



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**  
Escola Politécnica  
Ciência de Dados e Inteligência Artificial

GRUPO: HEMOCENTRO

**HEMOCAMP**

**Disciplina:** Programação Web  
**Professor:** Fernando Luiz de Almeida Silveira

**RELATÓRIO DO PROJETO INTEGRADOR**

**INTEGRANTES**

Amanda Oliveira Piton (RA: 24019916)  
Giulia Feitosa de Souza (RA: 24785545)  
Giulia Monteiro Garrido (RA: 24010281)  
Mateus Andrade Antezana (RA: 24021000)  
Willian David Gomes da Silva Lana (RA: 24007368)



**CAMPINAS, 2025**

### **1.1. Levantamento de Requisitos**

Durante a fase de ideação, foram definidos os seguintes objetivos e funcionalidades:

- Centralizar informações sobre hemocentros (localização, horários, requisitos).
- Permitir cadastro e login de usuários.
- Coletar preferências de datas/horários de doação para gerar estatísticas.
- Oferecer login específico para colaboradores dos hemocentros parceiros.

### **Funcionalidades desejadas**

Para doadores:

- Consulta de informações atualizadas sobre hemocentros.
- Cadastro e login de usuários (doadores).
- Histórico pessoal de doações.

**Para colaboradores:**

- Sistema de login para colaboradores dos hospitais parceiros.
- Cadastro e gestão de campanhas.
- Registro de doações realizadas.

### **Restrições técnicas**

- Sistema web responsivo.
- Banco de dados relacional MySQL.
- Backend em Python com framework Flask.
- Criptografia de senhas com bcrypt.

### **Personas**

- **Doador:** pessoa que deseja contribuir com doação de sangue e precisa de informações claras.



- **Colaborador do hemocentro:** responsável por gerenciar dados, campanhas e estatísticas.

### Cenários de uso

- **Cenário 1 (Doador):** O usuário acessa a plataforma, realiza o cadastro com validação de CPF, faz login e consulta o mapa de hemocentros. Ao escolher uma unidade, verifica os horários disponíveis e agenda uma doação.
- **Cenário 2 (Colaborador):** Um funcionário do hemocentro faz login com credenciais corporativas (CNPJ vinculado). Ele acessa o painel administrativo e atualiza o estoque de sangue (ex: marcando "O-" como crítico).

### Requisitos funcionais

- **RF01:** Permitir cadastro e login de usuários.
- **RF02:** Permitir consulta de informações sobre hemocentros.
- **RF03:** Registrar preferências de horários de doação.
- **RF04:** Disponibilizar estatísticas para hemocentros.
- **RF05:** Registrar histórico de doações.
- **RF06:** Calcular automaticamente próxima data permitida para doação.
- **RF07:** Verificar elegibilidade do doador.
- **RF08:** Controlar acesso por tipo de usuário (doador/colaborador).

### Requisitos não funcionais

- **RNF01:** Interface responsiva.
- **RNF02:** Armazenamento seguro de dados de usuários.
- **RNF03:** Disponibilidade mínima de 99% do sistema.
- **RNF04:** Backup automático do banco de dados.

Além disso, foram definidos alguns objetivos para implementações futuras, como o envio de notificações para os usuários sobre campanhas via email e a possibilidade de contato entre usuário e hemocentro através do site HemoCamp.

## 1.2. Planejamento

### Escopo

O HemoCamp será um site que integra informações de hemocentros, campanhas e preferências de doação dos usuários, fornecendo dados estatísticos relevantes para gestores de saúde.

### Tecnologias e ferramentas

- Frontend: HTML, CSS, JavaScript.
- Backend: Python 3.12 com framework Flask, Bcrypt cost factor 12 para criptografia de senhas.
- Banco de dados: MySQL 8.0.0, MySQL Connector Python 8.2.0 para integração com o backend. Hospedagem no serviço de nuvem Railway.
- Controle de versão: GitHub.
- Gerenciamento de projeto: Jira (metodologia Kanban), MySQL Workbench para visualização e testes do banco.

### Metodologia

O projeto foi estruturado em sprints quinzenais, seguindo a metodologia Kanban.

### Cronograma de atividades:

Sprint	Início - Fim	Atividades
Sprint 1	19/08/2025 - 26/08/2025	Levantamento de requisitos e modelagem do banco
Sprint 2	26/08/2025 - 09/09/2025	Começo da criação das telas frontend
Sprint 3	09/09/2025 - 23/09/2025	Desenvolvimento do backend ( <i>models e routes</i> )
Sprint 4	23/09/2025 - 07/10/2025	Criação do Banco de Dados
Sprint 5	07/10/2025 - 30/10/2025	Integração frontend-backend e testes

Sprint 6	30/10/2025 - 13/11/2025	Ajustes de telas e backend
Sprint 7	13/11/2025 - 27/11/2025	Realização de testes, correções e ajustes finais

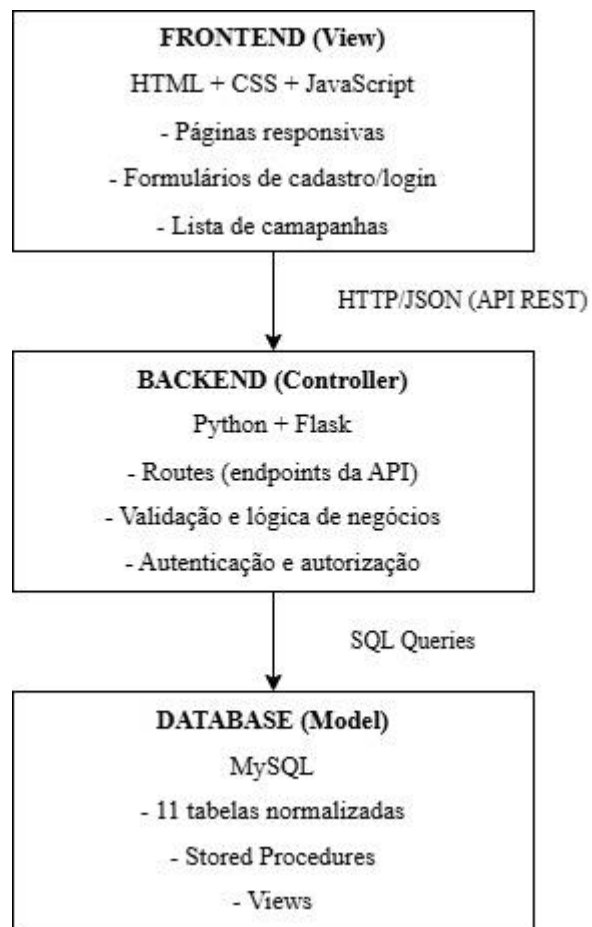
### 1.3. Desenvolvimento

#### Arquitetura do Sistema

O desenvolvimento do HemoCamp foi pensado de forma a manter o sistema organizado e fácil de manter. Para isso, a aplicação foi dividida em três camadas principais: interface com o usuário, lógica do sistema e banco de dados.

O sistema foi desenvolvido utilizando a arquitetura **MVC (Model-View-Controller)** adaptada para web, priorizando a modularização e a escalabilidade. A estrutura do código foi organizada da seguinte forma:

- **Backend (Controller/Model):** Desenvolvido em Python 3.12 com o microframework Flask. A aplicação utiliza o padrão de *Blueprints* para segmentar as rotas por domínio funcional (ex: *usuario\_routes*, *agendamento\_routes*, *hemocentro\_routes*), facilitando a manutenção.
- **Banco de Dados (Model):** Utiliza MySQL 8.0, hospedado na plataforma em nuvem Railway. A comunicação é feita via conector nativo *mysql-connector-python*, com classes de modelo (*models.py*) que encapsulam as queries SQL e regras de negócio. Foram criadas 4 views e 5 stored procedures para implementações futuras.
- **Frontend (View):** A interface é renderizada no servidor (*Server-Side Rendering*). O design visual foi construído com CSS3 nativo (sem frameworks pesados), garantindo leveza e personalização, com interatividade gerida por JavaScript para validações em tempo real no navegador.



## Estrutura do Banco de Dados

O banco de dados foi modelado com **11 tabelas principais** seguindo normalização até a 3ª Forma Normal (3FN):

### 1. Usuario

- Armazena doadores e colaboradores
- Campos: id, nome, email, senha (hash bcrypt), CPF, CNPJ (opcional), tipo\_usuario, tipo\_sanguineo, data\_nascimento
- Validações: CPF único, email único, senha criptografada

### 2. Hemocentros

- Cadastro de hemocentros parceiros
- Campos: id, nome, CNPJ, endereco, cidade, telefone, email, site
- Validações: CNPJ único e válido

### 3. HorarioFuncionamento

- Horários por dia da semana



- Relação 1:N com Hemocentros
- Constraint UNIQUE (id\_hemocentro, dia\_semana)

#### **4. Campanha**

- Campanhas de doação ativas
- Campos: nome, descrição, datas, tipo\_sanguineo\_necessario, metas, destaque
- Relação N:1 com Hemocentros

#### **5. Agendamento**

- Agendamentos de doação
- Status: pendente, confirmado, realizado, cancelado
- Relações com Usuario, Hemocentro e Campanha

#### **6. HistoricoDoacoes**

- Registro completo de doações
- Cálculo automático da próxima doação permitida
- Relações com Usuario e Hemocentro

#### **7. Notificacao**

- Futuro sistema de notificações por e-mail
- Tipos: campanha, agendamento, lembrete, urgente
- Controle de envio (enviado: true/false)

#### **8. PreferenciaDoacao**

- Preferências de dias e horários dos doadores
- Futuramente para uso de estatísticas e planejamento

#### **9. Contato**

- Mensagens enviadas pelo formulário do site
- Permite direcionamento a hemocentro específico

#### **10. Estoque**

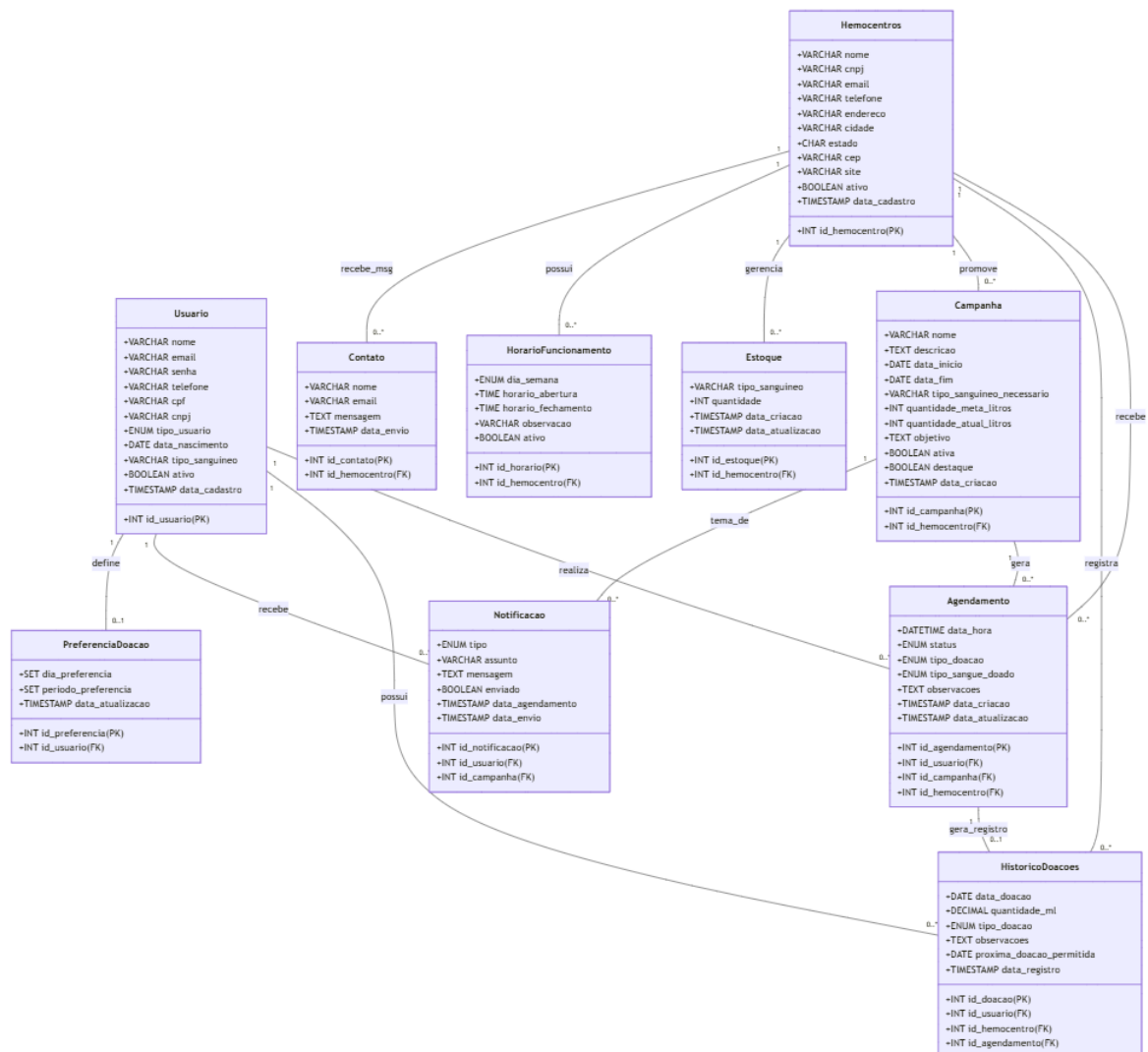
- Controla o inventário de bolsas de sangue de cada hemocentro
- Para auxiliar na tomada de decisão sobre campanhas

#### **11. AprovacaoTokens**

- Gerencia o fluxo de aprovação de novos cadastros (hemocentros e colaboradores)
- Status: pendente, aprovado, rejeitado

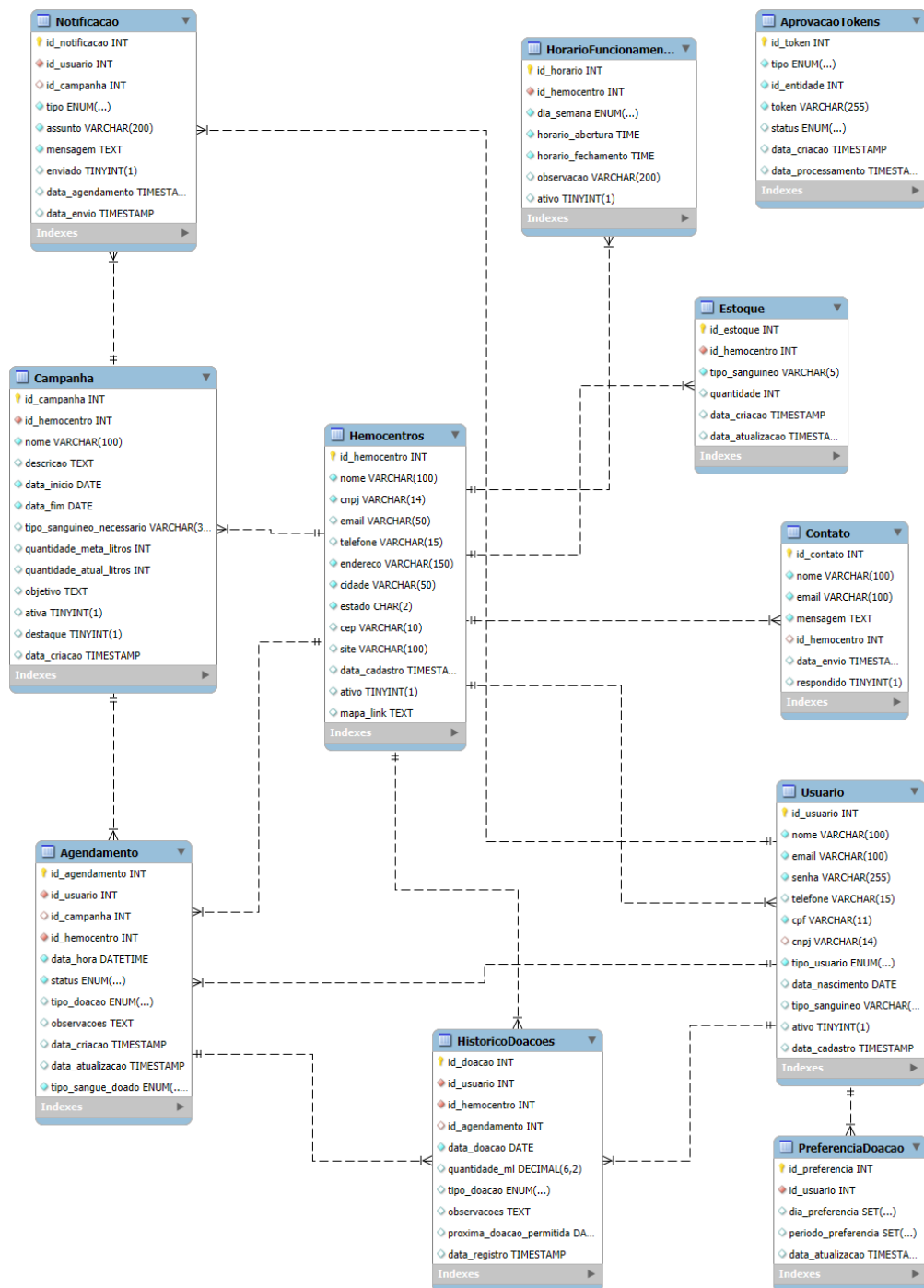
As tabelas “Notificacao” e “Contato” foram criadas visando as implementações futuras citadas anteriormente. Além disso, foram criadas 4 *Views* para facilitar consultas complexas (campanhas ativas, histórico completo, estatísticas gerais e próximos agendamentos.) e 5 *Stored Procedures* para operações comuns (calcular próxima doação, atualizar quantidade arrecadada por campanha, verificar elegibilidade do usuário, listar campanhas e gerar relatórios para os hemocentros).

## Diagrama:



DiagramaUML





Modelo Lógico

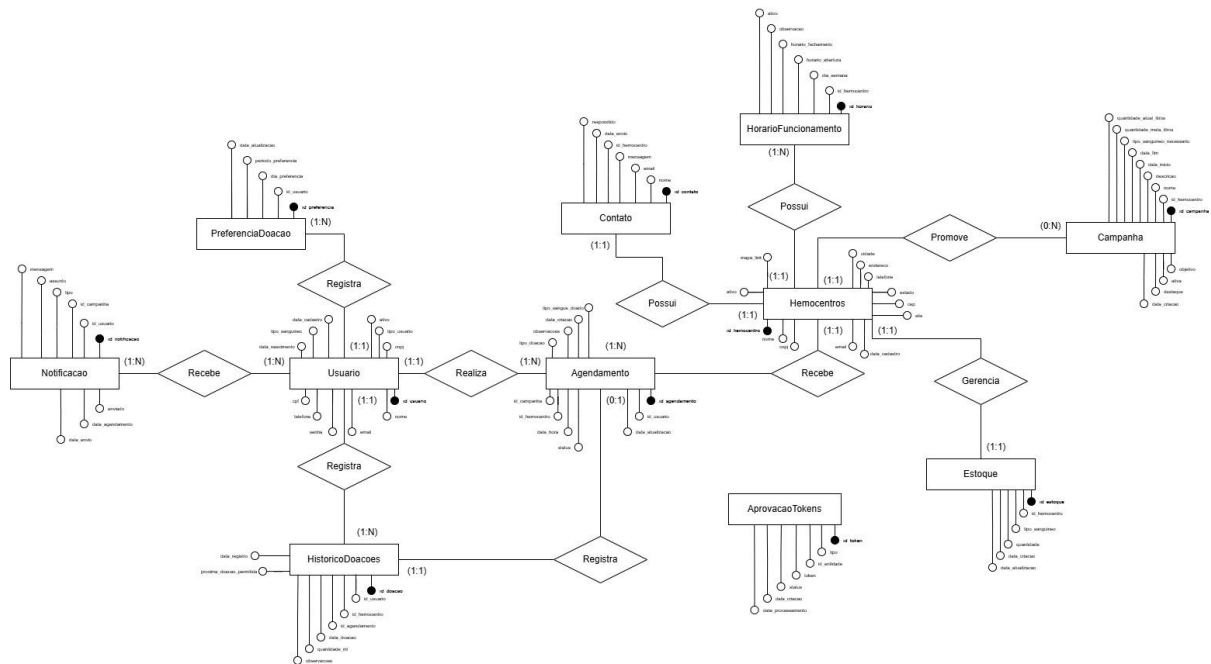


Diagrama DER

## Controle de versão

O projeto utiliza *Git* e *GitHub* para versionamento do código. Todas as alterações são registradas com *commits*, e o trabalho é organizado em *branches* para facilitar a colaboração entre os integrantes da equipe.

**Link do repositório:** <https://github.com/0-manda/Hemocentro>

## 1.4. Testes e Qualidade

Foram realizados testes para garantir que o aplicativo funcione corretamente em diferentes situações.

### Tipos de testes

Teste ou Bug	Tipo	Status
Login com credenciais corretas e incorretas.	Unitário	Feito

Tentativa de acesso a rotas protegidas sem autenticação.	Unitário	Feito
Verificação de duplicidade de email.	Unitário	Feito
Listagem de campanhas ativas.	Unitário	Feito
Cálculo de próxima data permitida para doação.	Unitário	Feito
Navegação intuitiva pelo site.	Exploratório	Feito
Responsividade em diferentes dispositivos.	Exploratório	Feito
Clareza das mensagens de erro.	Unitário	Feito
Amostragem de Estoque.	Exploratório	Bug não resolvido
Marcação de doação na mesma data para diferentes campanhas.	Exploratório	Não resolvido

### Ferramentas utilizadas

- Postman: testes de requisições HTTP para a API.
- MySQL Workbench: validação de queries e integridade dos dados.
- Chrome DevTools: debug do frontend e análise de requisições.

### Relatório de bugs e melhorias

Durante os testes, foram identificados e corrigidos os seguintes problemas:

1. **Bug:** Senha armazenada em texto plano inicialmente. **Solução:** Implementação de criptografia com bcrypt.
2. **Bug:** Campos desnecessários sendo solicitados nas rotas (como data\_cadastro, que é automático). **Solução:** Remoção dos campos autogerados das validações.

3. **Melhoria:** Mensagens de erro genéricas. **Solução:** Implementação de mensagens específicas para cada tipo de erro.

Os testes e bugs que ainda não foram corrigidos serão mapeados para correção futura.

## 1.5. Implantação e Manutenção

### Procedimentos de deploy

A implantação inicial do sistema está planejada para ser feita em servidores de hospedagem gratuitos ou de baixo custo:

- **Banco de dados:** Railway (até então na versão gratuita para o desenvolvimento do projeto).
- **Frontend e Backend:** O projeto está estruturado para containerização futura via Docker, permitindo o deploy em qualquer plataforma de nuvem que suporte containers (como AWS, Heroku ou Google Cloud Run).

### Futuros passos para deploy:

1. Configurar variáveis de ambiente no servidor (.env com credenciais do banco).
2. Fazer upload do código para o GitHub.
3. Conectar repositório com plataforma de hospedagem.
4. Configurar domínio personalizado (opcional).
5. Testar todas as funcionalidades em produção.

### Monitoramento de desempenho e erros

O sistema será monitorado através de:

- Logs automáticos de requisições HTTP.
- Registro de erros no banco de dados.
- Monitoramento de tempo de resposta das APIs.

### Estratégias de atualização e suporte futuras

- **Ciclo de revisões:** Semestralmente serão feitas revisões do código para correção de falhas e melhorias de segurança.
- **Atualizações:** Novas funcionalidades serão adicionadas conforme demanda dos usuários.
- **Suporte:** Canal de suporte por e-mail para auxiliar usuários em caso de dificuldades.

## 2. Considerações Finais

O HemoCamp representa uma iniciativa relevante para melhorar a comunicação entre hemocentros e doadores na região de Campinas. Mais do que um simples site informativo, a plataforma tem o potencial de se tornar um elo importante na rede de doação de sangue, ao oferecer uma experiência digital moderna, clara e interativa para a população.

O projeto alinha-se diretamente aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU, especificamente o ODS 3 (Saúde e Bem-Estar), ao buscar garantir o abastecimento de sangue e salvar vidas; o ODS 9 (Indústria, Inovação e Infraestrutura), por aplicar inovação tecnológica na gestão de saúde; e o ODS 17 (Parcerias e Meios de Implementação), ao fomentar a cooperação entre a comunidade acadêmica, a sociedade civil e as instituições de saúde. A iniciativa impacta diretamente o Primeiro Setor, auxiliando hemocentros públicos e o SUS, e o Segundo Setor, beneficiando hospitais e clínicas privadas que dependem desses estoques

Em longo prazo, a proposta pode ser expandida para abranger outras regiões e até mesmo ganhar uma versão mobile, fortalecendo ainda mais o impacto social. Dessa forma, o projeto cumpre seu objetivo de ampliar a conscientização, incentivar a solidariedade e ajudar a reduzir a escassez de sangue nos hospitais, contribuindo para salvar vidas.

O desenvolvimento do HemoCamp apresentou-se como uma experiência desafiadora e enriquecedora para toda a equipe. Este foi o primeiro projeto full-stack completo para a maioria dos integrantes, envolvendo desde a concepção da ideia até a implementação prática de um sistema web funcional.



Para melhor manejo de tempo, dividimos entre os integrantes áreas do desenvolvimento. Os alunos Giulia Feitosa e Willian se encarregaram do frontend, com foco no HTML e CSS. O aluno Mateus cuidou do JavaScript do projeto. O backend foi desenvolvido pela Amanda. A criação e gerenciamento do banco de dados foi encarregado à Giulia Monteiro.

Durante o desenvolvimento, alguns aspectos facilitaram o progresso do projeto. A escolha de Python como linguagem backend mostrou-se acertada, pois a equipe já possuía familiaridade com a sintaxe e conceitos básicos da linguagem, aprendidos em disciplinas anteriores. O framework Flask, por sua simplicidade e documentação clara, permitiu que compreendêssemos rapidamente os conceitos de rotas, requisições HTTP e construção de APIs.

A utilização do MySQL Workbench como ferramenta de modelagem e gestão do banco de dados facilitou significativamente a visualização das tabelas e relacionamentos. A interface gráfica permitiu testar queries de forma intuitiva e identificar problemas de estrutura antes mesmo de integrar com o backend. Além disso, a hospedagem do banco de dados no Railway mostrou-se uma solução prática e acessível, com configuração simplificada e recursos gratuitos suficientes para o desenvolvimento.

O trabalho em equipe também foi um ponto forte do projeto. A divisão de tarefas usando a metodologia Kanban no Jira ajudou a organizar o fluxo de trabalho, e o uso do GitHub para versionamento garantiu que todos pudessem contribuir sem conflitos graves no código.

Apesar dos pontos positivos, o projeto trouxe diversos desafios técnicos. A principal dificuldade foi compreender a comunicação entre frontend e backend. Inicialmente, houve confusão sobre como estruturar as requisições no JavaScript (usando fetch) e como o Flask deveria responder com dados JSON.

A segurança foi outro ponto de atenção. No início, as senhas estavam sendo armazenadas em texto puro, o que representa uma falha grave. Após pesquisas e orientações, implementamos a biblioteca bcrypt para criptografar as senhas de forma adequada.

Por fim, a própria modelagem do banco de dados passou por revisões. Inicialmente, não havíamos considerado campos importantes como CPF e CNPJ, nem pensado em como calcular



automaticamente a próxima data permitida para doação. A necessidade de adicionar índices surgiu conforme o sistema crescia e as consultas se tornavam mais complexas.

O projeto proporcionou aprendizados significativos em diversas áreas. No contexto de banco de dados, aprendemos na prática conceitos de normalização, relacionamentos (1:1, 1:N, N:M) e otimização de consultas através de índices. A criação de *views* e *stored procedures* mostrou como é possível encapsular lógica complexa diretamente no banco.

O uso de Python em um contexto profissional foi outro aprendizado importante. Conhecemos padrões de organização de código (separação em models, routes e utils), o uso de *decorators* para controle de acesso, context managers para gerenciamento de recursos (conexões com banco) e tratamento adequado de exceções.

Além dos aspectos técnicos, o projeto desenvolveu habilidades importantes como trabalho em equipe, comunicação técnica, resolução de problemas de forma colaborativa e leitura de documentação oficial, aprendemos a buscar soluções em fóruns (Stack Overflow).

O HemoCamp foi fundamental para consolidar conhecimentos que, até então, eram apenas teóricos. Conceitos vistos em aulas de Banco de Dados Relacionais, Programação Web e disciplinas anteriores ganharam sentido prático. Ver um sistema funcionando de ponta a ponta, desde o cadastro de um usuário até a visualização dos dados dos hemocentros, foi extremamente gratificante.

O projeto também nos preparou melhor para o mercado de trabalho. Aprendemos tecnologias amplamente utilizadas (Python, Flask, MySQL), boas práticas de desenvolvimento (versionamento, documentação, testes) e experimentamos desafios reais que profissionais da área enfrentam diariamente.

Para versões futuras do sistema, foram identificadas diversas oportunidades de evolução que podem ampliar o impacto do HemoCamp e melhorar a experiência dos usuários. Uma das principais expansões previstas é o desenvolvimento de um aplicativo mobile nativo em Flutter ou React Native, oferecendo maior praticidade para agendamentos, visualização de campanhas

e acompanhamento de notificações. Também se considera a integração com redes sociais, facilitando o compartilhamento de campanhas e aumentando o engajamento da comunidade.

Do ponto de vista funcional, planeja-se implementar recursos como recuperação de senha, edição de informações pessoais e atualização de dados de campanhas pelos colaboradores. Outra melhoria importante é a inclusão de notificações automáticas sobre novas campanhas, atualizações de agendamentos, avisos sobre baixas críticas no estoque e adição de notícias acerca dos hemocentros e campanhas. Além disso, um botão dedicado para exibir os níveis de estoque em tempo real permitiria maior transparência e incentivaria doações prioritárias dependendo da necessidade do hemocentro.

Para a área administrativa, pretende-se oferecer ferramentas mais robustas, como um painel para criação, edição e busca de notícias, além de um dashboard analítico com métricas de desempenho de campanhas, taxas de comparecimento, movimentação de estoque e estatísticas consolidadas em tempo real. A expansão também inclui o desenvolvimento de uma aba específica de estatísticas para colaboradores dos hemocentros, favorecendo decisões mais informadas.

Em relação à inteligência de dados, há potencial para integrar modelos de machine learning capazes de prever períodos de maior demanda de sangue com base em histórico de doações, padrões sazonais e emergências regionais. Isso permitiria campanhas mais eficientes e monitoramento mais preciso do estoque. Outro ponto a ser aprimorado envolve regras de negócio mais rigorosas, especialmente no cálculo do intervalo mínimo entre doações, garantindo que usuários não agendem coletas simultâneas em hemocentros diferentes.

Por fim, para melhorar a qualidade técnica e facilitar futuras manutenções, está prevista a adoção de testes automatizados com Pytest, containerização com Docker, além da criação de um pipeline CI/CD para entregas mais seguras e contínuas. Essas melhorias estruturais aumentariam a confiabilidade, escalabilidade e longevidade da plataforma.

O desenvolvimento do HemoCamp representou muito mais do que a entrega de um trabalho acadêmico. Foi uma oportunidade de aplicar conhecimentos de forma integrada, enfrentar desafios reais de desenvolvimento de software e contribuir, ainda que modestamente, para uma





causa social importante. A doação de sangue salva vidas diariamente, e poder facilitar esse processo através da tecnologia traz um senso de propósito ao trabalho realizado.

Os desafios enfrentados, longe de serem obstáculos, foram oportunidades de aprendizado. Cada erro, cada bug difícil de resolver e cada funcionalidade que finalmente funcionou contribuíram para o crescimento técnico e profissional da equipe. O projeto demonstrou que desenvolvimento de software é um processo iterativo, que exige paciência, pesquisa constante e trabalho colaborativo.

Concluindo, o HemoCamp cumpre seu objetivo não apenas como sistema tecnológico, mas também como experiência formativa. A equipe sai do projeto com conhecimentos sólidos, portfólio enriquecido e, principalmente, a confiança de que é capaz de desenvolver soluções digitais completas e funcionais que podem, de fato, impactar positivamente a sociedade.