | Alta |
| RF10.2 | Geral | Feedback de ações | O sistema deve exibir mensagens de sucesso/erro para todas as operações | Alta |

## 

## 3.2 Requisitos Não Funcionais

| ID | Categoria | Requisito | Descrição | Critério de Aceitação |
| --- | --- | --- | --- | --- |
| RN1.1 | Performance | Tempo de resposta | O sistema deve responder em no máximo 3 segundos | Operações críticas (ex: carregar agenda, salvar agendamento) devem responder em < 3s. O sistema deve fornecer feedback durante operações longas. |
| RN2.1 | Usabilidade | Interface intuitiva | O sistema deve ser fácil de usar sem treinamento extensivo | Usuário consegue usar funções básicas em < 10 min |
| RN2.2 | Usabilidade | Responsividade | O sistema deve funcionar em desktop, tablet e mobile | Layout adaptável em todas as resoluções |
| RN2.3 | Usabilidade | Feedback visual | Todas as ações devem ter feedback visual imediato | Confirmação imediata para ações críticas (ex: salvar, editar, excluir) |
| RN3.1 | Segurança | Controle de acesso | O sistema deve controlar acesso por perfil/função | Permissões por papel de usuário |
| RN3.2 | Segurança | Proteção de dados | Os dados devem ser protegidos contra acesso não autorizado | Criptografia de dados sensíveis |
| RN3.3 | Segurança | Sessões seguras | As sessões devem expirar após inatividade | Timeout de sessão em 30 min |
| RN4.1 | Confiabilidade | Backup de dados | Os dados devem ter backup automático diário | Backup diário automatizado |
| RN5.1 | Compatibilidade | Navegadores | O sistema deve funcionar nos principais navegadores | Chrome, Firefox |
| RN5.2 | Compatibilidade | Sistemas operacionais | O sistema deve funcionar em Windows e Mac. | Cross-platform compatibility |

### 

### 

### 

**3.6 - Critérios de Sucesso**

### Para que o projeto Agenda VTA seja considerado um sucesso, os seguintes critérios mensuráveis devem ser alcançados:

* **Entrega do MVP com Funcionalidades Críticas:** Implementação de 100% dos requisitos funcionais essenciais para a operação da clínica, incluindo:
  + Módulo de autenticação (UC01).
  + CRUD de Clientes e Pets (UC02).
  + Visualização da agenda por salas (dia/semana) (UC03).
  + Realização de agendamentos e bloqueios de horários (UC04, UC05).
  + Geração de texto-ticket para envio manual de confirmação via WhatsApp.
* **Validação e Aceitação pelo Cliente:** Aprovação formal do protótipo funcional pelo cliente (Sr. Tiago Siqueira Pereira) até o final do terceiro mês de projeto, validando que os fluxos de agendamento e visualização atendem às necessidades da recepção.
* **Redução Drástica de Erros Operacionais:** Eliminação completa de falhas de agendamento por sobreposição de horários e salas no sistema, um dos principais problemas do processo manual atual, medido após os primeiros 30 dias de uso contínuo.
* **Adoção e Eficiência da Equipe:** Atingir uma nota de satisfação mínima de 8/10 da equipe de recepção (usuários principais) em pesquisa aplicada após a implantação. O tempo médio para realizar um novo agendamento não deve ultrapassar 60 segundos.
* **Cumprimento do Cronograma:** Concluir e implantar a versão estável do sistema dentro dos prazos estabelecidos no cronograma do projeto, garantindo a entrega no Marco 3 (Implementação Completa na Clínica).

### 4. **Estrutura Analítica do Projeto (EAP**)

### **1.0 Gerenciamento e Planejamento do Projeto**

1.1 Iniciação e Documentação Principal

1.1.1 Elaboração do Termo de Abertura do Projeto (TAP)

1.1.2 Definição e Validação do Escopo (MVP)

1.1.3 Elaboração da EAP, Cronograma e Planos de Gerenciamento

1.2 Gestão Contínua

1.2.1 Realização de Reuniões de Acompanhamento Semanal

1.2.2 Controle de Versão dos Documentos e Artefatos

1.2.3 Encerramento Formal do Projeto e Lições Aprendidas

**2.0 Análise e Design (UX/UI)**

2.1 Levantamento de Requisitos

2.1.1 Entrevistas com Stakeholders (Cliente e Usuários)

2.1.2 Análise do Processo Atual (Planilhas Excel)

2.2 Design da Solução

2.2.1 Mapeamento de Fluxos de Usuário (Login, Agendamento, Bloqueio)

2.2.2 Criação de Wireframes e Protótipos de Baixa/Alta Fidelidade (Figma) 999

2.2.3 Validação do Protótipo com o Cliente

2.3 Arquitetura Técnica

2.3.1 Definição da Arquitetura do Sistema e Stack Tecnológica

2.3.2 Modelagem do Banco de Dados (PostgreSQL)

**3.0 Desenvolvimento e Implementação**

3.1 Configuração do Ambiente

3.1.1 Estruturação do Repositório de Código (Git)

3.1.2 Configuração do Ambiente de Desenvolvimento Local e Servidor

3.2 Desenvolvimento do Back-end (Python/Flask)

3.2.1 Implementação da API e Rotas Principais

3.2.2 Configuração da Conexão com Banco de Dados

3.2.3 Implementação do Módulo de Autenticação e Permissões (UC01)

3.2.4 Implementação das Regras de Negócio (CRUD Clientes/Pets, Agendamentos, Bloqueios)

3.3 Desenvolvimento do Front-end (HTML/CSS/JS)

3.3.1 Estruturação do Layout Base (Topo e Menu Lateral Fixos)

3.3.2 Implementação da Tela de Login (UC01)

3.3.3 Desenvolvimento da Tela de Agenda (Visualização Dia/Semana por Sala) (UC03)

3.3.4 Implementação dos Modais de Agendamento e Bloqueio (UC04, UC05)

3.3.5 Desenvolvimento das Telas de CRUD (Clientes, Pets, Usuários) (UC02)

3.3.6 Desenvolvimento das Telas de Relatórios (UC07, UC08)

3.4 Integração

3.4.1 Conexão das Telas (Front-end) com a API (Back-end)

3.4.2 Implementação de Feedback Visual (Mensagens de Sucesso/Erro) (UCs)

**4.0 Testes, Validação e Homologação**

4.1 Testes da Equipe

4.1.1 Testes Funcionais (Verificar se os requisitos foram atendidos)

4.1.2 Testes de Integração (Front-end + Back-end)

4.2 Validação com o Cliente

4.2.1 Testes de Usabilidade com Usuários Finais (Recepção)

4.2.2 Coleta e Análise de Feedback

4.3 Correção de Bugs

4.3.1 Triagem e Priorização dos Bugs Encontrados

4.3.2 Ciclos de Correção e Re-teste

**5.0 Implantação, Treinamento e Entrega Final**

5.1 Preparação do Ambiente de Produção

5.1.1 Configuração do Servidor de Hospedagem (Render/Railway)

5.1.2 Configuração de Domínio e Backup

5.2 Implantação (Go-Live)

5.2.1 Migração de Dados Iniciais (se aplicável)

5.2.2 Deploy da Versão Estável do Sistema

5.3 Treinamento e Documentação

5.3.1 Criação de Material de Treinamento (Tutorial em Vídeo/Texto)

5.3.2 Sessão de Treinamento com a Equipe da Clínica

5.3.3 Entrega da Documentação Final do Projeto

5.4 Entrega Final

5.4.1 Apresentação Final do Projeto

5.4.2 Entrega do Repositório de Código e Artefatos

**5. Plano de Gerenciamento de Custos (Versão Revisada)**

Este plano detalha a estrutura de custos do projeto Agenda VTA, dividida em duas fases distintas: a fase de desenvolvimento, que é integralmente subsidiada, e a fase de manutenção pós-entrega, cujos custos operacionais serão de responsabilidade da clínica.

#### 

#### **5.1. Fase de Desenvolvimento (Custo Zero para a Clínica)**

Durante os seis meses da "Etapa de Imersão em Computação Avançada", o desenvolvimento do sistema

**não terá custos diretos** para a clínica Vet Assistance. Todos os recursos necessários são subsidiados pelo programa Residência em TIC BRISA-FULBRA, conforme detalhado abaixo:

1. **Recursos Humanos:** Cobertura integral das bolsas de estudo para a equipe de cinco desenvolvedores dedicados ao projeto.
2. **Equipamentos:** Fornecimento de notebooks em regime de comodato para cada membro da equipe, garantindo a infraestrutura de trabalho.
3. **Software e Ferramentas:** Utilização de planos gratuitos de ferramentas essenciais para o projeto, como Figma para design e as plataformas Render ou Railway para hospedagem e testes, não gerando despesas adicionais.

#### 

#### **5.2. Custos de Manutenção (Pós-desenvolvimento)**

#### Após o período de 6 meses de desenvolvimento subsidiado, a manutenção da plataforma em ambiente de produção para a clínica incorrerá nos seguintes custos mensais, que serão de responsabilidade do cliente:

| **Categoria** | **Ferramenta/Serviço** | **Custo Mensal Estimado** | **Justificativa e Observações** |
| --- | --- | --- | --- |
| **Infraestrutura Essencial** | Domínio Personalizado | R$ 4,00 | Garante um endereço web profissional para a clínica (ex: www.agendavetassistance.com.br). O valor anual foi diluído para uma estimativa mensal. |
| **Servidor** | Hospedagem (Backend + Banco de Dados) | R$110,00 | Indispensável para manter o sistema online 24/7. Cobre o servidor da aplicação e o banco de dados. Plataformas: Railway, Render. |
| **Comunicação Recomendada** | Serviço de E-mail Transacional | Plano Gratuito | Útil para enviar relatórios ou como uma alternativa de notificação. A maioria dos serviços oferecem um plano gratuito robusto. Plataformas: SendGrid, Brevo. |
| **Monitoramento Recomendado** | Monitoramento de Erros | Plano Gratuito | Permite que a equipe de desenvolvimento identifique e corrija falhas no sistema rapidamente, muitas vezes antes que o usuário perceba. Plataformas: Sentry. |
| **Custo Total:** |  | R$ 114,00 / mês | Os serviços recomendados possuem planos gratuitos que atendem à demanda inicial, não adicionando custo imediato. |

### 6 - Plano de Gerenciamento de Tempo

### 6.1 Tabela de Gerenciamento de Tempo do Projeto.

| **Fase:** | **Início** | **Fim** | **Responsável** | **Atividade** |
| --- | --- | --- | --- | --- |
| **Fase 1:** | **Julho** | **Agosto** |  | **Análise e Design** |
| 1.1. | 14/07/25 | 26/07/25 | Roger | Elaboração do TAP, EAP e Planos |
| 1.2. | 21/07/25 | 08/08/25 | Patrick | Levantamento de Requisitos com Cliente |
| 1.3. | 04/08/25 | 22/08/25 | Mateus | Design da Interface (Protótipos no Figma) |
| 1.4. | 11/08/25 | 29/08/25 | Augusto | Modelagem do Banco de Dados (PostgreSQL) |
| 1.5. | 18/08/25 | 29/08/25 | Equipe | Definição da Arquitetura do Sistema |
| **Marco 1:** | **29/08/25** | **29/08/25** | **Equipe** | **Planejamento e Arquitetura Concluídos** |
| **Fase 2:** | **Agosto** | **Outubro** |  | **Desenvolvimento** |
| 2.1. | 25/08/25 | 26/09/25 | Mateus | Desenvolvimento do Front-end (HTML/CSS/JS) |
| 2.2. | 01/09/25 | 03/10/25 | Augusto | Desenvolvimento do Back-end (Python Flask, DB) |
| 2.3. | 22/09/25 | 17/10/25 | Patrick | Integração das Funcionalidades Principais |
| **Marco 2:** | **31/10/25** | **31/10/25** | **Equipe** | **Protótipo Funcional Navegável Entregue** |
| **Fase 3:** | **Outubro** | **Novembro** |  | **Testes e Validação** |
| 3.1. | 27/10/25 | 14/11/25 | Equipe | Testes Funcionais e de Navegação |
| 3.2. | 03/11/25 | 14/11/25 | Lucas | Testes de Usabilidade com o Cliente |
| 3.3. | 10/11/25 | 28/11/25 | Equipe | Correção de Bugs e Ajustes Finais |
| 3.4. | 24/11/25 | 29/11/25 | Patrick | Preparação do Ambiente e Implantação |
| **Marco 3:** |  | **30/11/25** | **Equipe** | **Implementação Completa na Clínica** |
| **Fase 4:** | **Dezembro** | **Dezembro** |  | **Entrega Final** |
| 4.1. | 01/12/25 | 14/12/25 | Augusto/Patrick | Desenvolvimento do Dashboard Gerencial |
| 4.2. | 15/12/25 | 21/12/25 | Lucas / Roger | Entrega da Documentação e Tutorial |
| 4.3. | 22/12/25 | 28/12/25 | Equipe | Entrega do Pacote Final (Vídeo e Repositório) |

### 6.2 - Tabela de Atividades Primeiro Trimestre

| **Tarefa Atividade** | **Data Início** | **Data Término** | **Descrição** |
| --- | --- | --- | --- |
| **Julho:** | **07/07** | **28/07** |  |
| Semana 1 | 07/07 | 13/07 | Apresentação da proposta de imersão, nivelamento das expectativas e definição dos temas. |
| Semana 2 | 14/07 | 19/07 | Gerência de Projetos: Elaboração do Termo de Abertura do Projeto (TAP). |
| Semana 3 | 21/07 | 26/07 | Análise do problema, cenário atual, stakeholders e delimitação do escopo. Elaboração da Estrutura Analítica do Projeto com Plano de Gerenciamento de Riscos, Custo e Tempo. |
| Semana 4 | 28/07 | 02/08 | Laboratório de Criatividade: Técnicas de ideação, construção de personas, brainstorming e mapa de empatia. |
| **Agosto:** | **02/08** | **30/08** |  |
| Semana 5 | 04/08 | 09/08 | Laboratório de Criação: Jornada do usuário, proposta de valor, fluxogramas, jornadas de usuário, requisitos iniciais, definição de MVP. |
| Semana 6 | 11/08 | 16/08 | Prototipagem inicial (rascunho digital, wireframes, mockups). |
| Semana 7 | 18/08 | 23/08 | Validação dos protótipos com colegas e início da modelagem técnica da solução. |
| Semana 8 | 25/08 | 30/08 | Arquitetura técnica, definição de tecnologias, modelagem de dados e estrutura do código. |
| **Setembro:** | **01/09** | **29/09** |  |
| Semana 9 | 01/09 | 06/09 | Início da implementação: configurações iniciais, estruturação do repositório. |
| Semana 10 | 08/09 | 13/09 | Desenvolvimento do front-end básico (interfaces e navegação inicial). |
| Semana 11 | 15/09 | 20/09 | Integração com banco de dados e back-end inicial. |
| Semana 12 | 22/09 | 27/09 | Implementação de funcionalidades principais (core do sistema). |
| Semana 13 | 29/09 | 04/10 | Testes e ajustes da primeira versão funcional. Preparação para avaliação parcial. |

### 

### 6.3 - Tabela de Atividades Segundo Trimestre.

| **Tarefa Atividade** | **Data Início** | **Data Término** | **Descrição** |
| --- | --- | --- | --- |
| **Outubro:** | **06/10** | **11/10** |  |
| Semana 1 | 06/10 | 11/10 | Análise dos feedbacks da avaliação parcial e revisão de rota de desenvolvimento. |
| Semana 2 | 13/10 | 18/10 | Continuação do desenvolvimento: aprimoramento do front-end e integração. |
| Semana 3 | 20/10 | 25/10 | Desenvolvimento de funcionalidades complementares e melhorias de usabilidade. |
| Semana 4 | 27/10 | 01/11 | Implementação de segurança, testes de navegação e responsividade. |
| **Agosto:** | **03/11** | **08/11** |  |
| Semana 5 | 03/11 | 08/11 | Documentação técnica inicial: requisitos implementados, estrutura de banco, fluxos. |
| Semana 6 | 10/11 | 15/11 | Preparação do pitch final e estruturação da apresentação. |
| Semana 7 | 17/11 | 22/11 | Simulado de apresentação final com banca simulada e revisão da documentação. |
| Semana 8 | 24/11 | 30/11 | Ajustes pós-simulados: melhorias técnicas, visuais e de comunicação. |
| **Setembro:** | **01/12** | **07/12** |  |
| Semana 9 | 01/12 | 07/12 | Finalização da implementação e início da preparação da entrega oficial. |
| Semana 10 | 08/12 | 14/12 | Testes finais e revisão completa (código, documentação, artefatos). |
| Semana 11 | 15/12 | 21/12 | Montagem do pacote de entrega final (repositório GitHub, guia de instalação, pitch). |
| Semana 12 | 22/12 | 28/12 | Atividade prática: cada grupo deve entregar um vídeo de até 3 minutos com o resumo do projeto (pitch). |
| Semana 13 | 29/12 | 04/01 | Revisão individual final com orientador e organização para a avaliação final. |

### 

### 7 - Plano de Gerenciamento de Riscos

Este tópico descreve de forma detalhada os principais riscos identificados no projeto Agenda VTA, destacando possíveis cenários realistas que poderiam afetar a execução ou qualidade da entrega, bem como estratégias adequadas de mitigação.

## Risco 1: Complexidade técnica do stack Flask + PostgreSQL

**Descrição:** A equipe pode enfrentar dificuldades significativas na implementação da arquitetura Python/Flask com PostgreSQL, especialmente na integração entre frontend e backend, gerenciamento de sessões e queries complexas para a visualização por salas.

**Análise de Risco:**

* Probabilidade: Alta
* Impacto: Muito Alto
* Categoria: Competência Técnica
* Tempo de Detecção: Nas primeiras 2-3 semanas de desenvolvimento
* Tempo de Recuperação: 2-4 semanas (com impacto no cronograma)

**Estratégia de Mitigação:**

1. Implementar treinamentos intensivos em Flask e PostgreSQL nas primeiras semanas
2. Criar protótipos simples para validar a arquitetura antes do desenvolvimento completo
3. Manter documentação técnica detalhada de todas as decisões arquiteturais

## 

## Risco 2: Dificuldades na implementação da funcionalidade core (visualização por salas)

**Descrição:** A visualização da agenda por salas em tempo real, sendo o principal diferencial do sistema, pode apresentar complexidade técnica elevada para uma equipe iniciante, especialmente na sincronização de dados e interface responsiva.

**Análise de Risco:**

* Probabilidade: Alta
* Impacto: Muito Alto
* Categoria: Funcionalidade Principal
* Tempo de Detecção: Durante o desenvolvimento do frontend
* Tempo de Recuperação: 1-3 semanas

**Estratégia de Mitigação:**

1. Priorizar o desenvolvimento da visualização por salas como primeira funcionalidade
2. Criar wireframes detalhados e validar a lógica de negócio antes da implementação
3. Implementar versão simplificada primeiro (apenas visualização) antes de adicionar interatividade

## 

## Risco 3: Problemas na migração de dados das planilhas Excel para PostgreSQL

**Descrição:** A migração das planilhas atuais do Excel para o banco PostgreSQL pode gerar perdas, inconsistências ou corrupção de dados, prejudicando o funcionamento inicial e a confiança dos usuários no sistema.

**Análise de Risco:**

* Probabilidade: Média
* Impacto: Alto
* Categoria: Dados
* Tempo de Detecção: Durante testes de migração
* Tempo de Recuperação: 3-7 dias

**Estratégia de Mitigação:**

1. Realizar análise completa da estrutura atual das planilhas antes do desenvolvimento
2. Criar múltiplos backups das planilhas originais em formatos diferentes
3. Testar a migração em ambiente controlado com dados de exemplo primeiro

## Risco 4: Inexperiência da equipe com metodologias ágeis e gestão de projeto

**Descrição:** Uma equipe iniciante pode ter dificuldades em seguir as metodologias ágeis propostas, gerenciar tarefas, cumprir sprints e manter comunicação efetiva, resultando em atrasos e desorganização.

**Análise de Risco:**

* Probabilidade: Muito Alta
* Impacto: Médio
* Categoria: Gestão de Projeto
* Tempo de Detecção: Nas primeiras entregas
* Tempo de Recuperação: 2-4 semanas

**Estratégia de Mitigação:**

1. Implementar reuniões diárias curtas (15 min) para acompanhamento
2. Utilizar ferramentas visuais como Kanban para gestão de tarefas
3. Estabelecer marcos semanais pequenos e alcançáveis
4. Definir claramente papéis e responsabilidades de cada membro
5. Realizar retrospectivas regulares para ajustar o processo

## 

## Risco 5: Falhas de segurança e violação da LGPD

**Descrição:** Devido à uma possível inexperiência da equipe, o sistema pode apresentar vulnerabilidades graves como armazenamento inadequado de senhas, falta de controle de acesso adequado, ou exposição de dados sensíveis dos clientes da clínica.

**Análise de Risco:**

* Probabilidade: Média
* Impacto: Muito Alto
* Categoria: Segurança e Conformidade
* Tempo de Detecção: Variável (imediato a meses após deploy)
* Tempo de Recuperação: Indefinido (pode incluir consequências legais)

**Estratégia de Mitigação:**

1. Implementar checklist de segurança obrigatório antes de cada release
2. Utilizar bibliotecas estabelecidas para autenticação (Flask-Login, Werkzeug)
3. Realizar code review focado em segurança para todas as funcionalidades críticas
4. Implementar criptografia de senhas desde o início do desenvolvimento
5. Contratar auditoria de segurança externa antes do go-live

## 

## Risco 6: Baixa aceitação pelos usuários finais (recepcionistas e veterinários)

**Descrição:** A equipe da clínica pode resistir à mudança do Excel para o novo sistema, especialmente se a interface não for intuitiva ou se houver curva de aprendizado elevada para uma equipe não-técnica.

**Análise de Risco:**

* Probabilidade: Baixa
* Impacto: Médio
* Categoria: Mudança Organizacional
* Tempo de Detecção: Durante testes com usuários ou pós-implantação
* Tempo de Recuperação: 2-6 semanas

**Estratégia de Mitigação:**

1. Envolver usuários finais desde a fase de prototipagem
2. Realizar testes de usabilidade semanais durante o desenvolvimento
3. Criar tutoriais em vídeo específicos para cada funcionalidade

### 

### 8. Considerações Finais

O projeto Agenda VTA foi concebido para solucionar um desafio crítico na rotina da clínica Vet Assistance: a substituição de um sistema de agendamento baseado em planilhas por uma plataforma centralizada, estável e segura. A dependência de um método manual gerava riscos operacionais significativos, como a exclusão acidental de dados, a falta de controle sobre alterações e a instabilidade no acesso simultâneo, impactando diretamente a eficiência da recepção.

A solução desenvolvida priorizou as necessidades do usuário principal — a equipe de recepção, entregando um sistema que valoriza a agilidade, simplicidade e, acima de tudo, a estabilidade. O foco no essencial permitiu construir uma ferramenta robusta, que atende diretamente às dores mapeadas, como a implementação da confirmação de exclusão em duas etapas, um requisito fundamental para mitigar erros humanos.

O diferencial do sistema, a visualização da agenda por salas, foi implementado como a funcionalidade central, otimizando a organização dos atendimentos e o uso da infraestrutura da clínica. Além disso, a decisão de não automatizar o envio de mensagens via WhatsApp, mas sim gerar um texto-ticket padronizado, respeitou o fluxo de trabalho existente e deu à equipe de atendimento total controle sobre a comunicação com o cliente, unindo a praticidade do sistema à pessoalidade do contato humano.

O projeto também entregou funcionalidades de valor agregado, como a geração de relatórios gerenciais e o histórico de atendimentos por pet, que fornecem dados para uma gestão mais informada.

Ao final deste ciclo, o Agenda VTA se estabelece como uma base sólida e confiável para a operação da clínica. A arquitetura planejada e o foco nos requisitos essenciais abrem caminho para futuras expansões, como uma eventual versão mobile, garantindo que o sistema possa evoluir junto com as necessidades da Vet Assistance.

### 9- Controle de Versões do Documento

### 

| Versão | Data | Alterações | Descrição da Alteração | Arquivo da Versão em PDF |
| --- | --- | --- | --- | --- |
| 1.0 | 22/07/25 | Correção | Versão inicial | [Projeto Agenda VTA - Versão 1.0](https://drive.google.com/file/d/1SyxznO21xpO1r1lZsopPm9MpcVCgPGFT/view?usp=drive_link) |
| 1.1 | 30/07/25 |  |  | [Projeto Agenda VTA - Versão 1.1](https://drive.google.com/file/d/1zh3jod1Jcb6YDt9_-1hgBqRHDNiPzAYJ/view?usp=drive_link) |
| 2.1 | 11/08/25 | Versão com MPV Atualizada |  | [Projeto Agenda VTA - Versão 2.0](https://drive.google.com/file/d/17RDhsrfWJPTq9BkSc-oFOfWIwzQO7TRN/view?usp=drive_link) |

### 

### 

### 10 - Área de Assinaturas

**Membros do Grupo**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Augusto Azambuya Moretto da Silva Lucas Ramos Alves

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mateus Franceschet Pereira Patrick Vargas Santos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roger Luiz do Nascimento Vesely

**Orientador Responsável Cliente / Empresa Parceira:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Caroline Pacheco da Rosa Tiago Siqueira Pereira