Logotipo, nome da empresa

O conteúdo gerado por IA pode estar incorreto.

**Elemento X**  
Voluntary

Ana Luiza Andrade Oliveira – Matutino – 4B

Luíza da Silva Pereira– Matutino – 4B

Marcelo Henrique – Matutino – 4B

Ryan Pereira Ribeiro– Matutino – 4B

Silas Querino– Matutino – 4B

Thalita Kamille Soares Véras – Matutino – 4B

Ana Luiza Andrade Oliveira – Matutino – 4B

Luíza da Silva Pereira– Matutino – 4B

Marcelo Henrique – Matutino – 4B

Ryan Pereira Ribeiro– Matutino – 4B

Silas Querino– Matutino – 4B

Thalita Kamille Soares Véras – Matutino – 4B

**Elemento X**

Voluntary

Professor: Geovane da Costa Oliveira

Engenharia de Software: 4 semestre

Matéria: Projeto integrador

A Elemento X é uma empresa que criou um portal de voluntariado chamado “Voluntary”. Uma página web que conecta voluntários e empresas

**1 Resumo 6**

[**2 Introdução 6**](#_aidod13ty2lx)

[**3 Justificativa 7**](#_w51vvu7oz1oi)

[3.1 Cronograma 8](#_qw48tn9y9bn8)

[3.2 Técnicas de Elicitação 9](#_8d9cwwq3goe)

[3.3 Stakeholders 10](#_aepm3h485b2q)

[3.4 Requisitos Funcionais 12](#_266h2yvgyqoz)

[3.5 Requisitos Não Funcionais 15](#_rgwmc08s3azk)

[3.6 Requisitos de Domínio 16](#_odrz71p1jwiy)

[3.7 Regras de Negócios 17](#_d8c2vz81wwks)

[3.8 Possíveis Desafios 19](#_nmzbmw29m135)

[3.10 Diagrama de sequência 21](#_29aridz7n6q8)

[3.11 Diagrama de classe 23](#_zc440tsuvg6w)

[3.12 Mapa de navegação 24](#_uabj3emcac07)

[3.13 Canva Model 25](#_gfjdbyqwxoan)

[3.14 BPNM 26](#_vjxkkfshph0y)

[**4 Metodologia 27**](#_h1tc21hyxxs0)

[CSS, (Cascading Style Sheets, em português Folha de Estilo em Cascata) 29](#_615c8i8a6qcn)

[JavaScript (JS): 29](#_1v4lppa28h46)

[Draw.io: 29](#_tgrd0n7xx0a2)

[Google Docs: 30](#_pi603lk8p9fi)

[**5 Desenvolvimento 30**](#_591fqzw7znxu)

[UserController 33](#_gxbjjy5boyar)

[Estrutura e Tecnologias Utilizadas 33](#_ftjaochhm40d)

[Processo de Cadastro de Usuário 34](#_pdm0d5dnmeg6)

[Verificação de E-mail 34](#_tokosmmfyhu)

[Autenticação (Login) 35](#_sy4vciajfu0o)

[Listagem de Usuários 35](#_dg7gsforhiw)

[Considerações Importantes e Boas Práticas 35](#_pqllytvo4epo)

[PasswordController 36](#_6hty2f2qha0y)

[EmpresaController 38](#_yrzd3zqhwmwp)

[Estrutura Geral e Finalidade do Módulo 38](#_9zmjhywb9m4k)

[Importação de Bibliotecas e Configurações 38](#_eudjhzf7ubz)

[Funções de Suporte 39](#_n4z5r06si9t0)

[Autenticação e Recuperação de Senha 39](#_5etgub6mwojb)

[Revisão Literária 47](#_hnzu8bu6mhiy)

[**5. Conclusão 50**](#_hhwuk7pi1emg)

[Descrição dos resultados 50](#_wbuu50wsq7ma)

[**6. Referências 52**](#_2lx24h7vk3fo)

|  |  |  |
| --- | --- | --- |
| Nome | cargo | Descrição |
| Ana Luiza Andrade Oliveira | Analista de requisitos | Levanta e define as necessidades do sistema junto ao cliente. |
| Luíza da Silva Pereira | Gerente | Planeja, organiza e acompanha todas as etapas do projeto. |
| Marcelo Henrique | Documentação | registra informações e garante a padronização dos artefatos. |
| Ryan Pereira Ribeiro | Desenvolvedor | Cria, testa e implementa o software conforme os requisitos definidos. |
| Silas Querino | Desenvolvedor | Cria, testa e implementa o software conforme os requisitos definidos. |
| Thalita Kamille Soares Véras | Desenvolvedora | Cria, testa e implementa o software conforme os requisitos definidos. |

Grupo de pessoas em pé posando para foto

O conteúdo gerado por IA pode estar incorreto.

**Figura 1: Imagem da equipe**

# 1 Resumo

O Voluntary nasce do desejo de tornar acessível o engajamento voluntário, por meio de seu portal de voluntariado, a empresa criará um pedido de ajuda facultativa onde quem quiser se candidatar conseguirá de forma voluntária. Seu diferencial estará na democratização da participação social, oferecendo uma ferramenta amigável que poderá servir tanto para organizações estruturadas quanto ações emergenciais ou de pequeno alcance. Podendo assim, criar oportunidades e experiências para os voluntários.

# 2 Introdução

O projeto Voluntary visará promover a solidariedade e o engajamento voluntário de forma acessível e eficaz. Teremos como objetivo a criação de um portal de voluntariado que conecta empresas, pequenos empreendimentos e iniciativas de solidariedade com voluntários dispostos a ajudar. Ao fortalecer essas conexões, a Voluntary busca não só ampliar o impacto social, mas também fomentar redes colaborativas que gerem benefícios reais às comunidades envolvidas.  
  
Assim, ao fortalecer as conexões entre quem deseja ajudar e quem precisa de apoio, o Voluntary pretende não apenas ampliar o impacto das ações de voluntariado, mas também fomentar uma cultura de solidariedade e colaboração contínua. O projeto se baseia na convicção de que, ao aproximar diferentes atores sociais em um ambiente digital confiável, é possível gerar benefícios reais, duradouros e transformadores para as comunidades envolvidas.  
  
O site será um portal de vagas de voluntariado, no qual empresas, organizações e pequenos empreendimentos poderão publicar oportunidades de voluntariado. Essas vagas deverão acompanhar tags que descrevem a área do serviço voluntário, facilitando a busca por parte dos voluntários. Dessa forma, cada pessoa interessada poderá utilizar filtros e palavras-chave para encontrar rapidamente atividades que estejam alinhadas aos seus interesses, competências ou causas que deseja apoiar.

# 3 Justificativa

A quantia de voluntariado vem aumentando significativamente no Brasil, sendo algo que beneficia várias pessoas, não só o voluntário, além de facilitar as conexões interpessoais. Baseado no artigo voluntariado: um fio que conecta a transformação social à realidade de muitos brasileiros, “Mais de 7,3 milhões de brasileiros se dedicaram a atividades voluntárias” SOUZA, Priscila Rodrigues de. Voluntariado: “um fio que conecta a transformação social à realidade de muitos brasileiros”. *Hora Campinas*, 08 jul. 2025.Podemos notar que o serviço voluntário vem ganhando bastante força no Brasil e por meio da criação da nossa plataforma iremos conseguir democratizar a realização desses serviços.

Outro ponto fundamental é o cuidado com o meio ambiente. Diversos projetos são voltados à preservação ambiental, como campanhas de limpeza, reflorestamento e educação ecológica. Essas iniciativas são essenciais para garantir um futuro mais sustentável e consciente para todos, promovendo um impacto positivo na vida das pessoas ao seu redor. Essa atitude vai além de uma ajuda pontual, ela representa comprometimento com a transformação social e com a melhoria das condições de vida coletivas.

Por fim, o voluntariado também contribui para o crescimento pessoal e profissional. Participar dessas atividades é uma oportunidade de adquirir novas habilidades, desenvolver competências interpessoais e até mesmo ganhar experiências que podem servir como diferencial em processos seletivos de emprego.

## 3.1 Cronograma

É um documento (ou tabela, gráfico etc.) que mostra as atividades do projeto distribuídas ao longo do tempo. Nesse projeto foi utilizado para ajudar a controlar o andamento do projeto, garantindo que todas as tarefas sejam concluídas no prazo e na ordem correta.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Figura 2: Cronograma**

## 3.2 Técnicas de Elicitação

métodos para obter e entender informações, especialmente na área de requisitos, e incluem ferramentas como entrevistas, questionários, observação, análise de documentos, brainstorming e workshops.

Prototipação: Criação de modelos visuais da plataforma para validar ideias e funcionalidades antes do desenvolvimento final.

Brainstorm: Sessões colaborativas com stakeholders para levantar necessidades, expectativas e soluções inovadoras.

Workshops: Reuniões com grupo de interessados para alinhar visões, discutir funcionalidades e chegar em um consenso.

Observação: Acompanhamento do comportamento dos usuários em seus ambientes reais para identificar práticas, dificuldades e necessidades.

Questionários: Coleta de informações em larga escala por meio de formulários, possibilitando identificar padrões de necessidade.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Figura 3: Técnicas de elicitação**

## 3.3 Stakeholders

São indivíduos, grupos ou outras organizações que têm um interesse direto ou indireto numa empresa ou projeto, sendo influenciados pelas suas decisões ou podendo influenciar os seus resultados. Exemplos comuns incluem clientes, funcionários, investidores, fornecedores e a comunidade local, e o seu envolvimento e satisfação são cruciais para o sucesso e sustentabilidade a longo prazo de qualquer negócio.

|  |  |  |  |
| --- | --- | --- | --- |
| Stakeholder | Papel no Projeto | Interesse | Impacto |
| Instituições/ONGs: Organizações que necessitam de voluntários para executar seus projetos sociais | Publicam vagas e oportunidades de voluntariado. | Encontrar voluntários confiáveis para seus projetos sociais. | Alto |
| Voluntários: Pessoas dispostas a doar seu tempo e habilidades em prol de causas sociais. | Se cadastrem e se candidatem às vagas. | Participar de ações solidárias, adquirir experiência e apoiar causas. | Alto |
| Empresas apoiadoras: Negócios que oferecem suporte através de vagas, doações ou parcerias. | Oferecem vagas, doações ou parcerias. | Ganhar visibilidade social, melhorar imagem institucional e engajar colaboradores. | Médio/Alto |
| Organizações Governamentais: Entidades públicas que podem apoiar e validar projetos de voluntariado. | Validam e apoiam iniciativas de voluntariado. | Garantir que o projeto esteja alinhado a políticas públicas e sociais. | Médio |
| Comunidades Necessitadas de Apoio: Grupos sociais que se beneficiam diretamente das ações voluntárias. | Recebem diretamente os benefícios das ações de voluntariado. | Serem apoiadas em demandas sociais, ambientais ou emergenciais. | Alto |

## 3.4 Requisitos Funcionais

É uma lista de comportamentos, funções e operações que um sistema deve ser capaz de realizar, os requisitos funcionais também descrevem como essas funções serão implementadas no sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Código | Função | Descrição | Prioridade |
|  | RF01 | Login e Cadastro (Voluntariado, Empresa e Administração) | O sistema deve permitir que voluntários se cadastrem criando login e senha.  Empresas e instituições terão o cadastro validado pelo administrador do sistema, para garantir a confiabilidade das empresas e prevenir a criação de perfis falsos que possam comprometer o sistema com vagas de voluntariado falsas.  O administrador terá acesso diferenciado, com permissões para gerenciar toda a plataforma. | Alta |
|  | RF02 | Gestão de Perfil | O voluntário poderá criar e editar seu perfil, adicionando suas informações pessoais, áreas de interesses, disponibilidade de tempo, localização, habilidades e experiências. Essas informações serão usadas para facilitar a busca de vagas disponíveis. | Alta |
|  | RF03 | Tela Principal – Voluntário | Uma área centralizada onde os voluntários poderão visualizar as vagas em destaque, oportunidades recentes, e acompanhar novidades. | Alta |
|  | RF04 | Tela de Publicação das Vagas – Empresas | Funcionalidade destinada às empresas parceiras, que possibilita o cadastro e a divulgação de oportunidades de voluntariado na plataforma. Essa tela deve permitir a inserção de informações detalhadas sobre a vaga, como descrição detalhada da atividade, requisitos, local de execução e período de atuação, tags e categoria das vagas, garantindo clareza e transparência para os voluntários interessados. | Alta |
|  | RF05 | Busca e Filtro de Vagas (na tela principal) – Voluntário | Facilita a localização de oportunidades específicas através de filtros como tag, categoria, área de atuação, localização, período ou tipo de atividade. | Média |
|  | RF06 | Candidatura pelo site – Voluntariado | O voluntário poderá se candidatar diretamente a uma vaga, com envio automático de notificação à empresa por painel interno do portal. | Alta |
|  | RF07 | Administração do Portal | O administrador terá acesso total ao monitoramento de vagas, usuários cadastrados e denúncias de conteúdos inadequados. | Alta |
|  | RF08 | Feedback – Voluntário e Empresa: | Após o término da experiência,os voluntários poderão deixar avaliações. Isso ajuda a criar reputação e credibilidade dentro da plataforma. | Média |
|  | RF09 | Gestão de Vagas – Empresa: | As empresas terão autonomia para cadastrar, atualizar, suspender ou remover vagas de voluntariado. | Alta |
|  | RF10 | Monitorar Publicações – Administrador | O administrador poderá supervisionar as vagas publicadas e intervir em casos de irregularidades, garantindo a confiabilidade das informações. | Alta |
|  | RF11 | Notificações e Alertas | O sistema enviará sobre novas vagas, mudanças em candidaturas e lembretes de atividades | Média |
|  | RF12 | Tela inicial | Primeira tela ao entrar no nosso site | Alta |

## 3.5 Requisitos Não Funcionais

São critérios que definem como o sistema deve operar, e não as funções que ele realiza. Eles garantem a qualidade, eficiência e experiência do usuário, servindo como parâmetros para avaliar se o sistema atende aos padrões esperados. Eles descrevem os atributos de qualidade do sistema, como desempenho, usabilidade, segurança, confiabilidade e escalabilidade do sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| Código | Função | Descrição | Tipo |
| RNF01 | Tempo de Resposta | O sistema deve garantir agilidade na navegação e carregamento das páginas, oferecendo uma boa experiência ao usuário. | Desempenho |
| RNF02 | Segurança – Criptografia de Dados Sensíveis | Todas as informações confidenciais, como dados pessoais e senhas, devem ser protegidas por mecanismos de criptografia. | Segurança |
| RNF03 | Usabilidade e Acessibilidade | A interface deve ser intuitiva e inclusiva, permitindo que qualquer usuário consiga utilizar o portal. | Utilidade |
| RNF04 | Escalabilidade – Capacidade de Expansão | A plataforma deve estar preparada para suportar um número crescente de usuários, vagas e empresas, sem comprometer o desempenho. | Desempenho |
| RNF05 | Manutenibilidade | O código deve ser estruturado e documentado de forma a facilitar futuras correções e evoluções do sistema. | Escalabilidade |
| RNF06 | Auditabilidade e Rastreabilidade | O sistema deve manter logs de acesso, cadastro e alterações feitas por voluntários, empresas e administradores. Deve ser possível rastrear ações em casos de fraude ou mau uso. | Segurança |

## 3.6 Requisitos de Domínio

Os requisitos de domínio estão associados a regras e padrões específicos do domínio de aplicação do software. Eles são moldados pelo contexto em que o software será utilizado e podem incluir regulamentações legais, padrões da indústria ou práticas específicas do negócio. Por exemplo, um software de contabilidade pode precisar aderir a padrões contábeis internacionais específicos.

Ajuda Voluntariado: O sistema deve apoiar de forma prática e acessível a conexão entre voluntários e instituições, promovendo impacto social positivo.

## 3.7 Regras de Negócios

Ao traduzir as necessidades de um negócio (como restrições e validações) em regras lógicas, a regra de negócio se torna o guia para toda a empresa. Ela cria um entendimento unificado entre as equipes de desenvolvimento, produto e negócio, permitindo que todos apliquem essas diretrizes com clareza. Isso não apenas otimiza o desenvolvimento, mas também garante um crescimento mais organizado e eficaz do produto.

|  |  |  |  |
| --- | --- | --- | --- |
| Código | FUNÇÃO | Regra | Requisito |
| RN01 | Cadastro de Voluntário | Todo voluntário deve fornecer dados pessoais básicos como nome, e-mail, telefone, CPF, data de nascimento, gênero. | O sistema deve exigir o preenchimento obrigatório desses campos |
| RN02 | Senha | A senha deve ter no mínimo 8 caracteres, contendo pelo menos 1 número, 1 letra maiúscula e 1 caractere especial. | O sistema deve validar a senha antes de concluir o cadastro ou login. |
| RN03 | Cadastro de ONGs/Instituições | Cada ONG deve fornecer CNPJ ou documento oficial para registro. | O sistema deve validar o formato e autenticidade do CNPJ. |
| RN04 | Oportunidades de Voluntariado | Cada vaga publicada deve conter título, descrição, local, presencial ou online, período, número de vagas e requisitos. | O sistema deve obrigar o preenchimento de todos esses campos. |
| RN05 | Inscrição de Voluntários | O voluntário só é considerado participante após ser aceito pela ONG. | O sistema deve registrar o status da inscrição como “pendente” até a aprovação. |
| RN06 | Gerenciamento de Inscrições | A ONG pode aceitar ou rejeitar voluntários inscritos. | O sistema deve permitir atualizar o status da inscrição (aceito/rejeitado). |
| RN07 | Banimento | O administrador pode bloquear usuários, voluntários ou ONGs em caso de fraude ou comportamento inadequado. | O sistema deve oferecer opção de bloqueio e impedir o login de usuários bloqueados. |
| RN08 | Histórico | O voluntário pode consultar seu histórico de atividades realizadas. | O sistema deve manter registro das oportunidades concluídas. |
| RN09 | Feedback | O Voluntário pode realizar uma avaliação sobre a vaga de voluntariado da ONG na qual participaram. | O sistema deve permitir associar avaliações ao perfil das ONG. |

## 3.8 Possíveis Desafios

barreiras e obstáculos que podem impedir o sucesso, geralmente incluem falhas na comunicação, gestão de recursos, cronograma e escopo (o aumento do tamanho do projeto). Outros desafios incluem conflitos na equipa, problemas com a gestão dos stakeholders (partes interessadas), imprevistos e a falta de objetivos claros.

* Alcançar Empresas: Dificuldade de atrair empresas para disponibilizar vagas e participar da plataforma. Exige campanhas de divulgação, parcerias estratégicas e benefícios claros para participação
* Alcançar Voluntários: Garantir que os voluntários criem perfis completos e se candidatem às vagas pode ser um desafio. É necessário oferecer uma plataforma simples, intuitiva e recompensadora com certificados digitais e reconhecimento social.
* Segurança e Confiabilidade: Risco de perfis falsos ou vagas fraudulentas. Exige sistema de verificação de empresas e denúncias ativas por parte da comunidade.
* Acessibilidade e Usabilidade: O sistema deve ser acessível a diferentes tipos de pessoas, incluindo recursos de acessibilidade (leitor de tela, contraste e navegação simplificada).
* Integração com Outros Sistemas: Possível necessidade de integração com redes sociais, e-mail marketing ou APIs de ONGs para facilitar divulgação e captação de voluntários.
* Dificuldade em Desenvolver: Barreiras técnicas e de recursos que podem atrasar ou complicar a implementação do sistema.
* Dificuldade em Documentar: A elaboração de documentação completa e organizada pode ser um processo trabalhoso, mas essencial para a manutenção do projeto.

**3.9 Diagrama de casos de uso**

Um diagrama de casos de uso é um tipo de diagrama da UML (Linguagem de Modelagem Unificada) que mostra como os usuários interagem com um sistema. Ele é uma das formas mais simples e visuais de representar o que o sistema deve fazer, sem precisar mostrar como ele funciona por dentro.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Figura 4: Diagrama de casos de uso**

## 3.10 Diagrama de sequência

Diagrama de sequência mostra como os objetos ou componentes de um sistema interagem ao longo do tempo. Ele representa a ordem das mensagens trocadas entre eles para executar uma função específica. Serve para entender o fluxo de comunicação e a lógica de execução de um processo dentro do sistema. Ajuda os desenvolvedores a visualizar o passo a passo de uma funcionalidade. É muito usado na fase de análise e design de sistemas.

Calendário

O conteúdo gerado por IA pode estar incorreto.

**Figura 5: Diagrama de sequência**

## 3.11 Diagrama de classe

Diagrama de classes mostra a estrutura do sistema, representando suas classes, atributos, métodos e relacionamentos.  
Ele serve para organizar e entender como os dados e funções estão conectados dentro do sistema.  
Cada classe representa um objeto ou entidade (como Usuário, Empresa, Vaga).  
Ajuda os desenvolvedores a planejar o código de forma clara e lógica.  
É essencial na modelagem orientada a objetos, antes da programação começar.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Figura 6: Diagrama de classes**

## 3.12 Mapa de navegação

Um mapa de navegação mostra como as páginas ou telas de um sistema estão conectadas entre si.  
Ele representa o caminho que o usuário pode seguir, como um mapa de links e botões.  
Serve para planejar a usabilidade e a estrutura de navegação do site ou aplicativo.  
Ajuda a equipe a visualizar o fluxo de telas e evitar confusões na interface.  
É muito usado na fase de design e prototipação de projetos digitais.

Gráfico, Diagrama, Gráfico de caixa estreita

O conteúdo gerado por IA pode estar incorreto.

**Figura 7: mapa de navegação**

## 3.13 Canva Model

É uma ferramenta visual usada para planejar e entender modelos de negócio, mostrando em um único quadro como a empresa cria, entrega e gera valor.  
É dividido em nove blocos, como proposta de valor, clientes, canais e receitas.  
Assim, facilitando a comunicação, o planejamento e a inovação em projetos ou empresas.  
Usamos o Canvas para organizar ideias e testar estratégias de forma rápida e simples.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

**Figura 8: Model Canvas**

## 3.14 BPNM

O BPMN (Business Process Model and Notation) é uma linguagem visual usada para modelar e representar processos de negócio.  
Ele mostra o passo a passo das atividades, decisões e fluxos dentro de um processo, facilitando o entendimento entre equipes técnicas e de gestão.  
Ele ajuda a identificar melhorias e otimizar processos.  
Estamos usando para organizar, padronizar e visualizar claramente como as tarefas acontecem.

Diagrama, Esquemático

O conteúdo gerado por IA pode estar incorreto.

**Figura 9: BPMN**

# 4 Metodologia

Análise do Problema e Justificativa**:** É a etapa inicial em que se identifica a necessidade de criar a plataforma, considerando o contexto social e tecnológico. Aqui o grupo reconheceu que o voluntariado está crescendo no Brasil, mas ainda faltava um espaço digital confiável que conectasse voluntários e instituições. O grupo decidiu a criação do portal Elemento X como resposta à dificuldade de encontrar oportunidades de voluntariado e facilitar o acesso.

**Modelagem do Sistema:**

Representação visual do sistema por meio de diagramas (UML, fluxos, mapas de navegação).Foi usado Draw.io para criar os diagramas. Isso facilita a comunicação entre equipe e stakeholders.

**Figma**

O Figma é uma ferramenta de design de interface,baseada na nuvem. Ele é usado para criar protótipos, wireframes, layouts e interfaces de sites, aplicativos e sistemas em geral.

**Visual Studio Code (VS Code)**

O Visual Studio Code, também chamado de VS Code, é um editor de código-fonte desenvolvido pela Microsoft. Ele é gratuito, leve, multiplataforma (funciona no Windows, Linux e macOS) e open source (código aberto). O VS Code é usado para programar e editar arquivos de código em diversas linguagens.

Foi usado no projeto como ambiente de desenvolvimento (IDE), para programar as páginas do portal, integrar o front-end com o back-end e organizar os arquivos do sistema, além de integrar com o GitHub para versionamento.

**GitHub**

O GitHub é uma plataforma online que permite armazenar, gerenciar e compartilhar projetos de software. Ele é baseado no Git, um sistema de controle de versão. O Git ajuda a controlar mudanças em arquivos de código ao longo do tempo. Ele também é um “local na nuvem” onde esse código fica salvo, permitindo colaboração entre várias pessoas.

No projeto foi usado para guardar todo o código do portal, controlar alterações feitas por cada membro do grupo, permitir que cada desenvolvedor trabalhasse em partes diferentes sem perder o histórico, e garantir organização e segurança do projeto.

**Recursos usados**    
HTML, (HyperText Markup Language, em português Língua de Marcação HiperTexto)

É uma linguagem de marcação utilizada para criar e estruturar páginas da web, de forma análoga pode-se comparar ao esqueleto do corpo humano que é a base do corpo, assim funcionando como a base de um site. Ele permite a inserção   
de textos, imagens, vídeos, links e vários outros elementos, além disso, atua através de tags para marcar e organizar os elementos. Será utilizado no site  para a criação e organização das páginas ;

## CSS, (Cascading Style Sheets, em português Folha de Estilo em Cascata)

É um mecanismo utilizado para a estilização de páginas web, de forma análoga, é como se fosse a pele, o cabelo e as roupas de uma pessoa, tudo aquilo que permite enxergarmos uma pessoa, a CSS define como os elementos serão organizados no site para a personalização das páginas;

## JavaScript (JS):

É uma linguagem de programação essencial para o desenvolvimento web, responsável por adicionar interatividade às páginas.   
 No projeto, o JavaScript será utilizado tanto no front-end (interações do usuário, animações, navegação dinâmica) quanto no back-end em conjunto com o Node.js, possibilitando a criação de APIs, autenticação de usuários e integração com o banco de dados.

## Draw.io:

O Draw.io é uma ferramenta gratuita e online utilizada para a criação de diagramas. Ele permite elaborar fluxogramas, mapas mentais, diagramas de casos de uso, diagramas de classe, de sequência, entre outros que são bastante aplicados na área de Engenharia de Software.

## Google Docs:

É uma ferramenta online de edição de textos desenvolvida pelo Google. Diferente dos editores tradicionais instalados no computador, o Google Docs funciona diretamente no navegador, permitindo que os documentos sejam criados, editados e compartilhados em tempo real pela internet.

Sua principal vantagem é a colaboração simultânea, ou seja, várias pessoas podem trabalhar no mesmo arquivo ao mesmo tempo, com as alterações sendo salvas automaticamente na nuvem. Além disso, ele mantém um histórico de versões, facilita comentários e sugestões e garante acesso de qualquer dispositivo conectado. Foi usado para elaborar a documentação do projeto.

# 5 Desenvolvimento

A criação do projeto Voluntary foi pensada a partir da necessidade de criar uma plataforma acessível e eficiente para conectar voluntários a organizações, empresas e iniciativas sociais. Para alcançar esse objetivo, a equipe adotou uma abordagem organizada, utilizando conceitos de Engenharia de Software e práticas de levantamento de requisitos, definição de regras de negócio e análise de stakeholders. Dessa forma, buscou-se alinhar tanto as demandas técnicas quanto sociais do projeto.

Começamos o projeto identificando os requisitos funcionais, que vão representar as ações essenciais que o sistema deve executar. Entre os requisitos mais importantes, destacam-se o cadastro e login de usuários (voluntários, empresas e administradores), a gestão de perfis, a publicação de vagas por parte das organizações e a busca de vagas de voluntariado com filtros, que permite aos voluntários encontrar oportunidades alinhadas a seus interesses e habilidades. Esses requisitos foram pensados pela equipe do elemento x para oferecer uma experiência prática. Além disso, a equipe também pensou em funcionalidades de feedback, notificações e relatórios informativos, garantindo que a plataforma possa atender os stakeholders.

Portanto, também foram definidos os requisitos não funcionais, que explicam melhor como a qualidade e o desempenho do sistema devem se comportar. A segurança é uma parte importante do projeto, onde um dos requisitos não funcionais tem como objetivo o uso de criptografia para proteger dados pessoais e senhas, além da implementação de logs e mecanismos de rastreabilidade. Um dos requisitos também foi a usabilidade e acessibilidade, garantindo que a interface seja intuitiva e inclusiva para diferentes perfis de usuários. Outro aspecto relevante foi a escalabilidade, já que o portal deverá suportar um número crescente de voluntários e vagas sem comprometer a performance.

Nossa equipe trabalhou junto para a criação das regras de negócio, que orientam como algumas partes do sistema devem funcionar. Pensamos em regras como a obrigatoriedade de dados básicos no cadastro do voluntário, a validação do CNPJ das ONGs, a definição de regras para publicação de vagas e o controle sobre inscrições pendentes até a aprovação da instituição. Essas regras visam manter a integridade do sistema e proteger tanto os voluntários quanto as organizações, evitando perfis falsos ou vagas enganosas. Além disso, o administrador possui autonomia para bloquear usuários em caso de irregularidades, com o objetivo de manter a plataforma segura.

Para compreender melhor o ambiente do projeto, foi feita uma análise dos stakeholders. Nela, foram identificadas as ONGs e instituições sociais, os voluntários individuais, as empresas apoiadoras e até órgãos governamentais que podem atuar como parceiros ou validadores. Cada um desses públicos exerce um papel estratégico no sucesso do sistema: os voluntários representam a força de trabalho solidária, as instituições são responsáveis por abrir oportunidades e as empresas podem dar credibilidade e sustentabilidade à plataforma. Essa análise permitiu alinhar as funcionalidades do sistema às necessidades reais de seus usuários finais.

O processo de elicitação de requisitos também foi um ponto crucial para a criação do projeto. Foram utilizadas técnicas como prototipação, para validar visualmente as ideias, brainstorming, que possibilitou o levantamento de ideias e requisitos para o sistema, e workshops, que ajudaram a alinhar as expectativas entre diferentes stakeholders. Além disso, questionários e observações foram fundamentais para compreender as dificuldades enfrentadas por voluntários e organizações. Essa combinação de métodos garantiu que os requisitos definidos fossem bem pensados para que o nosso sistema fosse criado sem problemas.

Apesar de todo o planejamento, a equipe reconhece os possíveis desafios que podem surgir. Entre eles, a dificuldade de atrair empresas para utilizar a plataforma para divulgar oportunidades de voluntariado, a necessidade de manter voluntários engajados e o risco de perfis ou oportunidades fraudulentas. Outro obstáculo é a questão da acessibilidade, que exige recursos técnicos adicionais, e a integração com outros sistemas, como redes sociais, que podem demandar soluções complexas. Também se destacam as dificuldades relacionadas ao próprio desenvolvimento e documentação do sistema, que precisam ser bem estruturadas para evitar atrasos e retrabalhos.

Portanto, o desenvolvimento do Voluntary não se limita apenas a programação, mas também utiliza de teorias e documentação sobre o voluntariado para a criação do sistema. A leitura e análise de artigos e livros demonstraram que o voluntariado tem evoluído no Brasil, passando de práticas assistencialistas para formas mais estruturadas, estratégicas e inclusivas. Isso reforça a relevância do projeto, que pretende contribuir para essa evolução, oferecendo uma ferramenta moderna, acessível e socialmente transformadora.

## UserController

Este código é um módulo de backend feito em JavaScript, usando o ambiente Node.js e algumas ferramentas como Prisma ORM, Bcrypt, JSON Web Token (JWT) e CryptoJS. É importante reiterar que seu propósito principal é oferecer funções básicas para gerenciamento de usuários, como cadastro, confirmação de e-mail, login e listagem de registros. Essas operações são feitas boas práticas atuais de segurança, proteção dos dados e autenticação, comuns em aplicações modernas de grande escala.

## Estrutura e Tecnologias Utilizadas

O módulo utiliza o padrão CommonJS (require e module.exports) para organização e exportação das funções. A comunicação com o banco de dados é realizada por meio do **Prisma ORM**, representado pelo objeto PrismaClient, que abstrai o acesso aos dados e fornece operações estruturadas por meio de métodos como findUnique, findFirst, create, update e findMany.

Para a segurança das credenciais, o código adota três camadas de proteção:

**CryptoJS** realiza a descriptografia da senha recebida do frontend, que originalmente chega criptografada por AES.

**bcrypt** é utilizado na criação de um hash seguro da senha antes de armazená-la no banco de dados.

**JWT** é usado para gerar tokens de validação de e-mail e autenticação de sessão.

Além disso, o módulo utiliza funções auxiliares para envio de e-mail e geração de tokens (sendVerificationEmail, generateToken), reforçando a modularidade da aplicação.

## Processo de Cadastro de Usuário

A função registrarUsuario implementa o fluxo de cadastro. Inicialmente, ela valida se o e-mail e o CPF fornecidos pertencem a usuários potencialmente banidos, impedindo a criação de contas vinculadas a dados bloqueados. Em seguida, valida se já existe um usuário cadastrado com o mesmo e-mail.

A senha recebida é descriptografada com CryptoJS.AES, utilizando uma chave secreta armazenada no ambiente (FLE\_MASTER\_KEY). Caso a descriptografia seja bem-sucedida, a senha é convertida em hash através do bcrypt, garantindo que nenhuma senha seja armazenada em texto puro no banco de dados.

Durante o cadastro, o sistema gera um token JWT de verificação de e-mail com duração de 1 hora. Esse token é armazenado no banco de dados e enviado ao usuário por e-mail. O campo validação é definido como falso, bloqueando o login até que o endereço eletrônico seja confirmado.

## Verificação de E-mail

A função verificarEmail trata do processo de ativação da conta. O token é recebido como parâmetro na URL e validado por meio do JWT. Após a decodificação do token, o sistema localiza o usuário e altera suas propriedades para marcar o e-mail como verificado, removendo o token anteriormente armazenado. Por fim, o sistema redireciona o usuário para uma página informativa na interface frontend.

Esse processo é fundamental para a proteção da plataforma, prevenindo cadastros falsos e garantindo que o e-mail associado seja válido.

## Autenticação (Login)

Na função loginUsuario, o sistema valida a presença obrigatória de e-mail e senha. Após confirmar a existência do usuário, verifica se a conta está banida, bloqueando acessos não permitidos. A senha recebida é novamente descriptografada por AES e comparada com o hash armazenado por meio do método bcrypt.compare.

Além disso, a função impede o login caso o e-mail ainda não tenha sido confirmado, reforçando o fluxo de segurança. Após a autenticação, um token JWT é gerado contendo o ID e o e-mail do usuário, permitindo acesso às áreas protegidas da aplicação. Informações sensíveis, como a senha, são removidas antes do envio ao cliente.

## Listagem de Usuários

A função listarUsuarios oferece um mecanismo simples para exibir uma lista de usuários cadastrados. Embora não contenha lógica complexa, ela demonstra o uso das operações de leitura do Prisma, selecionando apenas campos necessários para evitar exposição de informações sensíveis. Normalmente, tal funcionalidade seria restrita a administradores

## Considerações Importantes e Boas Práticas

O código aplicado demonstra diversos aspectos fundamentais de segurança e boas práticas no desenvolvimento de APIs modernas:

**Criptografia e Hashing de Senhas:** Aplicação de dois níveis de segurança, garantindo proteção tanto em trânsito quanto em armazenamento.

**Verificação de Identidade por E-mail:** Mecanismo importante de segurança e organização, impedindo cadastros anônimos.

**Uso de Tokens JWT:** Estratégia atual para autenticação sem estado (stateless), eficiente e escalável.

**Gerenciamento de Contas Banidas:** Impede reutilização de informações vinculadas a comportamentos inadequados.

**Validação Estruturada:** Verificação de todos os campos necessários e mensagens claras de erro.

## PasswordController

Um conjunto de funcionalidades importantes são implementadas neste código, tendo em vista o processo de recuperação e redefinição de senha de usuários em uma aplicação web. Diante disso, é válido citar que ele é desenvolvido em JavaScript, executado no ambiente Node.js, e utiliza o Prisma ORM como camada de acesso ao banco de dados. Além disso, incorpora o uso de bcrypt para hashing de senhas, bem como funções auxiliares para envio de e-mails e geração de códigos de verificação. A estrutura geral segue o modelo de controladores típicos de APIs REST, nos quais cada função exportada corresponde a um endpoint acessado pelo cliente.

O módulo inicia importando dependências fundamentais: o PrismaClient, responsável por comunicar-se com o banco de dados; o utilitário emailUtils, que contém funções para envio de e-mails e geração de códigos temporários; e a biblioteca bcrypt, utilizada para criptografar senhas de forma segura. A criação de uma instância do Prisma permite manipular os registros da tabela usuário, sendo esta a entidade central do processo de redefinição de senha.

A primeira função, enviarCodigoReset, é responsável por iniciar o fluxo de recuperação. Ela recebe o e-mail fornecido pelo usuário e verifica sua existência no banco de dados. Caso o usuário seja localizado, é gerado um código numérico temporário por meio da função gerarCodigo(), juntamente com um prazo de expiração definido para dez minutos. Esses dados são armazenados no banco para permitir validações posteriores. Em seguida, o código é enviado ao e-mail do usuário utilizando a função enviarEmail(). Assim, essa etapa garante que apenas o proprietário do endereço de e-mail possa prosseguir com a redefinição, estabelecendo uma camada adicional de segurança baseada em autenticação temporária.

A segunda função, verificarCodigoReset, tem como finalidade validar o código enviado ao usuário. O sistema recupera o registro correspondente ao e-mail informado e compara o código fornecido no corpo da requisição com o token armazenado no banco. Caso o código seja incorreto ou esteja expirado, o sistema retorna os erros apropriados. Em caso de sucesso, o token e sua expiração são apagados, permitindo ao usuário avançar para a etapa de criação de nova senha. A função também retorna um redirecionamento para a página adequada no frontend. Essa etapa reforça o controle de integridade do processo de recuperação, garantindo que apenas um token válido permite avançar para a redefinição.

A terceira função, resetarSenha, conclui o processo. Após o usuário fornecer um novo valor de senha, o sistema verifica se o e-mail corresponde a um usuário existente. Em seguida, utiliza a biblioteca bcrypt para gerar um hash seguro, substituindo a senha anterior armazenada no banco. Também zera quaisquer tokens ou campos temporários relacionados à recuperação de senha, prevenindo reutilizações indevidas. Finalmente, retorna ao cliente a confirmação de que a redefinição foi realizada com sucesso. O uso de hashing com bcrypt é um ponto crítico de segurança, uma vez que impede o armazenamento de senhas em texto puro e dificulta ataques de força bruta.

Em relação à estrutura, o módulo apresenta uma organização clara e alinhada às boas práticas de desenvolvimento de APIs em Node.js. As funções são assíncronas, utilizando async/await para lidar com operações de I/O, o que melhora a legibilidade e facilita o tratamento de erros. O uso consistente de validações, retornos HTTP adequados e tratamento de exceções reforça a confiabilidade do sistema. Além disso, o uso de variáveis desestruturadas torna o código mais conciso e explícito quanto aos dados esperados pelo cliente.

## EmpresaController

O código estabelece um módulo backend responsável pelo gerenciamento completo de entidades do tipo “Empresa” dentro de um sistema web. Esse módulo é implementado em JavaScript, executado no ambiente Node.js, utilizando como ORM o Prisma Client, além de diversas bibliotecas complementares para autenticação, criptografia, upload de arquivos, tratamento de imagens e envio de e-mails. Diante disso, é projetado para integrar-se a uma API REST construída sobre o framework Express.

## Estrutura Geral e Finalidade do Módulo

O módulo centraliza operações relacionadas ao ciclo de vida de empresas no sistema, incluindo: criação e validação de contas, login seguro, redefinição de senha, atualização de perfil, gerenciamento de mídias (logo e banner), além da criação e listagem de vagas vinculadas a uma empresa. Sendo assim, constitui a camada responsável pela lógica de negócios associada a empresas cadastradas na plataforma.

A presença do Prisma Client indica que o sistema utiliza um banco de dados relacional, abstraído por meio de modelos definidos previamente no esquema Prisma. Cada função exportada corresponde a um endpoint invocado através de rotas HTTP.

## Importação de Bibliotecas e Configurações

O módulo utiliza diversas bibliotecas:

**@prisma/client** para integração com o banco de dados.

**bcrypt** e **crypto** para hashing e geração de tokens seguros.

**jsonwebtoken (JWT)** para autenticação baseada em tokens.

**CryptoJS** para descriptografar senhas enviadas pelo frontend via AES.

**sharp** para análise e validação de imagens.

**nodemailer** para envio de e-mails transacionais.

**fs/promises** e **path** para manipulação de arquivos no servidor.

Diante disso, verifica-se que essas bibliotecas são combinadas para garantir

segurança, integridade de dados, upload robusto de arquivos e comunicação segura com o usuário.

## Funções de Suporte

O código define funções auxiliares fundamentais para padronização de dados, tais como:

**onlyDigits()**, que remove caracteres não numéricos.

**formatCNPJMask()**, que aplica máscara ao CNPJ.

**calcProgresso()**, responsável por calcular o progresso de preenchimento do perfil da empresa.

**safeUnlink()**, que exclui arquivos temporários de forma segura.

**normalizeList()**, usada para normalizar entradas que podem vir como string ou array, especialmente em requisições multipart/form-data.

É válido citar que essas funções ampliam a consistência dos dados recebidos e armazenados pelo sistema.

**Cadastro e Validação de Empresas**

A função **registrarEmpresa()** é responsável por criar simultaneamente o registro da empresa e de seu representante legal. Ela:

valida campos obrigatórios; verifica se o e-mail ou CNPJ já estão banidos; cria o hash da senha com bcrypt; gera token de validação de conta; realiza a persistência no banco; envia e-mail de confirmação para ativação do cadastro.

O processo segue um fluxo seguro, utilizando criptografia, garantindo unicidade e evitando criação de contas fraudulentas.

A função **verificarEmail()** realiza a validação de contas mediante token enviado ao e-mail do usuário.

## Autenticação e Recuperação de Senha

O módulo inclui funções completas de autenticação:

**loginEmpresa()**, que:

recebe senha criptografada em AES, escriptografa utilizando chave simétrica compartilhada, compara hash com bcrypt, verifica validação de conta, bloqueia empresas banidas, gera JWT autenticado por sete dias.

**solicitarRedefinicao()** e **redefinirSenha()**, que:  
realizam envio de e-mail com token de recuperação e aplicam hashing seguro à nova senha.

Esse modelo combina segurança simétrica (AES), hashing unidirecional (bcrypt) e autenticação por token (JWT).

**Gerenciamento de Perfil de Empresa**

As funções **obterPerfilPorId()**, **getEmpresa()** e **updateEmpresa()** permitem visualizar e atualizar dados públicos ou completos da empresa. A atualização inclui validações como:

normalização do nome de usuário (handle), conversão de listas, sanitização de campos sensíveis.

Além disso, recalculam o progresso de perfil, importante para regras de negócio como a publicação de vagas.

**Upload e Validação de Imagens**

A função **uploadImagem()** realiza upload de banners e logotipos da empresa. Ela:

obtém o arquivo via middleware Multer, valida dimensões de banners (mínimo 1200×800 e proporção ≥ 3:2), salva URL relativa do arquivo, atualiza o registro da empresa.

O uso do sharp permite a leitura confiável das dimensões da imagem, reduzindo riscos de uploads inadequados.

**Criação e Listagem de Vagas**

O módulo fornece funções específicas para administração de vagas:

**listarVagasDaEmpresa()**, que lista todas as vagas da empresa autenticada.

**criarVagaParaEmpresa()**, que: valida ID da empresa; verifica se o perfil está 100% completo (regra de publicação); normaliza campos como tags, turno e imagens; aceita uploads múltiplos de imagens da vaga; previne duplicidade combinando imagens antigas e novas.

Há ainda funções públicas:

**getEmpresaPublic()**, para perfis acessíveis publicamente, **listarVagasPublicasPorEmpresa()**, para exibir vagas visíveis no frontend.

Essas funções utilizam o helper **resolveEmpresaId()**, que aceita tanto o ID quanto o @usuario da empresa para facilitar URLs amigáveis.

**Exportação das Funcionalidades**

Ao final, todas as funções são exportadas no module.exports, permitindo que sejam utilizadas em rotas específicas do servidor Express. Essa modularização facilita a manutenção, testabilidade e expansão futura da API.

**AdminController**

No que diz respeito ao código, é válido citar que ele integra a camada de controle de uma aplicação web desenvolvida em Node.js, utilizando a biblioteca Express para gerenciar requisições HTTP e o Prisma ORM como ferramenta de acesso ao banco de dados. Seu principal objetivo é oferecer funcionalidades administrativas para gerenciar diferentes tipos de recursos, como usuários, empresas, vagas, denúncias e feedbacks. Ou seja, o módulo encapsula operações importantes no contexto de moderação e administração da plataforma, incluindo coleta de dados, banimento, desbanimento e remoção de registros.

Inicialmente, observa-se a importação do PrismaClient, instanciado por meio da variável prisma, o que possibilita a realização de consultas e atualizações no banco de dados de modo assíncrono e estruturado. Além disso, são importadas funções que enviam e-mails para notificar usuários e empresas sobre as ações administrativas realizadas na plataforma. A presença dessas funções de notificação demonstra uma integração entre a lógica de negócio e o fluxo de comunicação com os participantes do sistema.

A função auxiliar parseTipo() que está presente no código, exerce papel importante na padronização das operações, tendo em vista que recebe um tipo de recurso e o converte para um formato único, independentemente de variações de plural ou grafia. É válido destacar que a padronização é indispensável para evitar erros e para permitir que as demais funções lidem com entradas consistentes. Desse modo, a função converte valores como "usuarios" e "usuario" em um mesmo identificador "usuario", facilitando o controle interno.

Já a função getPainelDados() é responsável pela consulta simultânea de informações referentes a usuários, empresas e vagas. Utilizando Promise.all(), a função executa múltiplas requisições ao banco de dados de forma concorrente, o que melhora de maneira considerável a performance do sistema. Cada consulta utiliza o método findMany() do Prisma, retornando apenas os campos necessários definidos em select. Ao final, os dados são reunidos e enviados como resposta HTTP em formato JSON, compondo o conjunto de informações exibidas no painel administrativo. Observa-se também o tratamento de exceções, que garante a robustez da aplicação mediante falhas de conexão ou inconsistências.

As funções banirRecurso() e desbanir() tratam das operações de moderação relacionadas ao bloqueio e desbloqueio de entidades no sistema. Em ambos os casos, a função identifica o tipo de recurso e valida o parâmetro id, garantindo que a requisição seja válida. Na operação de banimento, além de marcar o recurso com isBanned: true e registrar a justificativa do banimento com banReason, o sistema envia e-mails automáticos utilizando templates pré-definidos, informando usuários e empresas sobre a penalidade aplicada. A lógica para vagas não envolve notificação, consistindo apenas na atualização dos atributos no banco. A função de desbanimento opera de forma análoga, revertendo os valores desses atributos e comunicando os envolvidos quando necessário.

A função removerItem() compreende a estrutura inicial para remoção de denúncias e feedbacks. Embora as operações estejam representadas como placeholders, sua existência indica a intenção de expandir o módulo para permitir exclusões definitivas no banco de dados por meio do método delete.

O módulo conclui-se com a exportação das funções, permitindo sua integração em outros arquivos da aplicação, seguindo o padrão de modularização do Node.js. Destaca-se que o código apresenta boas práticas, como utilização de try/catch, respostas padronizadas em JSON, validação de parâmetros e separação clara da lógica de negócio.

Em síntese, o código representa um controlador administrativo robusto que organiza, valida, modifica e comunica ações relacionadas aos principais recursos da plataforma. Sua implementação evidencia o uso de padrões contemporâneos no desenvolvimento web, especialmente no que se refere ao uso de Node.js, Express e Prisma ORM como tecnologias centrais para aplicações escaláveis e bem estruturadas.

**VagaController**

O Sistema de Gerenciamento de Vagas é responsável por controlar todo o ciclo de criação, edição, listagem e moderação das vagas que são disponibilizadas pelas empresas na plataforma. Ele também é desenvolvido em JavaScript com Node.js, utilizando o Prisma como ORM para lidar com o banco de dados e bibliotecas auxiliares para processamento de imagens e normalização de dados. Assim como os módulos anteriores, ele se integra a uma API REST baseada em Express.

A criação de uma vaga geralmente ocorre a partir de uma conta empresarial autenticada. Antes de permitir essa ação, o sistema verifica se a empresa está devidamente validada e se seu perfil está completo, isso ajuda a manter a qualidade das publicações e prevenir fraudes. No momento de cadastrar uma vaga, o sistema recebe informações como título, descrição, requisitos, localidade, carga horária, modalidade (remota ou presencial), benefícios e outras características que tornam a vaga compreensível e atraente para candidatos.

Se a vaga incluir imagens, como fotos da empresa ou materiais ilustrativos, elas são processadas seguindo o mesmo fluxo utilizado para banners e logos: o middleware recebe o arquivo, a biblioteca Sharp valida e ajusta caso necessário, e então o sistema armazena o arquivo e registra sua referência no banco de dados.

O módulo também é responsável pela listagem das vagas. Há dois tipos principais de exibição: a lista interna, onde a empresa visualiza suas próprias vagas para administrar, e a lista pública, onde candidatos ou visitantes podem visualizar oportunidades disponíveis. Para facilitar buscas e organização, alguns campos são padronizados e tratados para manter consistência, como tags e tipos de turno.

Além da criação e listagem, existe o processo de moderação. Administradores do sistema podem bloquear ou reativar vagas quando necessário, por exemplo, em casos de violação das regras da plataforma. Quando uma vaga é banida, seu status é atualizado no banco e ela deixa de ser exibida ao público. Caso a situação seja revertida, o status pode ser restaurado.

Todas as funcionalidades relacionadas às vagas são separadas em controladores que podem ser chamados tanto pelo sistema da empresa quanto pelo painel administrativo, garantindo flexibilidade e organização. Assim como nos demais módulos, as funções são exportadas de forma modular para integração com o sistema de rotas.

**Prototipação usando figma:**

Foram utilizadas paletas de cores em tons de laranja e azul em todo o site, promovendo unidade visual e padronização da identidade da plataforma.  
Na página inicial, foram posicionados dois botões principais no lado esquerdo da tela: um destinado ao acesso do voluntário e outro direcionado às empresas, acompanhados da logomarca oficial.  
No lado direito, apresenta-se um breve resumo institucional, oferecendo ao usuário uma visão geral sobre a proposta e os objetivos da empresa.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Figura 10: Tela inicial**

A tela de buscar vagas exibe uma listagem de vagas de voluntariado no site Voluntary, organizada em cartões com imagem, título, breve descrição, etiquetas e botão para ver detalhes. No topo, há o nome da plataforma, ícone de notificações e avatar do usuário. Duas barras de pesquisa permitem localizar vagas por palavra-chave. À esquerda, um painel de filtros permite refinar resultados por tags, turno (manhã, tarde, noturno) e modalidade (presencial, híbrido, online). Tags selecionadas aparecem destacadas em cores e podem ser removidas. As vagas são exibidas em múltiplas colunas, facilitando comparação visual. Cada cartão apresenta etiquetas coloridas indicando categoria, horário e modalidade. A navegação entre páginas é feita pela paginação inferior, com destaque para a página atual. O layout prioriza organização e clareza.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Figura 11: Buscar vagas**

A tela perfil do usuário na plataforma Voluntary, exibindo informações pessoais, competências e histórico de participação em projetos.  
No topo, há um banner ilustrativo, seguido da foto do usuário, nome, @usuário e uma breve descrição. Ao lado, surgem os botões “Denunciar” e “Editar Perfil”.  
Na lateral esquerda, um menu oferece navegação para: procurar vagas, notificações, configurações e sair.  
A seção Histórico mostra cartões de vagas já realizadas ou em andamento, cada um com imagem, título e status, como “Participando” ou “Finalizado”, além da opção “Avaliar projeto”.  
À direita, a área Competências exibe tags de habilidades (como Culinária, Educação, Limpeza e Programação), seguida de informações de disponibilidade (ex.: turno da manhã) e contatos, incluindo e-mail.  
A página é organizada com foco na apresentação do perfil e no acompanhamento das atividades voluntárias.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

O conteúdo gerado por IA pode estar incorreto.

**Figura 12 : Tela de perfil**

## Revisão Literária

O Voluntary nasce do desejo de tornar acessível o engajamento voluntário, por meio de seu portal de voluntariado, a empresa criará um pedido de ajuda facultativa onde quem quiser se candidatar conseguirá de forma voluntária. Para tornar esse projeto realidade, fizemos pesquisas em alguns sites e artigos online para poder entender melhor sobre o voluntariado e o seu crescimento no Brasil.

O texto de Helenara Silveira Fagundes (2006) apresenta o voluntariado como parte de um debate mais amplo sobre solidariedade e políticas sociais. A autora evidencia a tensão entre a responsabilidade do Estado e da sociedade civil na luta contra desigualdades, ressaltando que o voluntariado deve ser entendido não apenas como um gesto individual, mas também como um componente que pode fortalecer a cidadania e a justiça social.

Em um contexto mais recente, a reportagem publicada pelo Correio Braziliense (2024) apresenta dados que comprovam a ampliação do voluntariado no país. Segundo a pesquisa Datafolha, “56% da população adulta diz fazer ou já ter feito algum tipo de atividade voluntária” (CORREIO BRAZILIENSE, 2024), o que representa um crescimento expressivo em relação a anos anteriores. Além disso, a mesma fonte aponta que a pandemia de Covid-19 intensificou esse engajamento, já que “Nos picos mais altos da pandemia, mesmo com o isolamento social, 47% dos entrevistados relataram praticar mais o voluntariado” (CORREIO BRAZILIENSE, 2024). Esses dados reforçam a ideia de que o voluntariado, além de ser um gesto solidário, também é uma resposta social diante de crises.

Por sua vez, Souza (2025), em artigo publicado no portal *Hora Campinas*, define o voluntariado como “um fio que conecta a transformação social à realidade de muitos brasileiros”, destacando sua capacidade de gerar pertencimento e propósito tanto para quem doa quanto para quem recebe. A autora sublinha que, em 2022, “7,3 milhões de brasileiros se dedicaram a atividades voluntárias em 2022. Embora o número seja significativo, ele representa apenas 4,2% da população com 14 anos ou mais” (SOUZA, 2025), evidenciando que ainda existe um amplo espaço para crescimento. O texto também destaca o avanço do voluntariado corporativo, apontando que “89% dos executivos acreditam que o voluntariado melhora o clima interno das empresas e reforça o senso de pertencimento dos colaboradores.” (SOUZA, 2025).

O livro analisa o chamado “novo voluntariado”, que começou a ganhar destaque no Brasil a partir dos anos 1990. A autora mostra que ele não surgiu apenas da boa vontade individual, mas foi construído por meio da ação conjunta do Estado, das empresas, da Igreja e do serviço social, além de campanhas e discursos que ajudaram a dar legitimidade e modernidade a essa prática. Com a metáfora dos “andaimes”, Cunha explica que o voluntariado precisa de estruturas de apoio para se manter, como planejamento, organização e reconhecimento social. A obra também faz uma crítica ao voluntariado assistencialista, incentivando o leitor a refletir sobre quem ajuda, por que ajuda e quais interesses estão por trás dessa prática.

Em conjunto, os artigos e livros utilizados de referência permitem compreender que o voluntariado no Brasil evoluiu de uma prática centrada na solidariedade e no apoio às políticas sociais (2006) para um fenômeno mais amplo e diversificado (2024–2025), capaz de responder a emergências, promover conexões humanas e até integrar estratégias institucionais. Após a leitura e compreensão desses textos, fica mais fácil entender o motivo por trás da criação do voluntário, um site que tem como proposta facilitar e democratizar o serviço voluntário.

# 5. Conclusão

## Descrição dos resultados

O desenvolvimento do projeto Voluntary possibilitou compreender, de forma prática e aplicada, como a Engenharia de Software pode ser usada para solucionar demandas sociais reais. Através da análise de problemas, levantamento de requisitos e aplicação de técnicas de elicitação, foi possível estruturar uma plataforma que conecta voluntários, ONGs, empresas parceiras e comunidades em situação de vulnerabilidade de maneira organizada, segura e acessível. Durante o processo, conseguimos definir com clareza os requisitos funcionais e não funcionais, estabelecendo funcionalidades essenciais como cadastro, login, publicação de vagas, filtros de busca, feedback e controle administrativo, além de aspectos de segurança, desempenho e usabilidade. Também estruturamos regras de negócio que reforçam a confiabilidade da plataforma, prevenindo fraudes e fortalecendo a credibilidade entre os usuários. O uso de ferramentas como Figma, Draw.io, Visual Studio Code, GitHub e Node.js foi essencial para aplicar conceitos de prototipação, versionamento de código, organização e integração entre front-end e back-end. Isso nos ajudou a desenvolver um produto funcional com identidade visual moderna, boa responsividade e foco na experiência do usuário. Outro ponto importante foi a análise de stakeholders e o estudo do voluntariado no Brasil, que nos permitiu entender melhor o impacto social do projeto. Aprendemos que criar uma solução tecnológica não se resume apenas a programar, mas envolve planejamento, empatia, pesquisa, comunicação e colaboração em equipe. O Voluntary representa uma oportunidade de usar a tecnologia para ajudar na transformação social, conectando quem deseja ajudar com quem precisa de apoio.  
 Para gerenciar o desenvolvimento do Voluntary, adotamos o modelo de gerenciamento em cascata, uma metodologia tradicional que segue etapas lineares e correlativas. Essa abordagem foi escolhida pela sua organização, clareza e controle sobre cada fase do projeto, permitindo que o grupo acompanhasse o progresso de forma estruturada e sem sobreposição de etapas. O modelo foi aplicado de forma sequencial, garantindo qualidade, coerência e alinhamento entre todos os membros da equipe. As etapas compreenderam: • Análise do Problema e Justificativa: Identificação da necessidade de uma plataforma que aproximasse voluntários e instituições, considerando o cenário atual do voluntariado no Brasil.

Testes, Validação e Documentação: Verificação das funcionalidades, correção de erros, ajustes de interface e elaboração do relatório técnico final. Durante o processo, o desenvolvimento do Voluntary possibilitou compreender, de forma prática e aplicada, como a Engenharia de Software pode ser usada para solucionar demandas sociais reais. Foi possível estruturar uma plataforma que conecta voluntários, ONGs, empresas parceiras e comunidades em situação de vulnerabilidade de maneira organizada, segura e acessível. Definimos com clareza os requisitos funcionais e não funcionais, estabelecendo funcionalidades como cadastro, login, publicação de vagas, filtros de busca, feedback e controle administrativo, além de aspectos de segurança, desempenho e usabilidade. As regras de negócio reforçam a confiabilidade do sistema, prevenindo fraudes e fortalecendo a credibilidade entre os usuários.

Ao longo do projeto, a equipe aprimorou habilidades em documentação, definição de requisitos e criação de banco de dados. Desenvolveu competências em front-end, back-end e construção de diagramas, apesar de retrabalhos necessários. Houve evolução na elaboração de mapas de navegação, prototipagem visual e pesquisa acadêmica. A metodologia do projeto foi fortalecida, bem como a compreensão das regras de negócio e da estrutura do sistema. Também houve progresso na organização do repositório, criação de cronogramas e elaboração de apresentações. No geral, o grupo aperfeiçoou o trabalho em equipe, a comunicação, o cumprimento de prazos e a capacidade de revisão contínua, consolidando conhecimentos em modelagem, planejamento e desenvolvimento.

# 6. Referências

REDAÇÃO. Artigo – Voluntariado: um fio que conecta a transformação social à realidade de muitos brasileiros – por Priscila Rodrigues de Souza. Hora Campinas, 2025. Disponível em: <https://horacampinas.com.br/artigo-voluntariado-um-fio-que-conecta-a-transformacao-social-a-realidade-de-muitos-brasileiros-por-priscila-rodrigues-de-souza/> Acesso em: 12 Aug. 2025.

Fagundes, Helenara Silveira. "O voluntariado, a solidariedade e as políticas sociais." Textos & Contextos (Porto Alegre) 5.2 (2006): 1-19. Disponível em: <https://www.redalyc.org/pdf/3215/321527159003.pdf> Acesso em 22 Aug. 2025

GIUSTI, Maria Beatriz. Ajudar ao próximo segue em alta entre os brasileiros. Brasil. Disponível em: <https://www.correiobraziliense.com.br/brasil/2024/09/6937679-ajudar-ao-proximo-segue-em-alta-entre-os-brasileiros.html> Acesso em: 28 Aug. 2025.

Cunha, Márcia Pereira. Os andaimes do novo voluntariado. Brasil, Cortez Editora, 2014.

ALENCAR, Ricardo. O que são regras de negócio? Alura. São Paulo, 2024. Disponível em:<https://www.alura.com.br/artigos/o-que-sao-regras-de-negocio>. Acesso em: 4 set. 2025.

FIGMA. San Francisco: Figma, [2025]. Disponível em:<https://www.figma.com/>. Acesso em: 4 set. 2025.

VISUAL STUDIO CODE. Docs. Microsoft, [2025]. Disponível em:<https://code.visualstudio.com/docs>. Acesso em: 4 set. 2025.

GITHUB. About GitHub. GitHub, [2025]. Disponível em:<https://docs.github.com/pt/get-started/using-github/about-github>. Acesso em: 4 set. 2025.

MOZILLA. HTML. MDN Web Docs, [2025]. Disponível em:<https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 4 set. 2025.

MOZILLA. CSS. MDN Web Docs, [2025]. Disponível em:<https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 4 set. 2025.

GOOGLE. Google Docs. Google Workspace, [2025]. Disponível em:<https://workspace.google.com/intl/pt-BR/products/docs/>. Acesso em: 4 set. 2025.

EXAME. Negócios. Exame. Disponível em: [https://exame.com/negocios/](https://exame.com/negocios/?utm_source=chatgpt.com). Acesso em: 16 out. 2025.  
  
GESTÃO DE SEGURANÇA PRIVADA. O que é negócio: definições, objetivos e tipos de negócios. Gestão de Segurança Privada. Disponível em: <https://gestaodesegurancaprivada.com.br/o-que-e-negocio-definicoes-objetivos-e-tipos-de-negocios/>. Acesso em: 16 out. 2025.