

UNIVERSIDADE PAULISTA

Ana Karoliny Alves Marques

Ellen Oliveira Santos

Felipe Paulussi da Silva

Heitor Exposito de Sousa

Kayane Vitoria Rocha da Silva

Matheus Prazeres Santos

Natan de Sousa Oliveira

Thales Vienskis Zandomenegui

**SISTEMA WEB PARA VAGAS DE TRABALHO VOLUNTÁRIO EM ONGS**

SÃO PAULO

2025

UNIVERSIDADE PAULISTA

Ana Karoliny Alves Marques  
Ellen Oliveira Santos  
Felipe Paulussi da Silva  
Heitor Exposito de Sousa  
Kayane Vitoria Rocha da Silva  
Matheus Prazeres Santos  
Natan de Sousa Oliveira  
Thales Vienskis Zandomenegui

**SISTEMA WEB PARA VAGAS DE TRABALHO VOLUNTÁRIO EM ONGS**

Trabalho apresentado como requisito parcial  
para obtenção do título em bacharel em  
Ciência da Computação na Universidade  
Paulista.

Orientador: Prof. Me. Lauro Tomiatti

SÃO PAULO

2025

CIP - Catalogação na Publicação

SISTEMA WEB PARA VAGAS DE TRABALHO VOLUNTÁRIO  
EM ONGS / Ana Karoliny Alves Marques...[et al.]. - 2025.  
0100 f.

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto  
de Ciência Exatas e Tecnologia da Universidade Paulista, São Paulo,  
2025.

Área de Concentração: Tecnologia.

Orientador: Prof. Me. Lauro Rodrigues Tomiatti.

1. Sistema Web. 2. ONGs. 3. Voluntariado. 4. Gestão de Vagas. I.  
Marques, Ana Karoliny Alves. II. Tomiatti, Lauro Rodrigues (orientador).

## AGRADECIMENTOS

Agradecemos, primeiramente, aos nossos pais, que nos deram energia, sabedoria e benefícios para a conclusão de todo esse trabalho.

Agradecemos, aos nossos amigos e colegas, que nos incentivaram no processo para a elaboração deste trabalho.

Agradecemos a todos os professores e profissionais que compartilharam seus conhecimentos e experiências, enriquecendo assim o nosso aprendizado.

## RESUMO

As organizações não governamentais (ONGs) têm um papel crucial na promoção de causas sociais e na assistência à comunidade em situação de vulnerabilidade. Contudo, muitos desses grupos enfrentam dificuldades em divulgar e gerenciar vagas de voluntariado, essenciais para suas operações. Este projeto propõe o desenvolvimento de um sistema web para cadastro de vagas de trabalho voluntário em ONGs, com o objetivo de conectar essas organizações e pessoas interessadas, facilitando o acesso às oportunidades e otimizando os processos de recrutamento e gestão de voluntários.

Como embasamento para o tema, foram utilizados dados que evidenciam a ausência de ferramentas especializadas para ONGs, bem como o crescente interesse da população em iniciativas de voluntariado, evidenciadas através de formulário com a coleta de informações pertinentes. O projeto foca em objetivos como a criação de uma interface intuitiva, cadastro detalhado de vagas e a funcionalidade de busca personalizada.

Conclui-se que o sistema web para cadastro de vagas de trabalho voluntário em ONGs é uma solução prática e acessível, que atende a uma necessidade real e promove impacto social. A validação das informações apresentadas reforça a relevância da proposta, bem como a viabilidade de sua aplicação no ambiente tecnológico.

**Palavras-chave:** Sistema web, ONGs, Voluntariado, Gestão de vagas.

## ABSTRACT

Non-governmental organizations (NGO) play a crucial role in promoting social causes and assisting vulnerable communities. However, many of these groups encounter difficulties in publicizing and managing volunteer vacancies positions, which are essential for their operations. This project proposes the development of a web-based system for registering volunteer positions in NGOs, aiming to connect these organizations and interested individuals, facilitating access to opportunities and optimizing the recruitment and management processes of volunteers.

As a basis for the theme, we used data that demonstrates the lack of specialized tools for NGOs, as well as the growing interest of the population in volunteer initiatives, evidenced by a form that collects relevant information. The project focuses on goals such as creating an intuitive interface, detailed registration of vacancies and personalized search functionality.

We conclude that the web-based system for registering volunteer vacancies in NGOs is a practical and accessible solution that meets a real need and promotes social impact. The validation of the information presented reinforces the relevance of the proposal, as well as the viability of its application in a technological environment.

**Keywords:** Web system, non-governmental organizations (NGO), volunteer work, vacancy management.

## LISTA DE ILUSTRAÇÕES

Figura 01 - Kanban no site Trello .....	39
Quadro 01 - Requisitos Funcionais .....	42
Figura 02 - Diagrama de Classes .....	51
Figura 03 - Casos de Uso .....	53
Figura 04 - Diagrama de Atividades .....	54
Figura 05 - Diagrama Entidade Relacionamento .....	56
Figura 06 - Realização de Trabalho Voluntário .....	58
Figura 07 - Desejo de Realização de Trabalho Voluntário .....	58
Figura 08 - Empecilhos Encontrados .....	59
Figura 09 - Tipos de Trabalho Voluntário .....	59
Figura 10 - Logomarca WeONG .....	60
Figura 11 - Tela inicial .....	61
Figura 12 - Perfil do Candidato .....	61
Figura 13 - Perfil da ONG .....	62
Figura 14 - Mapa de ONGs .....	62
Figura 15 - Quantidade de Vagas .....	63
Figura 16 - Quantidade de Candidaturas .....	64
Figura 17 - Vagas preenchidas versus Vagas Abertas .....	65
Figura 18 - ONGs versus Voluntários .....	66
Figura 19 - Vagas por Área .....	67
Figura 20 - Resultado da Execução do Teste de Criação de ONG .....	69
Figura 21 - Resultado da execução dos testes de permissão de usuários .....	70
Figura 22 - Consolidado Financeiro .....	74

## Sumário

1	INTRODUÇÃO .....	9
1.1	Objetivos.....	10
1.2	Objetivos Específicos .....	10
1.3	Motivação .....	10
2	Referencial Teórico .....	12
2.1	Voluntariado.....	12
2.2	LGPD.....	13
2.3	Privacidade de dados .....	15
2.4	Conformidade .....	15
2.5	Importância do Consentimento e da Transparência.....	16
2.6	Aplicações Web .....	16
2.7	Acessibilidade Web.....	18
3	Materiais e Ferramentas.....	19
3.1	Bootstrap.....	21
3.2	Python.....	21
3.3	Django .....	23
3.4	HTML.....	25
3.5	CSS .....	25
3.6	Javascript .....	25
3.7	MySQL .....	26
3.8	Hospedagem .....	26
3.9	Microserviços.....	28
3.10	Brasil API .....	28
3.11	Biblioteca Validate-docbr .....	29
3.12	OpenCage Geocoder.....	29
3.13	Folium.....	30
3.14	Chart.js.....	31
3.15	Banco de Dados na AWS e Hospedagem .....	32
3.15.1	Banco de Dados na AWS .....	32
3.15.2	Hospedagem do Sistema.....	32
3.16	Git.....	33
3.17	GitHub .....	34
3.18	IPWHOIS.IO .....	35



4	METODOLOGIA .....	36
4.1	Técnicas de Pesquisa .....	36
4.2	Metodologia de Desenvolvimento .....	37
4.2.1	Modelo Incremental.....	37
5	Modelagem .....	41
5.1	Requisitos Funcionais.....	42
5.2	Requisitos Não Funcionais .....	45
5.3	Qualidade de Software.....	47
5.3.1	ISO/IEC 25010 .....	47
5.3.2	ISO/IEC 27001 .....	48
5.3.3	ISO/IEC 9001 .....	49
5.4	Diagrama de Classes.....	51
5.5	Diagrama de Casos de Uso .....	51
5.6	Diagrama de Atividades.....	53
5.7	Diagrama Entidade Relacionamento .....	54
6	Desenvolvimento.....	57
6.1	Formulário de pesquisa.....	57
6.2	Desenvolvimento da Interface do usuário - Front-end .....	59
6.3	Desenvolvimento da Lógica de Negócio - Back-end .....	66
6.4	Testes e Resultados .....	68
6.4.1	Testes no Back-end.....	69
6.4.2	Testes no Front-end.....	70
6.5	Análise Financeira .....	71
7	Trabalhos Futuros .....	74
8	Conclusão .....	76
	Referências.....	78
	Apêndice A – Documentação da Versão 1 .....	83
	Apêndice B – Formulário de Pesquisa.....	86
	Apêndice C – Relatório de Testes da Aplicação .....	88

## 1 INTRODUÇÃO

Atualmente, o trabalho voluntário desempenha papel vital no fortalecimento da malha social da sociedade brasileira, sendo esta malha constituída pelas relações interconectadas entre os indivíduos de uma sociedade em seus distintos níveis de profundidade. Promovendo o desenvolvimento sustentável e a inclusão de comunidades vulneráveis, as organizações não governamentais (ONGs) são no Brasil responsáveis por diversas iniciativas em áreas como saúde, educação e meio ambiente, entretanto, paralelamente enfrentam desafios significativos relacionados à gestão e divulgação de oportunidades em suas iniciativas de voluntariado, bem como a identificação e retenção de perfis compatíveis. Estudos apontam que o uso dos recursos de tecnologia da informação pode transformar as práticas gerenciais e de mensuração de desempenho interno e externo das ONGs, oferecendo além de maior eficiência operacional, impacto em suas iniciativas (OLIVEIRA; CALLADO, 2015). Nesse contexto, a tecnologia surge como um meio transformador e facilitador para as ONGs, capaz de conectar indivíduos e organizações de forma ágil e eficaz, criando redes profundas de colaboração e impacto social, alinhadas com as necessidades contemporâneas.

Diante desse cenário, este trabalho apresenta o desenvolvimento de um sistema web voltado à gestão de vagas de trabalho voluntário em ONGs. A proposta foca na criação de uma plataforma acessível e segura, que não apenas centraliza oportunidades de voluntariado, mas também garante a integridade e a privacidade dos dados em conformidade com legislações como a LGPD (BRASIL, 2018). Além disso, a metodologia empregada combina técnicas ágeis, como o modelo incremental, e ferramentas modernas, permitindo a entrega de um produto robusto e altamente aderente às demandas do setor. Este estudo, portanto, não visa se limitar a resolver uma necessidade técnica do terceiro setor, mas sim evidenciar o impacto positivo da aplicação de conceitos de ciência da computação em questões sociais, unindo inovação tecnológica e responsabilidade social.

Estudos e *benchmarks* sobre plataformas similares, como o VolunteerMatch e o Idealist, demonstram o impacto positivo que sistemas web bem estruturados podem ter no enlace entre organizações e voluntários. Estes sistemas combinam interfaces intuitivas e recursos avançados de busca, permitindo maior engajamento social e eficiência nos processos de gestão. Inspirado por essas iniciativas, o presente trabalho busca adaptar essas boas práticas ao contexto das ONGs brasileiras, criando uma solução tecnológica alinhada às suas necessidades e desafios específicos.

## 1.1 OBJETIVOS

Este Trabalho de Conclusão de Curso (TCC) propõe o desenvolvimento de um sistema web para facilitar a conexão entre organizações não governamentais (ONGs) e voluntários, promovendo a abertura e o gerenciamento de vagas para serviços comunitários. O sistema visa oferecer uma plataforma segura e de interface intuitiva, permitindo o cadastro de ONGs e voluntários, além da administração eficiente das tarefas comunitárias disponibilizadas pelas ONGs.

## 1.2 OBJETIVOS ESPECÍFICOS

- Interface para cadastro de voluntários e suas competências técnicas;
- Interface para cadastro de ONGs e seus projetos e características;
- Validação do cadastro de ONGs que solicitam trabalho voluntário;
- Exibição de vagas disponíveis no sistema;
- Exibição de características das vagas disponíveis no sistema;
- Opção de cadastro de novas vagas para as organizações;
- Apresentação de relatório com a quantidade de inscritos e seus respectivos perfis para cada vaga;
- Possibilidade de identificar os projetos mais próximos do voluntário.
- Levantar os principais requisitos funcionais e não funcionais para o sistema web;
- Projetar e implementar uma interface responsiva e acessível para usuários e instituições;
- Integrar um banco de dados relacional hospedado em ambiente de computação em nuvem (AWS);
- Garantir mecanismos de autenticação, controle de acesso e proteção de dados;
- Aplicar metodologias ágeis, como o modelo incremental, durante o desenvolvimento;
- Realizar testes funcionais e de usabilidade com foco em melhoria contínua;
- Publicar e disponibilizar a plataforma em ambiente online, garantindo sua operação segura.

## 1.3 MOTIVAÇÃO

A atuação de organizações não governamentais (ONGs) desempenha um papel fundamental na promoção de mudanças sociais, contribuindo para o desenvolvimento

sustentável, a redução de desigualdades e a assistência a populações vulneráveis. Contudo, muitas dessas organizações enfrentam desafios para atrair e gerenciar voluntários, recurso essencial para a realização de suas atividades.

No cenário atual, grande parte das ONGs depende de métodos tradicionais ou pouco eficientes para divulgar vagas de voluntariado, o que dificulta o alcance de potenciais interessados. Simultaneamente, muitas pessoas demonstram interesse em participar de iniciativas voluntárias, mas encontram barreiras para acessar informações centralizadas e confiáveis sobre oportunidades disponíveis.

Dessa forma, a proposta de um sistema web para cadastro de vagas de voluntariado em ONGs surge como uma solução tecnológica viável e acessível, que visa facilitar a conexão entre ONGs e voluntários. A ferramenta busca otimizar processos, ampliar a visibilidade das vagas e incentivar a participação cidadã, promovendo um impacto social positivo.

A escolha desse tema é motivada pela relevância da tecnologia como um meio transformador e pela necessidade de criar soluções práticas para problemas enfrentados no setor do terceiro setor. Além disso, o tema também atende à crescente demanda por plataformas que promovam ações colaborativas e solidárias, alinhando-se aos princípios de responsabilidade social e inovação tecnológica.

## 2 REFERENCIAL TEÓRICO

### 2.1 VOLUNTARIADO

De acordo com Alexandre (2022), a história do voluntariado remonta à Antiguidade, quando civilizações como Egito, Grécia e Roma desenvolveram práticas comunitárias que, mesmo não sendo denominadas “voluntariado”, tinham caráter de ajuda coletiva. No Egito, por exemplo, cidadãos contribuem para obras públicas e atividades religiosas. Na Grécia, a filantropia era uma forma de ganhar prestígio, com cidadãos ricos financiando festivais e projetos educacionais. Já em Roma, a elite praticava o patronato, financiando construções públicas e auxiliando os mais necessitados em tempos de crise.

Durante a Idade Média, o voluntariado ganhou uma forte conexão com a Igreja Católica. Era comum que monges e freiras cuidassem de doentes, órfãos e pobres, criando hospitais e albergues que desempenharam papel crucial na assistência social da época. Ordens religiosas, como os Cavaleiros Templários e Hospitalários, também ajudavam peregrinos e vítimas de guerras. A caridade era vista como uma obrigação cristã e uma forma de alcançar a salvação. Nesse período, eventos como a peste negra mobilizaram esforços de ajuda coletiva, especialmente liderados por grupos religiosos.

Conforme Hudson (1999), no Renascimento e na Idade Moderna, o voluntariado começou a se desvincular exclusivamente da religião. A Reforma Protestante trouxe uma visão mais prática da caridade, enquanto comerciantes ricos passaram a investir em escolas e hospitais. Já no século XIX, a Revolução Industrial gerou condições de vida precárias nas cidades, impulsionando a criação de organizações de caridade mais estruturadas. A fundação da Cruz Vermelha em 1863, por Henry Dunant, foi um marco global, introduzindo o conceito de assistência humanitária neutra. Movimentos abolicionistas e organizações de trabalhadores também mobilizaram voluntários para combater a escravidão e melhorar as condições laborais.

De acordo com Pilloti e Rizzinni (1995), o século XX consolidou o voluntariado como resposta a crises globais, especialmente nas duas guerras mundiais. Após a Segunda Guerra Mundial, organizações como a UNICEF e Médicos Sem Fronteiras surgiram para prestar auxílio em áreas devastadas por conflitos. Além disso, o período foi marcado por movimentos sociais, como o pelos direitos civis nos EUA, nos quais voluntários desempenharam papel crucial. O voluntariado internacional também ganhou força com o Peace Corps, fundado em 1961, que enviava jovens para ajudar em projetos educacionais e de saúde em países em desenvolvimento.

Conforme Hudson (1999), com o avanço da globalização e da internet no século XXI, o voluntariado se transformou. Surgiram plataformas digitais que permitem a colaboração remota em projetos humanitários e ambientais. Empresas passaram a incentivar seus funcionários a participarem de ações voluntárias como parte de programas de responsabilidade social corporativa. A sustentabilidade e o meio ambiente também se tornaram causas centrais, com voluntários envolvidos em reflorestamento, conservação ambiental e campanhas contra mudanças climáticas.

Conforme Peliano (2001), no Brasil, o voluntariado teve início com iniciativas ligadas à Igreja Católica, como a Sociedade de São Vicente de Paulo, fundada no início do século XX. Nas décadas de 1990 e 2000, o país viu o crescimento de ONGs focadas em saúde, educação, direitos humanos e meio ambiente. Programas governamentais, como o Programa Nacional de Voluntariado, também incentivaram a participação cidadã.

Hoje, o voluntariado é uma força essencial para enfrentar desafios globais e locais. Seja em ações comunitárias, campanhas de saúde ou resposta a desastres naturais, ele continua a se adaptar às necessidades da sociedade e a explorar novas tecnologias para maximizar seu impacto, mostrando que a solidariedade é atemporal e universal.

## 2.2 LGPD

De acordo com Brasil (2018), a Lei nº 13.709/2018, conhecida como Lei Geral de Proteção de Dados Pessoais (LGPD), foi estabelecida para salvaguardar os direitos fundamentais de liberdade e privacidade, além da livre formação da personalidade de cada pessoa. A LGPD aborda o processamento de dados pessoais, que podem ser armazenados digitalmente ou fisicamente, realizado por qualquer indivíduo ou entidade, pública ou privada, e inclui uma variedade de operações que podem ser realizadas manualmente ou digitalmente. Os principais fundamentos da LGPD são:

**Respeito à privacidade:** Assegura os direitos fundamentais de inviolabilidade da intimidade, da honra, da imagem e da vida privada.

**Autodeterminação informativa:** Expressa o direito do cidadão ao controle e, assim, à proteção de seus dados pessoais e íntimos.

**Liberdade de expressão, de informação, de comunicação e de opinião:** Direitos previstos na Constituição brasileira.

**Desenvolvimento econômico e tecnológico e a inovação:** Criação de um cenário de segurança jurídica em todo o país.

Livre iniciativa, a livre concorrência e a defesa do consumidor: Por meio de regras claras e válidas para todo o setor privado.

Direitos humanos, o livre desenvolvimento da personalidade, a dignidade e o exercício da cidadania pelas pessoas.

Conforme Rogalewski (2023), dentro do escopo da LGPD, o processamento de dados pessoais pode ser conduzido por dois agentes - o Controlador e o Operador. Existe também a figura do Encarregado, que é designado pelo Controlador para servir como ponto de contato entre o Controlador, o Operador, os titulares dos dados e a (ANPD) Autoridade Nacional de Proteção de Dados.

Um aspecto crucial abordado pela Lei é o processamento de dados, que se refere a qualquer atividade que envolva um dado pessoal em sua operação, como coleta, produção, recepção, classificação, uso, acesso, reprodução, transmissão, distribuição, processamento, arquivamento, armazenamento, eliminação, avaliação ou controle da informação, modificação, comunicação, transferência, difusão ou extração.

Antes de iniciar qualquer tipo de processamento de dados pessoais, o agente deve garantir que o objetivo da operação seja registrado de maneira clara e explícita, e que os propósitos especificados sejam informados ao titular dos dados. No caso do setor público, o principal objetivo do processamento está relacionado à implementação de políticas públicas, conforme previsto em lei, regulamentos ou respaldado por contratos, convênios ou instrumentos semelhantes.

De acordo com Rogalewski (2023), o compartilhamento de dados dentro da administração pública, no contexto da implementação de políticas públicas, é permitido pela lei e não requer consentimento específico. No entanto, o órgão que coleta os dados deve informar de forma transparente quais dados serão compartilhados e com quem. Por outro lado, o órgão que solicita o compartilhamento deve justificar esse acesso com base na implementação de uma política pública específica e claramente definida, descrevendo o motivo do pedido de acesso e o uso que será feito dos dados. Informações protegidas por sigilo permanecem protegidas e estão sujeitas a normas e regras específicas. Essas e outras questões fundamentais devem ser observadas pelos órgãos e entidades da administração federal, a fim de garantir a conformidade do processamento de dados pessoais de acordo com as hipóteses legais e princípios da LGPD.

De acordo com Sales (2020), a lei estabelece um quadro legal de direitos para os titulares de dados pessoais. Esses direitos devem ser assegurados durante todo o período de processamento dos dados pessoais realizado pela entidade ou órgão. Para o exercício dos direitos dos titulares, a LGPD fornece um conjunto de ferramentas que aprofundam as

obrigações de transparência ativa e passiva e estabelecem mecanismos processuais para mobilizar a Administração Pública.

### 2.3 PRIVACIDADE DE DADOS

Conforme Oliveira (2021), a privacidade de dados é um direito que o indivíduo tem de controlar o acesso, uso e divulgação de suas informações pessoais. Isso inclui dados como nome, endereço, número de telefone, informações financeiras, histórico de compras, entre outros.

A privacidade de dados é importante porque, para os indivíduos que estejam dispostos a se envolver online, eles precisam confiar que seus dados pessoais serão tratados com cuidado. As organizações usam práticas de proteção de dados para demonstrar a seus clientes e usuários que podem confiar seus dados pessoais.

De acordo com Donda (2020), os dados pessoais podem ser usados indevidamente de várias maneiras, se não forem mantidas em sigilo ou se as pessoas não tiverem a capacidade de controlar suas informações.

### 2.4 CONFORMIDADE

A conformidade das empresas com a Lei Geral de Proteção de Dados (LGPD) é um processo complexo que envolve várias etapas. Primeiramente, as empresas precisam ter um entendimento complexo da LGPD, incluindo seus princípios, direitos e obrigações. Isso envolve o mapeamento de dados, onde a empresa identifica quais dados pessoais são coletados, armazenados e processados, incluindo as fontes de dados, os processos de coleta, as finalidades e as bases legais para o tratamento.

Além disso, as empresas devem realizar uma análise de riscos para avaliar os riscos associados ao tratamento de dados pessoais, levando em consideração a possibilidade de violação da privacidade e a ocorrência de incidentes de segurança. Para mitigar esses riscos, as empresas devem adotar medidas técnicas e organizacionais para garantir a segurança dos dados pessoais. Isso pode incluir a criação de políticas e procedimentos internos, a realização de treinamentos para os colaboradores e a implementação de sistemas de controle e monitoramento.

Outro aspecto importante da conformidade com a LGPD é a transparência. As empresas devem ter uma política de privacidade clara e acessível, que informe sobre as práticas de



tratamento de dados adotados pela organização. Além disso, se a empresa compartilha dados pessoais com terceiros, deve haver contratos que garantam a conformidade dessas empresas com a LGPD.

Por fim, a LGPD exige que as organizações nomeiem um Encarregado de Proteção de Dados (DPO), que será responsável por garantir o cumprimento da lei dentro da empresa. Portanto, a conformidade com a LGPD é um processo contínuo que requer um compromisso constante das empresas para garantir a proteção dos dados pessoais dos usuários.

## 2.5 IMPORTÂNCIA DO CONSENTIMENTO E DA TRANSPARÊNCIA

De acordo com Sales (2020), a Lei Geral de Proteção de Dados (LGPD) fundamenta-se principalmente na transparência e no consentimento, que são direitos inalienáveis do usuário cujos dados são coletados e armazenados.

Embora seja um desafio que requer atenção constante, a conformidade com a LGPD é um passo crucial não apenas para cumprir a lei, mas também para proporcionar uma experiência mais eficaz e alinhada com as expectativas e necessidades do cliente.

Conforme Quinelato (2022), no que diz respeito à transparência, é vital, especialmente no início da implementação da lei, informar claramente ao cliente quais dados estão sendo coletados e como serão usados. Também é necessário esclarecer quais aspectos do serviço ou produto o cliente não poderá acessar se optar por não compartilhar as informações.

Renata Feijó, diretora jurídica do GuiaBolso, enfatiza que “neste momento de adaptação à LGPD, não basta apenas informar nos termos de uso”. Embora seja um documento de leitura obrigatória, a clareza é o aspecto mais importante da lei. Portanto, é essencial tornar essas informações as mais simples e acessíveis possível.

De acordo com Quinelato (2022), a importância da transparência é ainda mais evidente quando ocorre um vazamento de dados ou outro incidente de segurança. A divulgação ampla do incidente nos meios de comunicação é um dos fatores considerados ao determinar as sanções impostas à empresa.

## 2.6 APLICAÇÕES WEB

As aplicações web evoluíram junto com a internet, acompanhando as mudanças nas necessidades dos usuários. No início, as páginas eram basicamente estáticas, feitas apenas com HTML, e serviam como uma forma simples de exibir informações. De acordo com o World

Wide Web Consortium (W3C), essas tecnologias iniciais formaram a base do que hoje conhecemos como a Web (W3C, 2024). Com o tempo, tecnologias como JavaScript e CSS começaram a ser usadas, tornando os sites mais interativos e visuais.

Nos anos 2000, o conceito de Web 2.0 ganhou força. Nessa época, os sites deixaram de ser apenas vitrines para informações e passaram a permitir interações mais complexas, como comentários, uploads e outras formas de contribuição dos usuários. Isso abriu caminho para plataformas como redes sociais e sistemas mais robustos, como lojas virtuais e serviços online.

Hoje, as aplicações web estão por toda parte, sendo usadas tanto para sites simples quanto para sistemas empresariais completos. Elas têm a vantagem de rodar direto no navegador, sem precisar de instalação, e, com tecnologias como PWAs (*Progressive Web Apps*), oferecem uma experiência parecida com a de aplicativos nativos.

Uma das partes mais importantes de uma aplicação web é o *back-end*, ou seja, o sistema que gerencia os dados, a lógica do negócio e a comunicação com o banco de dados. Para este projeto, escolhemos o Django como base para o desenvolvimento. Esse framework, escrito em Python, é amplamente utilizado no mercado devido à sua robustez, segurança e facilidade de uso. Criado para desenvolver aplicações web rapidamente, o Django segue o princípio de "não se repita" (Don't Repeat Yourself - DRY), o que facilita a organização e a reutilização de código (DJANGO SOFTWARE FOUNDATION, 2024).

Uma das maiores vantagens do Django é o seu sistema de administração automática, que permite gerenciar os dados do sistema de forma rápida e eficiente. Além disso, o framework oferece ferramentas integradas para autenticação, gerenciamento de permissões, tratamento de formulários e até mesmo proteção contra ataques comuns, como CSRF (*Cross-Site Request Forgery*) e injeção de SQL (DJANGO SOFTWARE FOUNDATION, 2024). Isso reduz o tempo necessário para implementar funcionalidades complexas e torna o desenvolvimento mais seguro.

O Django é ideal para aplicações web porque é altamente escalável e flexível. Ele permite construir desde pequenos sites até sistemas mais complexos, como plataformas de e-commerce e redes sociais (HOLVANDER, 2020). Outro ponto importante é que o Django tem uma comunidade ativa e uma documentação extensa, o que facilita a resolução de problemas e a implementação de novas funcionalidades.

No nosso projeto, o uso do Django faz sentido porque queremos garantir um *back-end* robusto e seguro, com a capacidade de lidar com diferentes funcionalidades e um alto volume de usuários. Escolher o Django também nos permite seguir boas práticas de desenvolvimento e entregar uma aplicação confiável e de alta qualidade, alinhada às expectativas do mercado.

## 2.7 ACESSIBILIDADE WEB

A acessibilidade na web é um princípio essencial para garantir que todas as pessoas, incluindo aquelas com deficiência, possam navegar e interagir com sistemas online de maneira eficiente. Segundo as Diretrizes de Acessibilidade para Conteúdo Web (WCAG), desenvolvidas pelo W3C, um site acessível deve ser perceptível, operável, compreensível e robusto. Tem como objetivo conectar ONGs a voluntários de forma eficiente. Como o público-alvo pode incluir pessoas com deficiência visual, auditiva ou motora, a implementação de boas práticas de acessibilidade será fundamental para garantir a inclusão digital.

Principais Recursos de Acessibilidade a Serem Implementados:

- 1- Compatibilidade com Leitores de Tela: Uso de atributos *alt* em imagens, estruturação semântica do HTML e ARIA landmarks para facilitar a navegação.
- 2- Contraste e Legibilidade: Aplicação de cores com alto contraste e fontes de fácil leitura.
- 3- Navegação por Teclado: Garantir que todas as funções possam ser acessadas sem a necessidade do uso do mouse.
- 4- Formulários Acessíveis: Rotulagem adequada de campos e mensagens de erro claras.
- 5- Tamanho Flexível: Possibilidade de redimensionamento de texto sem perda de funcionalidade.

### 3 MATERIAIS E FERRAMENTAS

Para o desenvolvimento deste projeto foram definidas algumas linguagens de programação e frameworks para o *front-end* e *back-end*, além de um sistema de gerenciamento de bancos de dados para armazenamento, os quais serão descritos nos próximos subcapítulos.

O termo *front-end* refere-se à interface visual de uma aplicação, ou seja, a parte do sistema com a qual o usuário interage diretamente, como botões, menus e formulários. Ele é implementado utilizando linguagens como HTML, CSS e JavaScript, além de frameworks específicos como o Bootstrap, que simplifica o desenvolvimento de interfaces responsivas e modernas (W3C, 2024).

Já o *back-end*, ou 'lado do servidor', é a camada de uma aplicação que processa e gerencia os dados do sistema. É responsável por operações como autenticação de usuários, manipulação de dados no banco de dados e lógica de negócios. Ferramentas como o framework Django são frequentemente usadas nessa camada para criar sistemas robustos e seguros (GOYAL et al., 2024).

De acordo com Bylieva (2020), linguagens de programação facilitam a comunicação entre humanos e computadores, traduzindo os conceitos e ideias humanas em instruções compreendidas por máquinas. Diversas linguagens foram desenvolvidas ao longo dos anos para atender a necessidades específicas e resolver problemas diferentes. Cada linguagem possui níveis de complexidade, paradigmas de programação e finalidades distintas.

Pasquali e col. (2024) explicam que de um lado existem as linguagens de baixo nível, como Assembly, utilizada para trabalhar com sistemas embarcados e aplicações de alto desempenho, ela exige que os usuários saibam como as máquinas funcionam internamente. Do outro lado existem as linguagens de alto nível, como Python e Java, as quais oferecem abstrações simplificadas permitindo que os desenvolvedores foquem na lógica algorítmica.

Abstração é a principal característica de uma linguagem de programação, pois de acordo com Wang e col. (2022), ela permite que os desenvolvedores se concentrem nos processos mais críticos de um sistema sem se prenderem a detalhes de processamento de dados de baixo nível e assim lidar com problemas de alto nível, o que beneficia a funcionalidade do software e a manutenção que pode ser realizada posteriormente. Linguagens de programação com abstração de alto nível fornecem aos desenvolvedores classes, objetos, funções, módulos e componentes que compõem a lógica do algoritmo.

Cada linguagem de programação possui características únicas, que afetam a eficiência, legibilidade e facilidade de manutenção do código desenvolvido. A sintaxe e a semântica de

uma linguagem podem tornar o código e os recursos específicos da linguagem mais fáceis ou mais difíceis de entender, detectar e corrigir erros.

Linguagens de programação consistem em códigos escritos em uma mistura de inglês com lógica. O aprendizado de uma linguagem de programação é semelhante ao aprendizado de um idioma, quanto mais leitura e prática maior será o conhecimento sobre a linguagem. Segundo Yuill (2006) e Halpin (2006), esses códigos contêm instruções do que exatamente o computador deve fazer, porém ele não lê esse código diretamente.

O computador pode executar os comandos utilizando um interpretador, programa que lê, traduz e executa o código-fonte em tempo real linha por linha, ou um compilador, programa que transforma o código-fonte diretamente em código de máquina, gerando um arquivo executável que posteriormente será executado pelo computador. O código de máquina é composto de zeros e uns, comandos de registradores e outros elementos. De acordo com Pasquali e col. (2024), cada método possui suas vantagens e desvantagens que afetam o desempenho do software, assim como também a portabilidade e a facilidade de depuração.

Selecionar uma linguagem de programação é um desafio importante no desenvolvimento de software, pois é o que determina o quão eficiente será o processo e a solução construída para o problema proposto. Para este projeto, a linguagem de programação Python, descrita no capítulo 3.2 Python, foi escolhida para o desenvolvimento do *back-end*.

Um *framework* é um projeto de um conjunto de objetos que colaboram entre si para a execução de um conjunto de responsabilidades. Segundo Coad (1992), *framework* é um esqueleto de classes, objetos e relacionamentos agrupados para construir aplicações específicas, enquanto Johnson (1988) diz que é um conjunto de classes abstratas e concretas que provê uma infraestrutura genérica de soluções para um conjunto de problemas. Essas classes podem compor uma biblioteca de classes ou podem ser específicas da aplicação. Frameworks permitem reutilizar componentes isolados ou toda a arquitetura de um domínio específico.

Segundo Fayad (1997), há três tipos de *framework*: *frameworks* de infraestrutura do sistema, utilizados para o desenvolvimento da infraestrutura de sistemas portáteis, *frameworks* de integração de “*middleware*”, utilizados para integrar aplicações e componentes distribuídos e por fim *frameworks* de aplicação empresarial, utilizados em domínios de aplicação mais amplos, como telecomunicações, aviação, manufatura e engenharia financeira. Neste projeto foram utilizados apenas *frameworks* de integração para modularizar, reutilizar e estender a infraestrutura do software.

*Frameworks* são utilizados com o objetivo de reutilizar código, melhorar a produtividade, qualidade e manutenção de softwares. Para este projeto foram escolhidos os

frameworks Bootstrap, descrito no subcapítulo 3.1 Bootstrap, para ser utilizado no desenvolvimento do *front-end*, e Django, descrito no subcapítulo 3.3 Django, para ser utilizado no desenvolvimento do *back-end* e realizar a integração com o *front-end*.

*Back-end*, também conhecido como “lado do servidor”, conforme mencionado anteriormente, é o que está por trás da interface gráfica de um sistema, ele é responsável por proporcionar funcionalidades ao sistema, realizar toda a comunicação entre banco de dados e o navegador executando atualizações, mudanças, segurança, gerenciamento de conteúdo, estrutura e outros. Um servidor fornece serviços aos computadores, enquanto bancos de dados são como uma fonte de informações para os sistemas, e o *back-end* é responsável por manter essas informações seguras.

### 3.1 BOOTSTRAP

O Bootstrap é um dos frameworks de desenvolvimento *front-end* mais frequentemente empregados na criação de interfaces de usuário (UI) responsivas e contemporâneas. Originalmente desenvolvido pelo Twitter, ele disponibiliza uma vasta gama de elementos pré-projetados, tais como botões, formulários, menus de navegação e grades, que simplificam a elaboração de layouts consistentes e visualmente atraentes. Uma de suas maiores vantagens é a responsividade nativa, que possibilita que os websites se adaptem de forma automática a variados tamanhos de tela e aparelhos, um atributo crucial na era móvel. Ademais, o Bootstrap conta com extensa documentação e uma comunidade engajada, o que simplifica o processo de aprendizagem e solução de problemas.

### 3.2 PYTHON

Python é uma linguagem de programação interpretada de tipagem dinâmica, possui estruturas de dados de alto nível eficientes e uma abordagem simples à programação orientada a objetos. É uma linguagem ideal para scripts e desenvolvimento rápido de aplicações. Python está disponível no site oficial do Python, <http://www.python.org/>, gratuitamente em formato de código-fonte ou binário para todas as principais plataformas e pode ser distribuída livremente. O site oficial também disponibiliza a sua documentação, módulos, programas e ferramentas relevantes.

De acordo com Rossum (2003), criador dessa linguagem de programação, e Fred (2003), membro da equipe PythonLabs, ela oferece estrutura e suporte para programas grandes, sendo

ela de alto nível, possui tipos de dados de alto nível integrados, como *arrays* (vetores) flexíveis e dicionários, vem com uma coleção de módulos padrão que pode ser usada como base para programas. Também há módulos integrados que fornecem funcionalidades como entrada e saída de arquivos, chamadas de sistema, sockets e até interfaces para kits de ferramentas de interfaces gráficas, tratamento de exceções e gerenciamento automático de memória. Além de tudo, Python permite dividir o programa em módulos que podem ser reutilizados em outros programas Python. Portanto, essa linguagem pode ser aplicada na resolução de diversos problemas.

Por ser uma linguagem interpretada, o desenvolvedor pode economizar tempo durante o desenvolvimento do programa, pois não há necessidade de compilação. O interpretador pode ser usado de forma interativa, o que facilita experimentar recursos da linguagem, escrever programas descartáveis ou testar funções durante o desenvolvimento de programas de forma incremental. Python permite escrever programas compactos e legíveis, pois o agrupamento de instruções é feito por indentação em vez de colchetes de início e fim, não é necessário declarar variáveis ou argumentos, o que não é da mesma forma com outras linguagens de programação utilizadas no *back-end* de aplicações.

Python é uma linguagem de programação versátil, pois de acordo com Silva (2019) e Silva (2019), ela é capaz de lidar com diversas tarefas, desde o desenvolvimento de sistemas até inteligência artificial, aprendizagem automática ou geoprocessamento. Além disso, quando bem arquitetado, Python fornece legibilidade na produção do código e fácil entendimento.

Segundo Pasquali e col. (2024), Python é capaz de trabalhar com conjuntos de dados grandes e complexos, essenciais para análise decisória, de forma simples e flexível, realizando conexões com fontes de dados, usando as suas bibliotecas e estruturas que simplificam o processo de extração, preparação, análise e visualização de dados e então possibilitando a criação de painéis e relatórios, os quais podem ser utilizados para análises rápidas e eficientes com o objetivo de visualizar e compreender informações cruciais para a gestão empresarial e tomada de decisões. Este projeto, por exemplo, propõe disponibilizar um relatório com a quantidade de inscritos em vagas e seus perfis técnicos para auxiliar ONGs a encontrar voluntários capazes de executar as tarefas solicitadas.

Assim, além de simplificar a criação e manutenção de softwares complexos, Python oferece ferramentas necessárias para um sistema de tomada de decisão. Com tudo, encontra-se um equilíbrio entre entregar tarefas de forma eficaz e ao mesmo tempo usar os recursos oferecidos pelo Python. Levando em consideração todas as características da linguagem, seu fácil aprendizado, principalmente pelo time que desenvolve este sistema, incluindo a

experiência de alguns integrantes, e os objetivos desse projeto se fez necessário escolher essa linguagem de programação para o desenvolvimento do *back-end* da aplicação proposta neste projeto.

### 3.3 DJANGO

Segundo Goyal e col. (2024), Django é um *framework* web Python de alto nível, que roda em muitas plataformas, como Linux, Windows e macOS. Ele permite o desenvolvimento rápido de sites seguros e sustentáveis, desde sistemas de gerenciamento de conteúdo e wikis, até redes sociais e sites de notícias, por exemplo, Instagram e Pinterest utilizam Django. Este *framework* permite que os desenvolvedores se concentrem em escrever um sistema sem precisar reinventar a roda. É gratuito e de código aberto, tem uma ótima documentação e muitas opções de suporte gratuito e pago.

Para desenvolver páginas da web se utiliza HTML, CSS e *frameworks*, como o Bootstrap, descrito no subcapítulo 3.1 Bootstrap, para análise de dados, módulos Python e para realizar a integração entre esses componentes, o Django se mostra uma excelente opção, pois fornece suporte tanto para o *front-end* quanto para o *back-end*, permitindo a disponibilização dos dados e relatórios em página web de fácil acesso para os usuários. Ainda de acordo com Goyal e col. (2024), o Django ajuda a evitar erros comuns de segurança ao fornecer uma estrutura para proteger o site automaticamente. Por exemplo, o Django fornece uma maneira segura de gerenciar contas de usuários e senhas, evitando erros comuns como colocar informações de sessão em cookies onde elas são vulneráveis (em vez disso, os cookies contêm apenas uma chave, e os dados reais são armazenados no banco de dados) ou armazenar senhas diretamente em vez de um *hash* de senha.

Sendo que um *hash* de senha é um valor de comprimento fixo criado utilizando a senha por meio de uma função *hash* que realiza a criptografia. O Django é capaz de verificar se uma senha inserida está correta criptografando-a por meio de uma função *hash* e comparando a saída com o valor *hash* armazenado. Mesmo se um valor *hash* armazenado for comprometido, é difícil para um invasor descobrir a senha original. Além de tudo, Django também habilita proteção contra muitas vulnerabilidades por padrão, incluindo injeção de SQL, script entre sites, falsificação de solicitações entre sites e *clickjacking*.

Ainda segundo Vamsi (2021), ao criar um projeto Django, uma tabela chamada `user_auth` é criada no banco de dados, ela é responsável por realizar a autenticação de usuários. Em vez de buscar os dados diretamente no banco e validar o usuário manualmente, usa-se o



método *authenticate*, que verifica o usuário automaticamente, o que torna o *framework* Django ainda mais eficiente.

Goyal e col. (2024) explicam que Django utiliza uma arquitetura baseada em componentes, onde cada parte é independente, e assim pode ser substituída ou alterada sem afetar outras partes do projeto. Com tudo, um projeto que utiliza esse *framework* pode ser dimensionado para tráfego aumentado adicionando hardware em qualquer nível, como servidores de cache, banco de dados ou de aplicativos. Django também evita duplicação de código desnecessária e que possa ser reutilizado onde for necessário, reduzindo assim a quantidade de código. Além disso, ele também permite a criação de “aplicativos” reutilizáveis, os quais agrupam funcionalidades em comum, e ainda agrupam o código em módulos.

Sistemas web aguardam solicitações HTTP (*Hypertext Transfer Protocol*) do cliente, como um navegador, e quando recebe avalia o que deve ser com base no tipo de solicitação, se é GET, vai ler informações, ou POST, vai guardar informações em um banco de dados, e após o processamento da solicitação, o sistema retorna uma resposta ao cliente. Sistemas desenvolvidos com Django agrupam o código dessas etapas em arquivos separados, conforme descritos abaixo:

Models: definem a estrutura dos dados do sistema, e fornecem ferramentas para manipular os registros no banco de dados;

Templates: é um arquivo de texto que define a estrutura ou o layout de um arquivo, como uma página HTML, com espaços reservados usados para representar o conteúdo real;

URLs: redireciona solicitações HTTP para a *view* apropriada com base na URL da solicitação;

View: recebe as solicitações HTTP e retorna respostas também HTTP.

Uma *view* pode criar dinamicamente uma página HTML usando um *template* HTML, preenchendo-a com dados de um model, acessando os dados necessários para satisfazer a solicitação, enquanto o *template* é responsável por definir a estrutura de qualquer tipo de arquivo, não necessariamente HTML.

Além de tudo Django também oferece outros recursos como, formulários para coletar dados do usuário, autenticação de usuário e permissões, armazenamento em cache para não precisar uma página por completo mais de uma vez, exceto quando necessário, site de administração integrado, que é uma página administrativa usada para gerenciar usuários e o banco de dados, e também facilita a serialização e a veiculação de dados como XML (*Extensible Markup Language*) ou JSON (*JavaScript Object Notation*), que pode ser aplicado na criação de um serviço web ou um site no qual o código do lado do cliente lida com todas as renderizações.

Este projeto propõe um site que possui formulários de cadastro de usuários e ONGs, login, cadastro de vagas para trabalho voluntário e candidaturas a essas vagas. Os dados inseridos por meio dessas páginas são armazenados em um banco de dados e posteriormente recuperados. Por fim, os usuários podem visualizar um mapa com as ONGs ativas no sistema e o alcance do projeto, apresentado em relatórios com a quantidade de ONGs e voluntários registrados e ativos no sistema, vagas já cadastradas, candidaturas realizadas, e a disposição de vagas por área.

### 3.4 HTML

HTML (*HyperText Markup Language*) é a linguagem padrão utilizada para estruturar o conteúdo das páginas web. De acordo com o World Wide Web Consortium (W3C, 2024), o HTML é responsável por fornecer a estrutura básica das páginas, como cabeçalhos, parágrafos, listas e links. É a base para todo conteúdo visível em uma página, permitindo a organização lógica das informações. Ele trabalha em conjunto com outras tecnologias, como CSS e JavaScript, para criar experiências web completas.

### 3.5 CSS

CSS (*Cascading Style Sheets*) é a tecnologia utilizada para definir a aparência e o layout de páginas web. Segundo o W3C (2024), o CSS permite estilizar elementos HTML, controlando cores, fontes, tamanhos e espaçamento. Ele também é essencial para a criação de interfaces responsivas, que se adaptam a diferentes tamanhos de tela, como dispositivos móveis e *desktops*. A separação entre conteúdo (HTML) e apresentação (CSS) melhora a manutenção e a reutilização do código.

### 3.6 JAVASCRIPT

JavaScript é uma linguagem de programação que adiciona interatividade e dinamismo às páginas web. Conforme Flanagan (2020), o JavaScript permite a manipulação de elementos HTML e CSS em tempo real, possibilitando funcionalidades como validação de formulários, animações e atualizações de conteúdo sem a necessidade de recarregar a página. Por ser executado diretamente no navegador do usuário, ele é essencial para a criação de experiências web modernas e interativas.

### 3.7 MYSQL

De acordo com Medeiros (2018) um banco de dados nada mais é do que coleções de dados interligados entre si e organizados para fornecer informações. Para que possamos entender o que é um banco de dados é muito importante saber a diferença entre dados e informações. Dados são fatos brutos, em forma primária, e grande parte das vezes podem não fazer sentido sozinhos. Informações consistem no agrupamento de dados de forma organizada para fazer sentido, gerar conhecimento. Ele é formado por quatro componentes básicos: dados, hardware, software e usuários.

De acordo com Milani (2006) o banco de dados MySQL é considerado um banco de dados robusto, completo e extremamente rápido, e possui todas as características consideradas existentes em outras versões de banco de dados pagas. Conhecido pelas suas versões de uso gratuito, tanto para fins acadêmicos como para a realização de uso comerciais.

Segundo Neves e Ruas (2005) o MySQL é um sistema de gestão de bases de dados relacionais, suporta SQL que é a linguagem estruturada do banco de dados, é um dos SGBDs para utilização profissional mais utilizado e mais conhecido a nível mundial. Se destaca pela sua escalabilidade e estabilidade, possui um alto desempenho na inserção e recuperação de dados.

Segundo Milani (2006) são considerados algumas vantagens do MySQL é que possui o número ilimitado de utilizações pelos seus usuários simultâneos e a capacidade de manipular tabelas que possuem mais de 50.000.000 de registros, a alta velocidade para a execução de comandos e a facilidade e eficiência no controle de privilégios de seus usuários.

O MySQL disponibiliza ao usuário vários tipos de tabelas para armazenamento de dados, tendo cada tipo com suas características específicas. A grande vantagem dessa variedade de tabelas é a possibilidade de escolher o tipo em cada contexto submetido. Enquanto um tipo pode vir a priorizar velocidade, outro pode ter como função priorizar os volumes de dados.

### 3.8 HOSPEDAGEM

A hospedagem de aplicações web em um provedor de nuvem é um processo essencial para tornar o projeto acessível aos usuários finais. Ela envolve a configuração de um ambiente que atenda aos requisitos mínimos necessários para o funcionamento do *front-end* e do *back-end*, considerando as características e demandas específicas de cada um.

No *front-end*, que representa a interface de interação com os usuários, é crucial hospedar os arquivos estáticos, como HTML, CSS e JavaScript, em um ambiente estável e de fácil acesso. O conteúdo deve ser otimizado para garantir tempos de carregamento rápidos. Os principais passos para o *deploy* do *front-end* de uma aplicação incluem:

**Geração do Build:** A aplicação é empacotada em uma versão otimizada para produção, garantindo maior estabilidade.

**Armazenamento dos Arquivos:** Os arquivos são enviados para um serviço de hospedagem que permita o acesso público.

**Configuração de Domínio e HTTPS:** Para proporcionar uma experiência profissional e segura, configura-se um domínio personalizado e certificados de segurança (SSL/TLS).

O *back-end*, que é responsável pela lógica do sistema e pela interação com bancos de dados, exige um ambiente de execução bem configurado, preparado para suportar o *runtime* da aplicação. Gerenciar conexões de forma eficiente e garantir a segurança das informações são aspectos fundamentais. Embora a segurança seja uma prioridade, a eficiência também é indispensável, já que o *back-end* consome mais recursos computacionais que o *front-end*.

Os principais passos para o *deploy* do *back-end* da aplicação incluem:

**Configuração do Ambiente:** Criação de um ambiente compatível com os requisitos do sistema, como linguagens de programação e dependências.

**Upload e Configuração do Código:** O código-fonte é enviado e ajustado para execução no ambiente de produção.

**Definição de Variáveis de Ambiente:** Configuração segura de informações sensíveis, como chaves de API e credenciais.

**Segurança:** Aplicação de regras de firewall e controle de acesso para proteger a aplicação.

**Integração entre *Front-end* e *Back-end*.**

A hospedagem eficiente de uma aplicação web exige atenção a diversos fatores, desde a configuração do ambiente até a segurança e integração dos sistemas. A correta configuração do *front-end* assegura que a interface seja acessível, rápida e profissional, enquanto o *deploy* do *back-end* garante a execução eficiente das funções críticas do sistema, com foco em segurança e desempenho.

Além disso, o processo de integração entre *front-end* e *back-end* desempenha um papel estratégico, pois define como os diferentes componentes da aplicação trabalham juntos para atender às demandas necessárias.

### 3.9 MICROSERVIÇOS

Os microsserviços representam uma abordagem arquitetural para o desenvolvimento de software que busca a modularização de sistemas, dividindo-os em pequenas aplicações independentes que se comunicam entre si. Diferente da arquitetura monolítica, onde todos os componentes do sistema estão interligados em um único bloco de código, os microsserviços operam de forma isolada, permitindo maior escalabilidade, flexibilidade e manutenção simplificada (FOWLER, 2017).

Cada microsserviço é responsável por uma funcionalidade específica do sistema e pode ser desenvolvido, testado e implantado de forma independente. Além disso, esses serviços geralmente se comunicam por meio de APIs (*Application Programming Interfaces*), utilizando protocolos como HTTP/REST ou mensagens assíncronas via filas, como RabbitMQ ou Kafka. Essa estrutura possibilita que diferentes partes do sistema evoluam de maneira independente, facilitando atualizações e melhorando a resiliência da aplicação.

Segundo FOWLER (2017), a adoção de microsserviços traz diversas vantagens, especialmente para aplicações que exigem alta disponibilidade e desempenho. Entre os principais benefícios, destacam-se:

**Escalabilidade:** Cada serviço pode ser dimensionado conforme sua necessidade, permitindo a alocação eficiente de recursos.

**Facilidade de manutenção:** Como os serviços são independentes, é possível modificar ou substituir um deles sem afetar os demais.

**Resiliência:** Em caso de falha de um microsserviço, os outros continuam operando, minimizando impactos no sistema.

**Tecnologia diversificada:** Cada serviço pode ser desenvolvido em diferentes linguagens de programação ou frameworks, conforme a necessidade.

A arquitetura baseada em microsserviços desempenha um papel essencial na modularização e eficiência do sistema web desenvolvido para conectar voluntários a serviços sociais cadastrados por ONGs. Essa abordagem permite que diferentes funcionalidades sejam tratadas de forma independente, facilitando a manutenção, escalabilidade e aprimoramento contínuo da plataforma. Além disso, possibilita a integração de novos serviços conforme as necessidades evoluem, garantindo maior flexibilidade na adaptação do sistema.

### 3.10 BRASIL API

Esta API disponibiliza gratuitamente *endpoints* com baixa latência para a consulta das seguintes informações: Código de Endereçamento Postal (CEP), Discagem Direta à Distância (DDD), informações sobre sistema bancário, Cadastro Nacional da Pessoa Jurídica (CNPJ), informações sobre estados provenientes do Instituto Brasileiro de Geografia e Estatística (IBGE), feriados nacionais, preço médio de veículos fornecido pela Fundação Instituto de Pesquisas Econômicas (FIPE), informações sobre livros publicados no Brasil a partir do Número Internacional Padrão de Livro, registros de domínio br, taxas de juros e alguns índices oficiais do Brasil, entre outras informações. A sua documentação pode ser encontrada na URL <<https://brasilapi.com.br/docs>>.

Esta ferramenta foi utilizada para realizar a consulta do CEP e DDD de voluntários, além do CNPJ das ONGs para garantir que os usuários informem dados reais e válidos para o sistema. Esta API se tornou uma escolha válida pois é gratuita, de fácil manipulação e possui uma documentação completa.

### 3.11 BIBLIOTECA VALIDATE-DOCBR

Esta biblioteca Python permite validar gratuitamente os seguintes documentos brasileiros: Cadastro de Pessoas Físicas (CPF), Carteira Nacional de Habilitação (CNH), CNPJ, Cartão Nacional de Saúde (CNS), Programa de Integração Social (PIS) e título eleitoral, o cadastro que permite cidadãos brasileiros votar. Ela é útil pois facilita a validação desses documentos de forma fácil sem precisar desenvolver uma solução nova para essas situações. A sua documentação pode ser encontrada na URL <<https://alvarofpp.github.io/validate-docbr/>>.

Para utilizar este pacote basta realizar a sua instalação, importar o documento que deseja validar e então verificar. Este pacote foi utilizado para validar o número do CPF dos voluntários ao tentarem realizar um cadastro de usuário no sistema e assim não permitindo a entrada de usuários com documentos falsos, por fim facilitando a validação dos usuários pelos administradores do sistema.

### 3.12 OPENCAGE GEOCODER

O uso da ferramenta OpenCage Geocoder como ponte com a API do OpenStreetMap classificou de forma eficaz as coordenadas de diversos locais, incluindo organizações não governamentais para serem plotados em um mapa. Tais etapas são viáveis para identificar e

buscar dados espaciais, conectando-se diretamente às seguintes áreas de computação: Sistemas Distribuídos, Ciência de Dados e Databases.

Quando acessa o OpenCage Geocoder API, você está interagindo com um sistema distribuído. Um sistema distribuído é composto por vários componentes que frequentemente ficam separados em várias máquinas ou servidores. Neste caso, o código está requerendo a localização do servidor remoto da OpenCage, que por sua vez, retorna a localização geográfica com base na solicitação passada ao sistema, ou simplesmente o nome ou endereço de uma ONG.

Outrossim, a geolocalização é útil também na Ciência de Dados, quando se trabalha com análise de dados espacial. Utilizando as coordenadas geográficas coletadas pela requisição à API, podemos conceber análises que nos auxiliem a compreender a dispersão das ONGs, por exemplo saber onde são mais comuns e traçar grupos de localização entre as classes. Isso poderia suscitar descobertas interessantes sobre a eficácia dos investimentos e impacto social, por exemplo. Usando técnicas de clusterização, é possível agrupar as ONGs em regiões semelhantes, e análises de proximidade podem revelar quais organizações estão mais próximas entre si, ajudando na criação de um mapa de rede das ONGs.

Armazenar essas coordenadas é fundamental e ressalta a importância dos Bancos de Dados. Depois de coletar as coordenadas geográficas das ONGs, o próximo passo é guardá-las de forma eficiente para futuras consultas. Os bancos de dados geoespaciais, como o MySQL, são feitos para lidar com esse tipo de informação. Nesses bancos, é possível criar tabelas que armazenam as coordenadas de latitude e longitude das ONGs. Isso torna mais fácil realizar consultas espaciais, como encontrar ONGs dentro de um determinado raio ou calcular distâncias entre elas.

### 3.13 FOLIUM

A biblioteca Folium é uma ferramenta eficaz para a criação de mapas interativos, sendo empregada em projetos de visualização de dados geoespaciais, como no caso de mapear a localização de ONGs. Ela é construída sobre o Leaflet.js, uma biblioteca popular de JavaScript voltada para mapas, permitindo a renderização de mapas diretamente na interface de aplicações web. Com ela, fica fácil integrar dados geoespaciais, incluindo coordenadas de latitude e longitude, e visualizá-los de maneira interativa, seja através de marcadores, camadas ou mapas de base.

As coordenadas das ONGs são obtidas por meio da API do OpenCage Geocoder, a Folium pode ser empregada para marcar esses pontos em um mapa. Geralmente, o

procedimento consiste em criar um objeto de mapa, acrescentar marcadores que representam cada ONG em seus respectivos locais e, por fim, exibir o mapa em uma página web. Ao combinar a biblioteca Folium com ferramentas de georreferenciamento, como o OpenCage Geocoder ou outras APIs de geocodificação, essencialmente une o potencial do georreferenciamento, técnica de associar dados a coordenadas geográficas, com a capacidade de visualização e interação oferecida pela Folium. Essa técnica permite mapear informações espaciais, como as localizações das ONGs, às coordenadas geográficas reais (latitude e longitude), através do uso de APIs que convertem nomes de lugares ou endereços em coordenadas precisas.

Com as coordenadas de latitude e longitude em mãos, obtidas via API de geocodificação, a Folium possibilita criar mapas interativos onde as ONGs podem ser representadas por marcadores posicionados exatamente nos locais. Esses marcadores podem receber personalizações em cores ou ícones para destacar diferentes aspectos, além de poderem exibir informações adicionais, como nomes, detalhes ou serviços de cada ONG. Haverá um link direto para o perfil, permitindo que o usuário acesse mais informações ou entre em contato facilmente.

### 3.14 CHART.JS

No âmbito de Ciência de Dados e Inteligência de Negócios, a apresentação visual de dados é crucial para converter informações complexas em percepções simples e compreensíveis. Com o uso da biblioteca Chart.js, foram criados painéis interativos que se conectam a bases de dados, possibilitando a atualização em tempo real das informações exibidas. Esses painéis foram elaborados para satisfazer as exigências de análise e supervisão de dados estratégicos, tais como o total de candidaturas mensais, a comparação entre vagas disponíveis e ocupadas, a interação entre ONGs e voluntários, o número de vagas por setor e a trajetória das vagas abertas ao longo dos meses.

A implementação do Chart.js permite a elaboração de gráficos responsivos que são de fácil compreensão, fundamentais para a criação de narrativas visuais que facilitam decisões fundamentadas em dados. Ao conectar essas visualizações a fontes de dados organizadas, é viável produzir relatórios dinâmicos e monitorar métricas essenciais de desempenho em tempo real, fomentando uma cultura direcionada por dados tanto em iniciativas sociais como em contextos empresariais. Assim, a integração entre análise de dados, visualização eficaz e



ferramentas de Inteligência de Negócios contribui diretamente para a criação de soluções mais rápidas, estratégicas e sustentadas por evidências.

### 3.15 BANCO DE DADOS NA AWS E HOSPEDAGEM

Para garantir a robustez, a segurança e o desempenho do sistema web proposto, foram utilizadas ferramentas modernas e amplamente reconhecidas no ecossistema de desenvolvimento e infraestrutura de aplicações web.

#### 3.15.1 Banco de Dados na AWS

O banco de dados utilizado para o armazenamento de informações de usuários, ONGs e vagas de voluntariado foi implementado na Amazon Web Services (AWS) por meio do serviço Amazon RDS (*Relational Database Service*), com as seguintes características:

Tipo de Banco: MySQL, pela sua robustez, compatibilidade com aplicações modernas e recursos avançados de segurança e integridade transacional.

Segurança: Implementação de grupos de segurança (*Security Groups*), criptografia de dados em repouso (*Encryption at rest*) e backups automatizados.

Alta Disponibilidade: Uso de *Multi-AZ Deployment*, garantindo resiliência e tolerância a falhas.

#### 3.15.2 Hospedagem do Sistema

A aplicação, disponibilizada na URL <<https://weong.onrender.com/vagas/>>, foi hospedada na plataforma Render utilizando a infraestrutura em nuvem para assegurar escalabilidade, alta disponibilidade e aderência às boas práticas de desenvolvimento web. A escolha pelo Render se deu por sua capacidade de simplificar o processo de deploy contínuo, garantindo integração eficiente com repositórios Git. Os principais recursos empregados na hospedagem foram:

Render Web Services: Serviço responsável por provisionar o ambiente de execução da aplicação Django, permitindo a configuração automática de variáveis de ambiente, gerenciamento de instâncias e *deploy* contínuo a partir do repositório Git. A infraestrutura oferecida pelo Render assegura balanceamento de carga, reinicialização automática em caso de falhas e escalonamento de recursos de acordo com a demanda da aplicação.

Gerenciamento de arquivos estáticos e mídia: A plataforma possibilita a integração com serviços externos de armazenamento, garantindo o correto manuseio de arquivos estáticos e de mídia da aplicação Django. Embora o Render não ofereça um serviço nativo equivalente ao Amazon S3, foi configurado um provedor externo para essa finalidade, assegurando o desempenho e a organização dos ativos.

Certificados SSL (via *Render SSL Management*): A plataforma disponibiliza, de forma automatizada, a emissão e renovação de certificados SSL gratuitos, assegurando comunicação segura por meio do protocolo HTTPS. Essa prática garante a confidencialidade e integridade das informações trafegadas, atendendo às exigências da Lei Geral de Proteção de Dados (LGPD) e demais regulamentações aplicáveis.

Essas soluções em nuvem foram escolhidas por sua flexibilidade, escalabilidade e ampla aceitação no mercado. A combinação dessas ferramentas garante que o sistema atenda não apenas aos requisitos técnicos do projeto, mas também aos padrões atuais de segurança e usabilidade exigidos por aplicações voltadas ao terceiro setor.

### 3.16 GIT

É importante documentar o código desenvolvido na construção de um sistema para rastreamento de versões, análise, compartilhamento, saber o que adicionar ou retirar, o que deve ser aprimorado sem quebrar o que já está funcionando. Uma solução é armazenar uma cópia do código antes de editar, porém isso pode gerar muitos arquivos, fica difícil saber a diferença entre um arquivo e outro, e o que cada arquivo faz ou deixa de fazer, principalmente se passar muito tempo sem mexer no código, além de dificultar o compartilhamento de todas essas versões e suas particularidades entre outras pessoas.

Para solucionar esses problemas, de acordo com Blischak, Davenport e Wilson (2016), foi criado o controle de versão que gera um histórico das alterações feitas no código, permitindo acessar cada versão com a possibilidade de reverter as alterações, além de permitir documentar cada atualização e ao mesmo tempo compartilhar tudo isso com outras pessoas.

O Git é um sistema de controle de versão distribuído, gratuito e de código aberto, projetado para lidar com tudo, desde projetos pequenos até muito grandes, com rapidez e eficiência. Essa ferramenta permite versionar códigos, armazenar histórico de alterações permitindo o seu rastreio e o compartilhamento com outros desenvolvedores, onde o ato de salvar uma alteração é chamado de *commit* que acompanha uma descrição do que foi feito.

Uma característica importante do Git é o seu modelo de ramificação ou *branch*, que de acordo com Louzada (2024), permite desenvolver código separadamente do código principal do projeto sem afetar o que já está pronto. Cada *branch* é uma ramificação independente do código-fonte, possibilitando que isole e desenvolva novas funcionalidades, refatore o código ou faça correções e testes em paralelo, sem interferir no código existente na *branch* principal, que geralmente é nomeada como "*main*". Em um projeto com *branches* diferentes, a fusão, ou merge, permite combinar as alterações dessas *branches* de volta ao código principal, quando as alterações estão prontas.

O Git salva snapshots das alterações feitas nos arquivos salvando uma versão permanente sempre que for solicitado e após isso é possível compartilhar com outros colaboradores publicando o código online, o que pode ser feito utilizando serviços de hospedagem de repositórios, como o GitHub, descrito no Subcapítulo 3.17 GitHub. Mais detalhes sobre essa tecnologia podem ser encontrados na URL <<https://git-scm.com/>>.

Os colaboradores deste projeto buscaram seguir boas práticas ao usar o Git aplicando padrões e nomenclaturas em nomes de *branches* e *commits*, descritos a seguir:

Prefixos de *branches*:

- *bugfix*/: são feitas para corrigir erros de forma imediata;
- *feature*/: são feitas para criar funcionalidades;
- *hotfix*/: correção de cor, textos, sendo alterações não tão urgentes;
- *improvement*/: melhoria de algo já existente no projeto.

Prefixos de *commits*:

- *feat*: utilizado em *commits* de novas funcionalidades;
- *fix*: para *commits* de correção;
- *docs*: para alterações relacionadas à documentação;
- *style*: para alterações no CSS;
- *refactor*: para melhorias realizadas em códigos já existentes;
- *perf*: utilizado em alterações relacionadas a performance;
- *test*: utilizado em testes;
- *chore*: para alterações relacionadas a build, configurações e afins.

### 3.17 GITHUB

O GitHub é uma plataforma baseada em nuvem em que é possível armazenar, compartilhar e trabalhar com outros colaboradores desenvolvendo códigos, inclusive é graças

ao Git que é possível trabalhar em equipe em um projeto, e ele serviu como base para a criação do GitHub. Essa ferramenta permite demonstrar ou compartilhar projetos, acompanhar e gerenciar alterações no código ao longo do tempo, que outras pessoas revisem o código e deem sugestões para melhorá-lo, colaborar em um projeto compartilhado sem se preocupar que as alterações afetarão o trabalho de outros colaboradores.

Os arquivos carregados no GitHub ficam armazenados em um “repositório Git”, ou seja, quando faz alterações nos arquivos que estão no GitHub, o Git monitora e gerencia essas alterações automaticamente. Algumas funcionalidades do Git podem também ser utilizadas no GitHub, como criar um repositório Git, criar *branches*, carregar e editar arquivos. Ao longo do desenvolvimento deste projeto os colaboradores carregaram as alterações da própria máquina para o GitHub e ao mesmo tempo foram obtendo novas atualizações dos códigos do GitHub para o próprio computador. Mais detalhes sobre essa ferramenta podem ser encontrados na URL <<https://docs.github.com/pt>>.

### 3.18 IPWHOIS.IO

Esta é uma API de geolocalização gratuita capaz de retornar informações sobre endereço ou domínio de Protocolo de Internet Versão 4 (IPv4) e versão 6 (IPv6) que fornece resultados em formato JSON, XML ou *Comma Separated Values* (CSV). Este serviço utiliza inteligência artificial e analisa dados de diversas fontes, garantindo as informações de geolocalização mais precisas, além do banco de dados ser atualizado quase em tempo real, garantindo precisão e sem intervenção manual. Essa API permite até 10.000 solicitações por mês gratuitamente, sendo assim ela se torna inicialmente uma opção viável para esse projeto. A sua documentação pode ser encontrada na URL <<https://ipwhois.io/documentation>>.

A API pode ser consumida enviando requisições HTTP GET para [https://ipwho.is/\[IP address\]](https://ipwho.is/[IP address]), onde *IP address* pode ser um endereço IPv4 ou IPv6, ou nenhum para usar o endereço IP de onde está sendo feita a requisição. Este projeto realiza essa requisição sempre que um usuário acessar o mapa de ONGs, pois é necessário obter a localização do usuário para que então o sistema disponibilize as ONGs da cidade que o usuário se encontra.

## 4 METODOLOGIA

Neste capítulo, são aqui apresentados os métodos seguidos na realização de software para ONGs para simplificar as conexões entre voluntários e as instituições sociais.

O sistema foi desenvolvido para permitir que as pessoas interessadas tenham acesso a tarefas voluntárias que estejam de acordo com seus interesses e habilidade, o que lhes dá a escolha em várias áreas de trabalho disponíveis.

Alinhado com os princípios de usabilidade de Krug (2014) para prover habilidade e alcançar essa finalidade utilizamos uma abordagem com ênfase na experiência do usuário, garantindo que esta seja intuitiva e eficaz, envolvendo desde a pesquisa e entendimento do “problema”, o processo de projetar e abordar a plataforma até a instalação e teste que permitem ao pessoal ONG conectar mais simplificada e mais dinamicamente com os voluntários, fazendo com que a participação seja mais fácil e mais dinâmica.

### 4.1 TÉCNICAS DE PESQUISA

No contexto atual, as preocupações ambientais, climáticas e populacionais, frente à exponencial demanda de voluntários capacitados, representam cada dia mais um fator crítico na realização das atividades de diversos segmentos da sociedade. As organizações não-governamentais (ONGs), por sua vez, enfrentam o duplo desafio de otimizar as suas ações e minimizar o desperdício dos recursos disponíveis. Por esta e outras razões, o presente estudo, com abordagens tanto quantitativas quanto qualitativas, explora estratégias para a implementação de software visando enfrentar tais obstáculos. Baseando-se no uso de técnicas de *design pattern* como MVC (Gamma et al., 1994) com foco na experiência do cliente, otimização de projetos e disponibilidade de voluntários. A pesquisa inicial foi conduzida através do levantamento de conjuntos de dados públicos sobre as instituições e seus critérios de regularização frente ao país de atuação e, realizando em paralelo, visitas/conversas para o levantamento de dados e a solicitação regulamentada para permitir a utilização, edição e replicação dos dados. O critério final de escolha para as ONGs levou em consideração três características quantitativas principais:

Visando tanto garantir a qualidade do projeto antes da implementação, quanto aumentar o seu impacto na sociedade, foram realizadas pesquisas técnicas para aplicação de etapas de limpeza dos dados, padronização e normalização das informações. Estas, por sua vez, demonstraram que a utilização de técnicas de ETL podem ser alternativas eficazes para otimizar

a disponibilidade de recursos (Han, J., 2011). O sistema desenvolvido neste trabalho apresenta resultados promissores, com potencial para aplicação ampla no mercado, como soluções SaaS

A melhoria da experiência dos voluntários e a diminuição do desperdício de recursos pelas ONGs foram alcançadas através de uma abordagem multifacetada que envolveu o treinamento de funcionários da ONGs e divulgação eficiente das iniciativas. Tais práticas não apenas elevam a entrega final, mas também promovem a sustentabilidade ambiental, consciência social e inclusão.

## 4.2 METODOLOGIA DE DESENVOLVIMENTO

Segundo Koscianski (2007) a metodologia de desenvolvimento de software é um conjunto de técnicas, práticas e processos utilizados para a criação de um programa de computador. A metodologia é utilizada em todo o ciclo de vida de um software, desde a sua concepção até a sua entrega final. Existem diversas metodologias, cada uma com suas abordagens e características específicas. Com base no trabalho apresentado, utilizaremos o método Incremental, que, diferentemente das metodologias em cascata, é uma abordagem ágil, iterativa e incremental.

### 4.2.1 Modelo Incremental

Segundo Amaral (2017), o modelo incremental é uma abordagem de desenvolvimento de software que visa melhorar o tradicional modelo em cascata. Diferente do modelo em cascata que desenvolve todo o sistema de uma só vez, o Incremental trabalha com incrementos, ou seja, pequenas partes do desenvolvimento são entregues progressivamente. Segue abaixo algumas características desta:

A) Núcleo do Produto: o primeiro incremento feito é nomeado de “núcleo do produto” que contém a implementação dos requisitos básicos necessários para que o sistema funcione e atenda às necessidades básicas do cliente;

B) Iterativo e Rápido Feedback: Diferente de outros modelos, as atividades de Projeto, Validação, Implementação e Especificações são intercaladas em cada nova versão feita, permitindo feedbacks rápidos entre suas etapas e o ajuste em tempo de entrega;

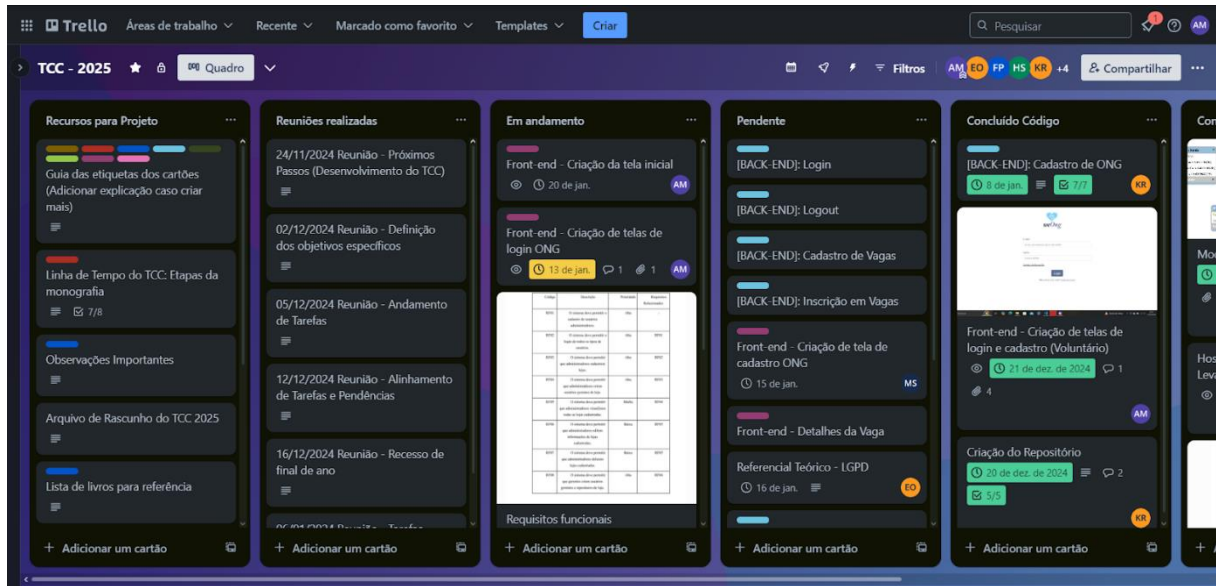
C) Prioridades e Entregas Graduais: Pode ser estabelecido prioridades de desenvolvimentos do sistema, o que acabam resultando em entregas mais dinâmicas;

D) Desenvolvimento por Incremento: Cada um dos incrementos é desenvolvido de forma linear, semelhante ao modelo em cascata, porém após cada fase ser implementada, esta é exposta aos comentários dos clientes para garantir o alinhamento das expectativas.

Neste projeto, foi empregado o método Kanban para organização dos membros em subgrupos, sendo definidas para cada subgrupo, suas próprias metas, entregas e demandas. Para o desenvolvimento do projeto, entre os diversos objetivos distribuídos entre os subgrupos, inclui-se o desenvolvimento e ajustes do *back-end*, a determinação e organização do banco de dados para o armazenamento dos cadastros, o desenvolvimento do *front-end* e as métricas de avaliação da experiência dos usuários (UX). Com essa estrutura para os objetivos, a divisão em subgrupos facilitou o desenvolvimento do projeto. Orientada por objetivos específicos, a estratégia de divisão escolhida facilitou o desenvolvimento eficiente do projeto. Sendo assim, a metodologia teórica consistiu em aplicar no sistema uma abordagem que tem como principais características a gestão de recursos e processos, quais visam aumentar a eficiência da comunicação, auxiliar no controle do estoque, otimizar a produção e ter uma melhor comunicação entre as equipes, permitindo uma resposta com mais agilidade às mudanças e demandas do processo.

Com isso, se fez possível gerir melhor os nossos objetivos, e para uma organização mais estruturada utilizaremos a plataforma Trello, uma plataforma de gerenciamento de projetos baseada em nuvem que utiliza o método Kanban para organizar tarefas, conforme demonstrado na Figura 01. Ele é utilizado em diferentes contextos, desde gerenciamento de projetos pessoais até colaboração em equipes de trabalho, permitindo atribuir, distribuir e colocar datas de conclusão para uma melhor organização das tarefas, para que todo o projeto esteja alinhado com os objetivos citados. Além de organizar as notas, podemos atribuir tarefas para cada membro, e acompanhar o desenvolvimento de cada um e, caso necessário, fazer ajustes durante o desenvolvimento.

Figura 01 - Kanban no site Trello



Fonte: Os autores.

Mediante a aplicação do Trello, foi elaborado um quadro que consiste em delineamento de fases rumo ao nosso objetivo, a saber:

**Ideias e Referências:** Esta parte diz respeito à elaboração de um documento que contém uma lista de ideias, referências e artigos entre outros materiais relevantes identificados durante a pesquisa inicial ou adicionados durante o desenvolvimento do projeto.

**Planejamento:** A lista é dividida em cartões para cada fase do projeto, como pesquisa teórica, desenvolvimento de algoritmos, testes, elaboração da parte escrita do TCC. É o momento em que são definidos os prazos e etiquetas para priorização das tarefas.

**Desenvolvimento do Sistema:** Criação de cartões para cada etapa do desenvolvimento da plataforma para cadastramento das ONGs e voluntários, como a página de login, a conectividade das telas de registro com o banco de dados.

**Testes e Avaliação:** Nesta fase são realizadas duas baterias de testes. Na primeira são criados os cartões e realizados os procedimentos para cada etapa de teste do sistema, incluindo testes de responsividade das páginas, análise do tempo de execução e testes de desempenho. Já na segunda, é feita uma validação final do sistema como um todo, permitindo assim obter tanto uma melhor percepção de cada fase individualmente quanto do projeto como um todo.

**Escrita do TCC:** É feita a divisão do trabalho escrito em seções, como introdução, revisão de literatura, metodologia, resultados e conclusão para distribuição das atividades entre os integrantes.

**Revisão e Correção:** Momento crucial, no qual todos os integrantes realizam a revisão minuciosa de suas respectivas partes no trabalho, sendo criadas etiquetas para acompanhar as



revisões e correções feitas no TCC, incluindo feedback do orientador e quaisquer outros ajustes necessários.

Finalização e Entrega: Diz respeito às etapas finais do projeto, como formatação do documento, elaboração da apresentação, preparação das falas, revisão final e entrega do TCC.

Cada um dos pontos apresentados previamente permitiu o desenvolvimento do projeto de forma eficiente e o atingimento dos objetivos de uma forma organizada, ágil e controlada.

## 5 MODELAGEM

De acordo com Booch, Jacobson e Rumbaugh (2006), a modelagem permite visualizar o sistema como é desejado, especificar a estrutura e o comportamento, serve de guia para a construção e como documentação para as decisões que foram e podem ser tomadas futuramente. Quanto maior e mais complexo for um sistema, mais importante a modelagem se torna, pois com ela é possível delimitar o problema e restringir o foco em um subproblema por vez.

Segundo Machado (2015), um sistema de boa qualidade deve estar de acordo com os requisitos pré-estabelecidos e para isso é importante que seja realizado o entendimento dos requisitos definindo as informações a serem processadas, funções que o sistema deve realizar (Requisitos Funcionais), interfaces que serão disponibilizadas para o usuário, critérios para validações e as restrições do projeto (Requisitos Não Funcionais). É no processo de análise de requisitos que se obtém o máximo de informações sobre o problema em questão, abstraindo conceitos sobre a aplicação para entender o que está sendo construído e deixar claro para os membros da equipe de desenvolvimento o que o cliente está buscando e reduzir ao máximo a complexidade.

Em engenharia de software existem os modelos de análise e os modelos de projeto. Enquanto os modelos de análise se preocupam com os requisitos do cliente, como informações necessárias e funções desejadas, os modelos de projeto se preocupam com as características de software que auxiliam os profissionais na construção do sistema, como a arquitetura, interface e componentes. Existem diferentes tipos de modelos que são capazes de representar as características de diferentes sistemas, porém é importante que as ferramentas utilizadas sejam gráficas, permitindo a visualização do sistema de forma subdividida e de fácil entendimento para o leitor.

Booch, Jacobson e Rumbaugh (2006) dizem que o método orientado a objetos vem provando seu valor para a construção de sistemas complexos. Neste método são utilizados dois conceitos importantes: objetos, que podem ser tidos como objetos do mundo real, e classes, que é a descrição de um conjunto de objetos. A Linguagem Unificada de Modelagem (UML) tem como objetivo visualizar, especificar e construir a documentação de sistemas orientados a objetos. A UML permite a expressão de requisitos, realização de testes, modelagem das atividades do projeto e gerenciamento de versões.

A UML é baseada em três blocos de construção: itens, relacionamentos e diagramas. Itens são os principais blocos de um modelo, os relacionamentos unem os itens e os diagramas agrupam coleções de itens projetando um determinado sistema e permitindo a visualização sob

diferentes perspectivas. Dentre os diagramas existentes na UML é possível citar o Diagrama de Classes, Diagrama de Casos de Uso e Diagrama de Atividades.

## 5.1 REQUISITOS FUNCIONAIS

Segundo Sommerville (2018), os requisitos funcionais descrevem o que o sistema deve fazer, a reação em entradas específicas e em determinadas situações. Além disso, pode estabelecer explicitamente o que o sistema não deve fazer. Primeiramente os requisitos do sistema são definidos como requisitos de usuários, pois assim eles são compreendidos pelos usuários e pelos gerentes do projeto e a partir disso são criados os requisitos funcionais para os desenvolvedores descrevendo com detalhes as funções do sistema, entradas, saídas e exceções.

A especificação dos requisitos funcionais deve ser completa e coerente, porém isso se torna difícil em sistemas grandes e complexos. As necessidades descritas pelo cliente podem ser inconsistentes ou o time de gerentes e desenvolvedores podem entender de forma equivocada e isso pode ser descoberto apenas após uma análise mais profunda ou durante o desenvolvimento levando assim a criação de novos requisitos ou ajustes em requisitos existentes resultando em atraso na entrega e aumento de custos. No Quadro 01 são descritos os requisitos funcionais determinados para este projeto durante o desenvolvimento da documentação e a prioridade de cada um.

Quadro 01 – Requisitos Funcionais.

Código	Descrição	Prioridade	Requisitos Relacionados
RF01	O sistema deve permitir o cadastro de usuários administradores.	Alta	-
RF02	O sistema deve permitir o login de todos os tipos de usuários.	Alta	RF01

RF03	O sistema deve permitir que administradores cadastrem ongs.	Alta	RF02
RF04	O sistema deve permitir que administradores criem usuários coordenadores de ong.	Alta	RF03
RF05	O sistema deve permitir que administradores visualizem todas as ongs cadastradas.	Média	RF04
RF06	O sistema deve permitir que administradores editem informações de ongs cadastradas.	Baixa	RF05
RF07	O sistema deve permitir que administradores deletem ongs cadastradas.	Baixa	RF05
RF08	O sistema deve permitir que coordenadores criem usuários coordenadores e funcionários de ongs.	Alta	RF04
RF09	O sistema deve permitir que os coordenadores visualizem a lista de usuários funcionários.	Baixa	RF08

RF10	O sistema deve permitir que coordenadores alterem perfis de funcionários.	Baixa	RF08
RF11	O sistema deve permitir que todos os tipos de usuários visualizem o próprio perfil.	Baixa	RF02
RF12	O sistema deve permitir que todos os tipos de usuários alterem informações do próprio perfil.	Baixa	RF11
RF13	O sistema deve permitir que administradores e coordenadores deletem o próprio perfil.	Baixa	RF11
RF14	O sistema deve permitir que os coordenadores deletem perfis de funcionários.	Baixa	RF11
RF15	O sistema deve permitir que coordenadores e funcionários visualizem as ongs pertencentes.	Alta	RF02
RF16	O sistema deve permitir que coordenadores e funcionários visualizem os estados das ongs.	Alta	RF15

RF17	O sistema deve permitir que coordenadores e funcionários alterem o estado das vagas.	Alta	RF16
RF18	O sistema deve ser capaz de identificar as vagas disponíveis.	Alta	-
RF19	O sistema deve ser capaz de identificar modificações de vagas.	Alta	RF13
RF20	O sistema deve ser capaz de alterar o status das vagas	Alta	RF19, RF16, RF14
RF21	O sistema deve enviar notificações aos coordenadores e funcionários sobre a alteração das vagas.	Alta	RF11, RF18, RF19, RF20
RF22	O sistema deve permitir que coordenadores e funcionários visualizem as notificações.	Alta	RF02, RF21
RF23	O sistema deve permitir que todos os usuários consigam sair da conta.	Alta	RF02

Fonte: Os autores.

## 5.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais são as definições das características gerais de como um sistema deve operar. Isso significa que os requisitos não funcionais de um sistema não estão

ligados diretamente com as funcionalidades específicas que o sistema irá exercer, mas sim em garantir que o sistema execute suas funções corretamente e atenda aos padrões de segurança, usabilidade e desempenho.

Tratados geralmente como restrições e premissas técnicas de um projeto, os requisitos não funcionais são praticamente todas as necessidades que não podem ser atendidas através de funcionalidades. Um exemplo de um requisito não funcional citado por Sommerville (Engenharia de Software, 2018, p.93) é: “O sistema deve ser fácil de usar pela equipe médica e ser organizado de tal modo que os erros de usuário sejam minimizados.” Este exemplo pode descrever a premissa de um stakeholder que visa minimizar a taxa de erros cometidos pelos usuários.

Normalmente, podemos mensurar os requisitos funcionais. Por exemplo:

O usuário terá que efetuar login a cada dois dias. Nenhum usuário poderá passar mais de 16 horas seguidas com o login ativo.

Nesse exemplo, limitamos o tempo de uso do software mensurando as restrições de uso (login a cada dois dias e 16 horas de uso constante), com o fim de aumentar a segurança da aplicação. O requisito citado acima pode não ser mencionado explicitamente pelos gestores e stakeholders, mas todos querem um software seguro e confiável. Por isso, nem sempre os requisitos não funcionais estão ligados a funcionalidades específicas.

Conforme descrito por Sommerville (2018), os requisitos não funcionais têm de ser muito bem definidos para garantir que o sistema cumpra com as suas funcionalidades sem intervir na regra de negócio.

- Proteção de Dados Pessoais: Garantir que dados sensíveis dos voluntários, como endereço e informações de contato, sejam armazenados e transmitidos de forma segura.

- Fraudes: Todos os cadastros devem ser revisados para evitar fraudes.

- Interface Intuitiva: O sistema deve ser fácil de usar, com fluxos de navegação claros e acessíveis para pessoas com diferentes níveis de habilidade.

- Compatibilidade Multidispositivos: Garantir o funcionamento em navegadores modernos e dispositivos móveis (responsividade).

- Código Modular: O código deve ser organizado de forma que seja fácil adicionar novas funcionalidades ou corrigir defeitos.

- LGPD/GDPR: O sistema deve estar em conformidade com as regulamentações de proteção de dados pessoais, como a Lei Geral de Proteção de Dados (LGPD) no Brasil e o Regulamento Geral de Proteção de Dados (GDPR) na Europa.

Além disso, o sistema deve atender às seguintes características gerais:

- 1- Escalabilidade: Capacidade de lidar com um aumento no número de usuários e volume de dados sem afetar o desempenho.
- 2- Disponibilidade: O sistema deve estar disponível em horários pré-definidos, com um tempo de atividade determinado pelo gestor.
- 3- Segurança: Implementação de criptografia de ponta a ponta para proteger imagens e dados, além de autenticação de dois fatores para garantir integridade e confidencialidade.
- 4- Confiabilidade: O sistema deve atingir pelo menos 90% de precisão em suas funcionalidades principais.
- 5- Manutenibilidade: O código-fonte deve ser bem documentado e modularizado para facilitar atualizações e manutenções.
- 6- Eficiência de Recursos: O sistema deve minimizar o consumo de recursos computacionais e largura de banda para garantir desempenho eficiente.

Esses requisitos asseguram que o sistema seja robusto, confiável e em conformidade com os padrões legais e técnicos exigidos.

### 5.3 QUALIDADE DE SOFTWARE

A aplicação das normas ISO no desenvolvimento de software é amplamente reconhecida como uma prática fundamental para garantir qualidade, segurança e eficiência em projetos tecnológicos. De acordo com NBR ISO/IEC 25010, a qualidade do software deve ser avaliada com base em critérios como funcionalidade, usabilidade e segurança, enquanto a NBR ISO/IEC 27001 destaca a necessidade de gerenciar riscos e proteger dados sensíveis por meio de controles de segurança. Já a ISO 9001 reforça a importância de processos bem definidos e de uma abordagem centrada no cliente, garantindo que o produto atenda às expectativas e promova melhorias contínuas. Essas normas, aplicadas de forma integrada, oferecem uma base sólida para o desenvolvimento de soluções confiáveis e alinhadas às necessidades do público-alvo, especialmente em projetos com impacto social, como o sistema de cadastro de vagas de voluntariado em ONGs.

#### 5.3.1 ISO/IEC 25010

Segundo Silva (s. d.), a ISO/IEC 25010 é um padrão internacional que estabelece padrões de qualidade de software, garantindo confiabilidade, funcionalidade e conformidade com os requisitos do produto e do usuário. Ao desenvolver um sistema web para cadastramento



de vagas de voluntariado em ONGs, a norma é uma diretriz técnica que garante a funcionalidade, segurança e acessibilidade do software, promove impacto positivo e atende às reais necessidades da organização.

Na norma ISO/IEC 25010, o desempenho é excelente, garantindo que o sistema atenda aos seus requisitos, como registros postais claros e busca independente por voluntários para garantir a precisão das informações. A usabilidade é igualmente importante, sendo dada prioridade ao fornecimento de uma interface amigável e de fácil acesso para ONGs e voluntários. A norma também enfatiza a importância da acessibilidade e garante que os sistemas cumpram as Diretrizes de Acessibilidade da Web (WCAG) para melhorar a acessibilidade para usuários com deficiência.

De acordo com Silva (s. d.), a confiabilidade é outro aspecto importante, garantindo desempenho consistente do sistema e gerenciamento adequado de erros sem comprometer a experiência do usuário ou a integridade dos dados. O desempenho é necessário para o processamento eficiente de informações, o que requer o tratamento simultâneo de um grande número de usuários. De acordo com a Lei Geral de Proteção de Dados (LGPD), ênfase especial é dada à segurança e à necessidade de proteção dos dados pessoais.

Conforme a Microsoft (s. d.), além disso, as diretrizes de padrões facilitam a distribuição de software, garantindo que ele seja modular, facilmente atualizável e compatível com diferentes dispositivos e navegadores. Em relação à qualidade de uso, a ISO/IEC 25010 enfatiza a satisfação do usuário, levando em consideração o contexto de uso, e garantindo que o sistema possa compreender e funcionar em diferentes situações, incluindo ambientes de baixo nível

O desenvolvimento de aplicações Web é baseado nas melhores práticas, desde a coleta de requisitos até a entrega final, usando os princípios da ISO/IEC 25010. Além de fornecer documentação técnica e guias do usuário, isso inclui testes contínuos para garantir desempenho, funcionalidade e segurança. A norma, portanto, garante que o programa forneça soluções eficazes e acessíveis que atendam às necessidades de ONGs e voluntários e promova um impacto social significativo através de uma comunicação eficaz entre organizações e pessoas que desejam contribuir para causas sociais.

### **5.3.2 ISO/IEC 27001**

De acordo com a IBM, a ISO/IEC 27001 é um padrão aceito internacionalmente que define os requisitos para Sistema de Gerenciamento de Segurança da Informação (SGSI). Sua aplicação é importante para projetos de trabalho onde são processadas informações sensíveis,

como um site de registro de voluntários para organizações não governamentais. Garantir confidencialidade, integridade e disponibilidade.

Há muitas vantagens em usar a ISO/IEC 27001 neste projeto. Confidencialidade, garante que apenas pessoal autorizado possa acessar os arquivos do sistema e protege a confidencialidade de informações sensíveis, como informações de contato e documentos de voluntários e ONGs. A Integridade garante que os dados registrados no sistema não serão alterados e que os dados serão armazenados. A Disponibilidade garante que os sistemas possam ser acessados e operados quando necessário, minimizando interrupções que possam causar interrupções na organização e no trabalho voluntário.

Para atender aos requisitos da ISO/IEC 27001, o projeto deve incluir controles de segurança, como acesso para proteger dados em trânsito e em repouso, ter autenticação forte para garantir que o acesso não seja restrito e identificar e mitigar riscos contínuos. Também é importante realizar uma avaliação de risco para identificar ameaças ao seu corpo e tomar as precauções adequadas.

Outro aspecto importante é o cumprimento da LGPD, que exige que os dados pessoais sejam coletados, armazenados e utilizados de forma justa e transparente. A ISO/IEC 27001 atende a esses requisitos, fornecendo uma base sólida para gerenciar e proteger essas informações, aumentando assim a confiança do usuário no sistema.

Esta norma também enfatiza a importância de um processo bem documentado e auditado para garantir que as práticas de segurança sejam consistentes e possam ser continuamente revisadas e melhoradas. Isto inclui o estabelecimento de políticas claras, a formação de equipas envolvidas no desenvolvimento e nas operações e a realização de auditorias regulares para verificar a eficácia das medidas de segurança.

Aplicando a ISO/IEC 27001 para criar um sistema de rede para uma ONG, o projeto não apenas atende aos requisitos regulatórios, mas também apoia seu compromisso com a proteção de dados e segurança da informação. Isto aumenta a confiança do utilizador, contribui para a estabilidade das soluções e garante que o sistema funciona bem para conectar organizações e voluntários, melhorando assim as relações de trabalho de reparação.

### **5.3.3 ISO/IEC 9001**

A ISO 9001 é uma norma internacional dedicada à gestão da qualidade, estabelecendo critérios para garantir que produtos e serviços atendam consistentemente às expectativas dos clientes e aos requisitos regulatórios. No desenvolvimento de um sistema web para cadastro de

vagas de trabalho voluntário em ONGs, a aplicação dessa norma é crucial para assegurar que o projeto seja conduzido de maneira organizada, eficiente e orientada a resultados de alta qualidade.

A ISO 9001 contribui significativamente ao estruturar processos claros e documentados, abrangendo desde o levantamento de requisitos das ONGs e voluntários até o desenvolvimento, testes, implantação e manutenção do sistema. Um dos pilares da norma é a abordagem baseada em processos, que auxilia na identificação e gestão das etapas críticas do projeto, garantindo a entrega de uma solução que atenda às necessidades das ONGs e proporcione uma experiência positiva aos voluntários.

Outro princípio central da ISO 9001 é o foco no cliente. Neste projeto, isso implica em entender profundamente os desafios enfrentados pelas ONGs, como a dificuldade de gerenciar e divulgar vagas de voluntariado, e desenvolver um sistema que solucione esses problemas de maneira prática e acessível. O software deve ser projetado para facilitar o uso por voluntários, promovendo a satisfação de todos os usuários envolvidos.

A norma também enfatiza a gestão baseada em evidências, aplicável ao uso de dados coletados previamente, como respostas de formulários que evidenciam a falta de ferramentas especializadas para ONGs. Esses dados devem orientar as decisões de design, funcionalidades e priorização de requisitos do sistema.

Outro aspecto essencial da ISO 9001 é a melhoria contínua, que assegura que o sistema evolua para atender novas demandas. Isso inclui implementar mecanismos para coletar feedback dos usuários, monitorar o desempenho do sistema e identificar oportunidades de aprimoramento. Um sistema web bem-sucedido deve resolver problemas iniciais e se adaptar ao crescimento das ONGs e às mudanças tecnológicas.

A norma também promove a liderança e o engajamento de pessoas, essenciais para o sucesso do projeto. Desde a equipe de desenvolvimento até os parceiros e stakeholders das ONGs, todos devem estar alinhados aos objetivos do sistema e comprometidos com a entrega de uma solução de qualidade.

Ao adotar a ISO 9001 no desenvolvimento do sistema web para ONGs, o projeto adota uma abordagem estruturada e consistente, focada na entrega de valor aos usuários e na manutenção da qualidade ao longo do tempo. Isso contribui para criar um software confiável, eficiente e capaz de gerar impacto positivo no gerenciamento de vagas de voluntariado, fortalecendo as operações das ONGs e promovendo maior engajamento social.

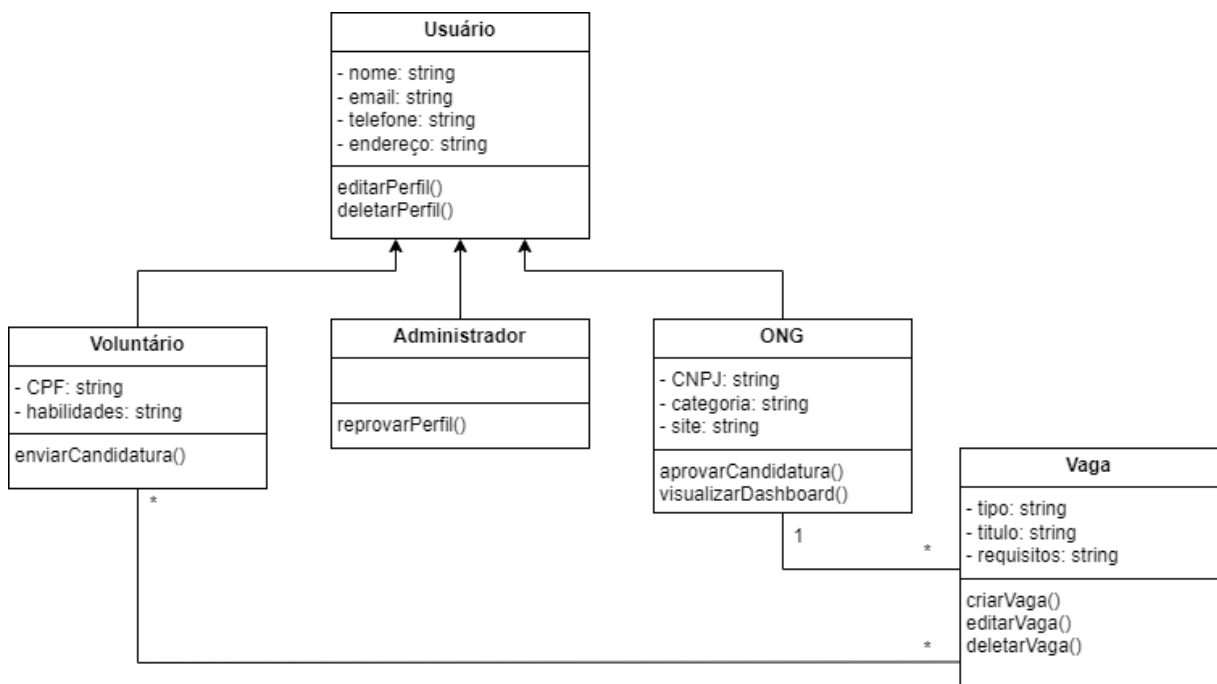
## 5.4 DIAGRAMA DE CLASSES

Segundo Sommerville (2018), o diagrama de classes é um modelo que mostra como um projeto é organizado utilizando as classes e objetos de um sistema e suas associações ou relacionamentos lógicos, como herança, generalização e agregação. Uma classe é a definição de objetos do sistema e os relacionamentos são os vínculos que esses objetos têm entre si.

Um diagrama de classes pode ser construído utilizando caixas para representar classes, informando o nome da classe no topo e abaixo os atributos indicando as características das classes, enquanto os relacionamentos são representados por uma linha entre as classes, a Figura 02 apresenta o diagrama de classes proposto para este projeto. Na extremidade de cada associação é indicado o número de objetos envolvidos no relacionamento, por exemplo, a classe Voluntário tem a relação  $^{*}:^{*}$  com a classe Vaga, ou seja, um voluntário pode se candidatar a várias vagas e uma vaga pode ter inúmeros voluntários, o asterisco representa uma quantidade indefinida de objetos envolvidos na associação.

Neste diagrama (Figura 02) são utilizadas outras multiplicidades possíveis, por exemplo, a relação entre ONG e Vaga é  $1:^{*}$ , ou seja, uma ONG pode ter inúmeras vagas cadastradas, enquanto uma vaga pode estar associada apenas a uma ONG.

Figura 02 - Diagrama de Classes



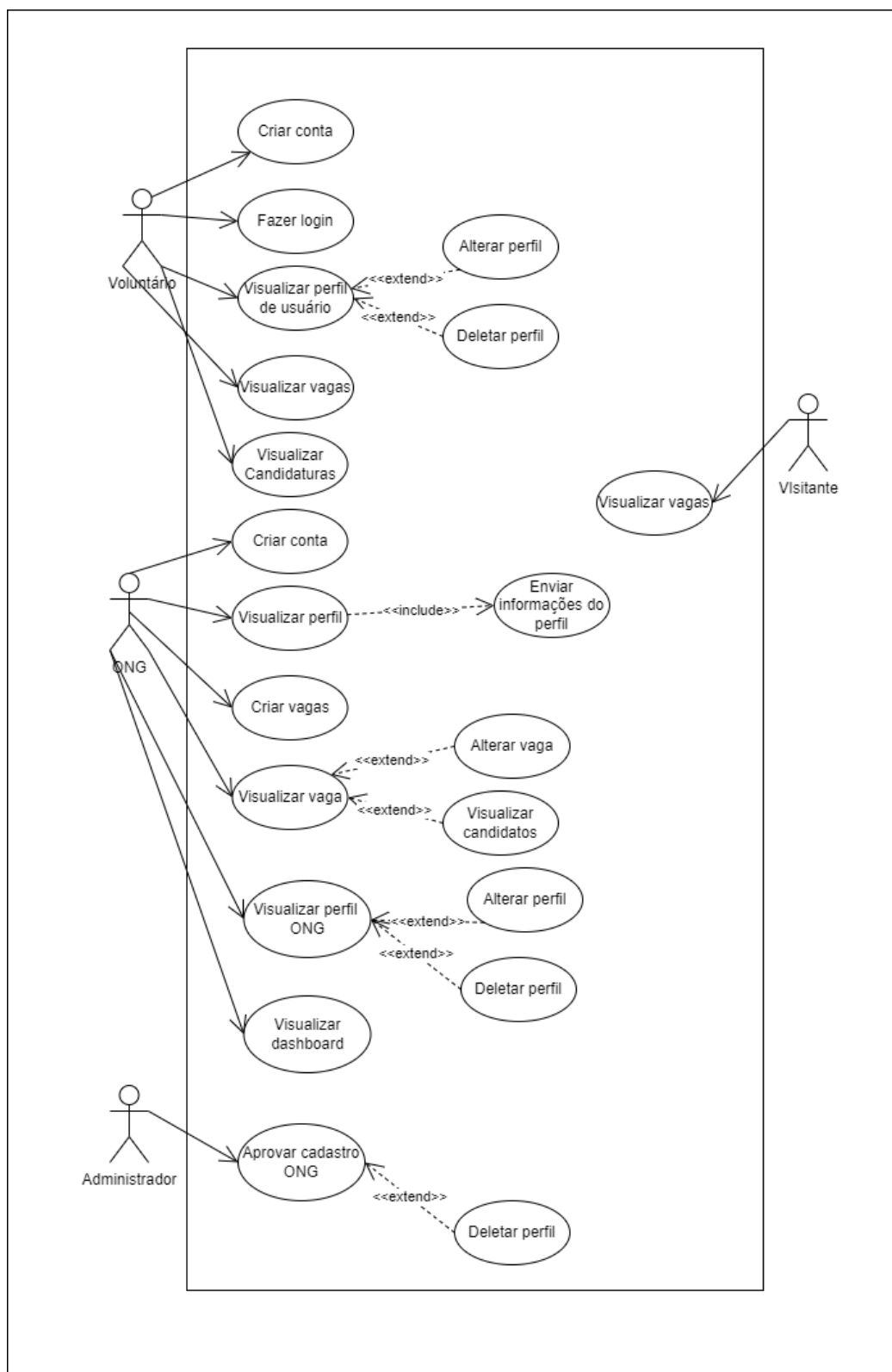
Fonte: Os autores

## 5.5 DIAGRAMA DE CASOS DE USO

De acordo com Ribeiro (2012), o diagrama de casos de uso aborda as principais funcionalidades do sistema e a interação com os usuários usando um modelo gráfico. É composto por atores, representando os tipos de usuários, que são: voluntário, ONG e administrador para a validação de ONGs. O ator é relacionado aos casos de uso, que são funcionalidades do sistema, sendo assim, é possível saber as funções que podem ser exercidas por cada tipo de usuário na aplicação.

Segundo Sommerville (2018), os casos de uso são documentados utilizando diagramas de alto nível representando todas as interações possíveis listadas nos requisitos do sistema. Os atores podem ser humanos ou outros sistemas, os quais são representados por bonecos de palitos e cada interação é representada por uma elipse, algumas interações podem incluir de forma obrigatória outra interação, este caso pode ser chamado de *include*, por outro lado também pode incluir de forma opcional outra interação, o que pode ser chamado de *extend*. Para este projeto foram desenvolvidos os casos de uso da Figura 03.

Figura 03 - Casos de Uso



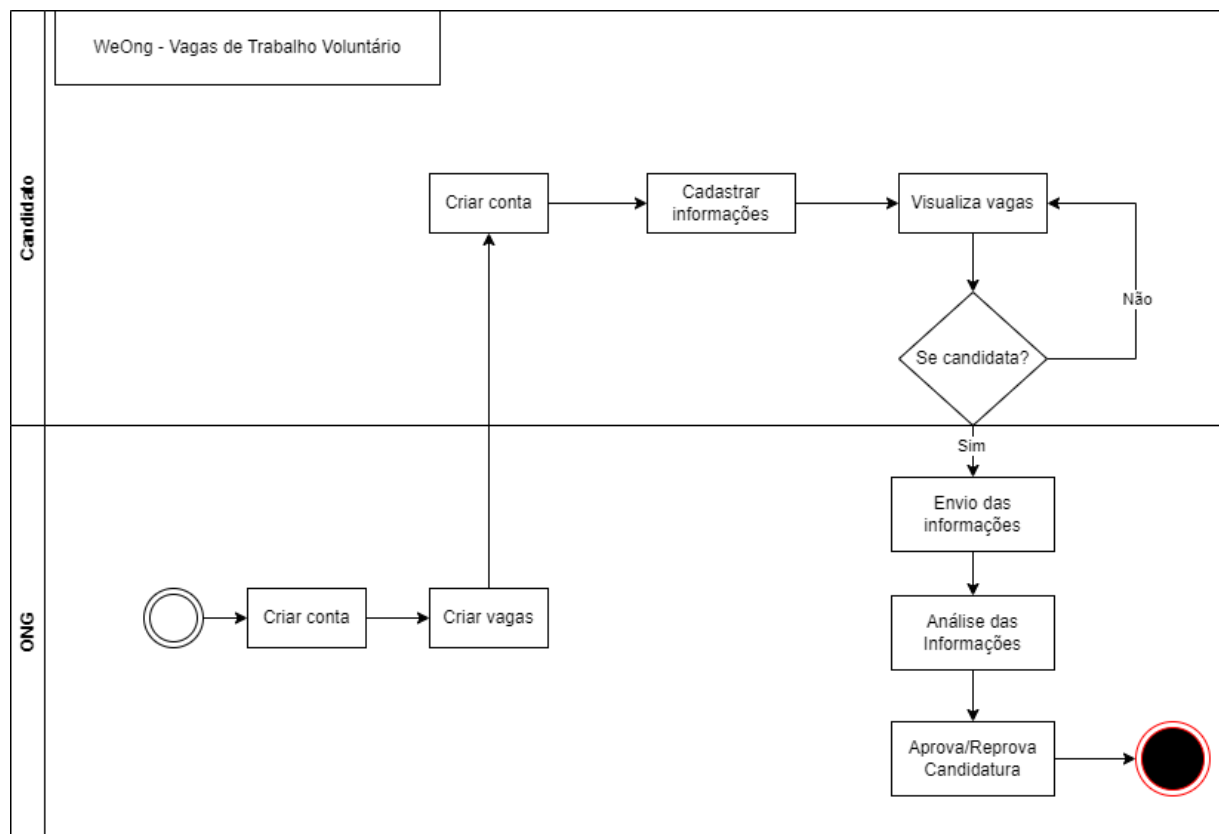
Fonte: Os autores.

## 5.6 DIAGRAMA DE ATIVIDADES

Segundo Murta (s.d.), o diagrama de atividades elabora processos de negócio, fluxos de controle, casos de uso e passos de execução de uma rotina envolvendo objetos e estados do objeto. As atividades são ações de comunicação ou manipulação de objetos sendo representadas por retângulos de canto arredondado, enquanto os fluxos de controle são representados por linhas. O início do processo é representado por um círculo com borda vazia, fim representado por um círculo com borda preenchida e uma decisão é representada por um losango determinado qual caminho o fluxo deve seguir.

Para este projeto foi desenvolvido um diagrama de atividades (Figura 04) apresentando o principal objetivo do sistema proposto, uma plataforma para cadastro de vagas de trabalho voluntário e a possibilidade de voluntários se inscreverem para as respectivas vagas.

Figura 04 - Diagrama de Atividades



Fonte: Os autores.

## 5.7 DIAGRAMA ENTIDADE RELACIONAMENTO

De acordo com Oliveira (2023), o Diagrama Entidade Relacionamento é a representação visual do banco de dados e do esquema proposto em um Modelo Entidade Relacionamento, que por sua vez, representa as entidades que fazem parte de um projeto, seus atributos e seus relacionamentos.

Conforme mostrado na figura 05, o diagrama entidade relacionamento possui x entidades, cada uma disposta com seu número de relacionamentos, como por exemplo a entidade Candidaturas, ligada a entidade Voluntarios numa ligação 1 para muitos, ou seja, um mesmo voluntário pode realizar várias candidaturas, sem necessidade de esperar a aprovação ou reprovação em uma vaga.

Figura 05 - Diagrama Entidade Relacionamento





## 6 DESENVOLVIMENTO

O sistema proposto é uma aplicação Web desenvolvida utilizando tecnologias de *front-end*, *back-end* e armazenamento de dados, as quais estão listadas nos subcapítulos seguintes. O Git e GitHub foram utilizados como ferramentas de versionamento de código, que além de serem úteis em trabalhos que envolvem uma equipe também permite disponibilizar o código do projeto online, que pode ser encontrado no repositório weong disponível na URL <<https://github.com/kayanerocha/weong>>.

Para cada nova atualização do código, os colaboradores do projeto seguiram o procedimento de criar uma nova *branch* e após o desenvolvimento e teste realizaram a junção do novo código com o código principal, o que fica na *branch main*. Após a conclusão da primeira versão do sistema, documentada no Apêndice A, foi criada a *branch develop* para então continuar o desenvolvimento de novas funcionalidades e melhorias da versão um e após a conclusão do desenvolvimento foi realizada a junção das duas versões, sendo a que estava na *main* e a que estava na *develop*.

### 6.1 FORMULÁRIO DE PESQUISA

A pesquisa realizada por meio de formulário online Google Forms tem como propósito investigar e entender o interesse e a disponibilidade das pessoas para contribuir com as organizações não governamentais (ONGs). As ONGs acabam desempenhando um papel fundamental na promoção de causas ambientais, sociais e humanitárias. Compreender os fatores que motivam a participação, as áreas de atuação preferidas e os desafios enfrentados pelos voluntários permite aprimorar estratégias de recrutamento e retenção, garantindo maior eficiência nas ações dessas entidades. Além disso, a análise das respostas possibilita o desenvolvimento de iniciativas mais alinhadas às expectativas dos colaboradores, promovendo um impacto social mais significativo.

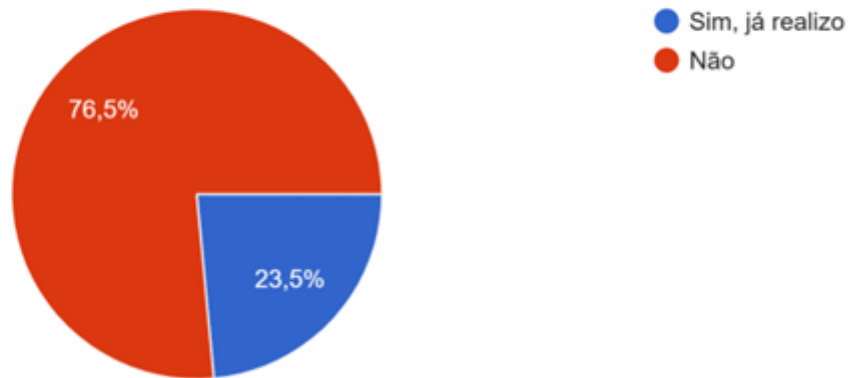
Os dados coletados por meio deste estudo serão essenciais para a formulação de políticas e práticas que facilitem o envolvimento de voluntários nas ONGs, fortalecendo sua atuação e ampliando a rede de apoio às comunidades atendidas. A pesquisa também busca identificar potenciais barreiras que dificultam a adesão ao voluntariado, proporcionando insights valiosos para a criação de programas mais acessíveis e inclusivos.

Conforme mostrado na figura 06, dentre os entrevistados, 76% nunca realizaram algum trabalho voluntário em sua vida.

Figura 06 - Realização de Trabalho Voluntário.

Você já realiza trabalho voluntário?

51 respostas



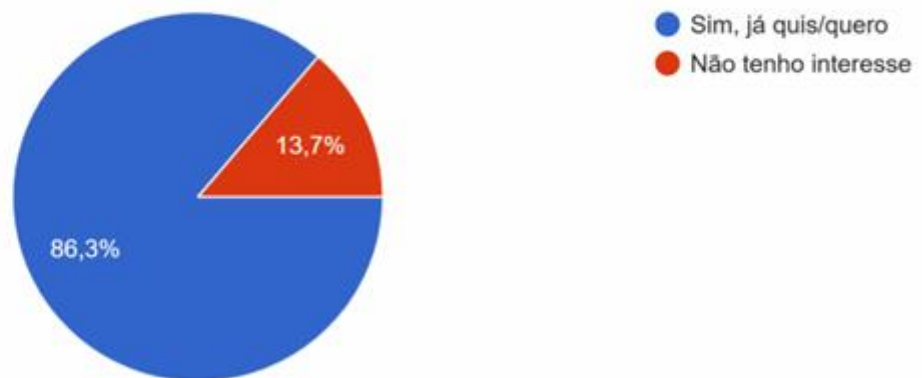
Fonte: Os Autores.

Conforme apresentado na figura 07, é demonstrado que cerca 86% dos entrevistados já possuíam interesse em trabalhar em alguma ONG, mas por diversos motivos nunca realizaram o trabalho voluntário.

Figura 07 - Desejo de Realização de Trabalho Voluntário

Se não realiza, já desejou realizar trabalho voluntário?

51 respostas

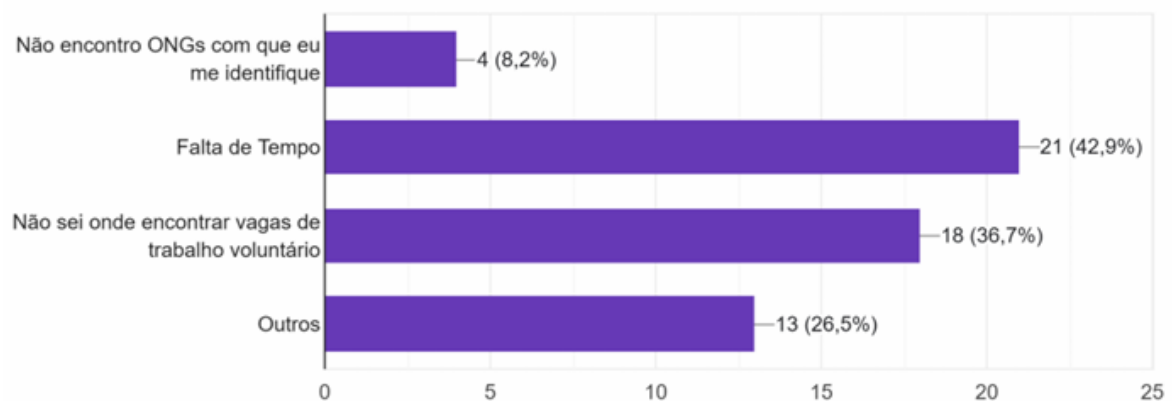


Fonte: Os autores.

Segundo a figura 08, entre os motivos mais comuns estão: a falta de tempo e/ou a dificuldade de encontrar vagas de trabalhos voluntários em ONGs.

Figura 08 – Empecilhos Encontrados

Se deseja realizar, qual o maior empecilho encontrado? Selecione as opções que se aplicam.  
49 respostas

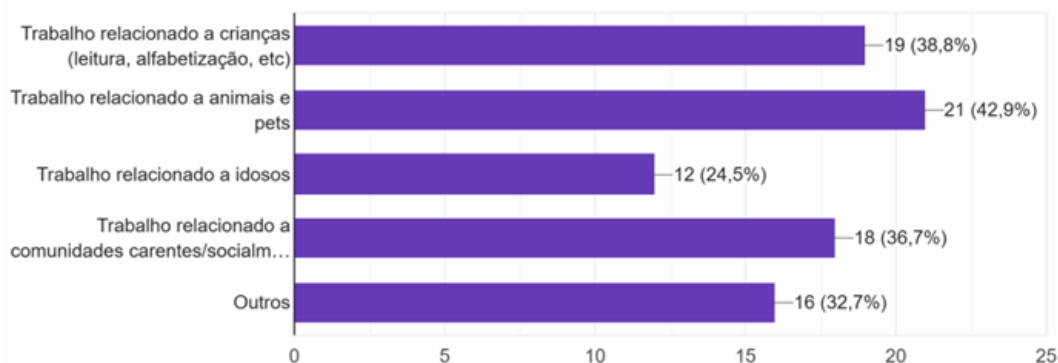


Fonte: Os autores.

Com a pesquisa realizada, também foi possível mensurar qual tipo/categoria de trabalho voluntário os entrevistados possuem mais interesse de exercer, sendo a mais escolhida o trabalho relacionado a animais e pets, e logo em seguida trabalhos relacionados a crianças e adolescentes, conforme demonstrado na figura 09.

Figura 09 – Tipos de Trabalho Voluntário

Qual tipo de trabalho voluntário você realiza ou gostaria de realizar? Selecione as opções que se aplicam.  
49 respostas



Fonte: Os autores.

## 6.2 DESENVOLVIMENTO DA INTERFACE DO USUÁRIO - FRONT-END

Para o desenvolvimento da interface, foi utilizado majoritariamente o framework Bootstrap, conforme citado anteriormente. O desenvolvimento se deu por meio do planejamento das telas exibidas ao usuário, como a tela inicial, telas de login e cadastro, e a aplicação do projeto utilizando o framework como base, além de códigos em HTML e CSS para a estruturação das informações.

Foi escolhida uma paleta de cores em tons de azul e logomarca da instituição, denominada “WeONG”, solidificando a ideia de uma empresa focada no gerenciamento de vagas de trabalho voluntário em organizações não governamentais (ONGs). A escolha da logomarca, demonstrada na Figura 10, deu ênfase ao acolhimento e amparo geralmente oferecidos pelas ONGs à população brasileira.

Figura 10 - Logomarca WeONG



Fonte: Os autores.

Ao entrar no site, o usuário encontra as vagas já cadastradas na plataforma, juntamente com um aviso no canto superior esquerdo indicando que o usuário não está logado nem como ONG ou como voluntário, conforme indicado na figura 11. A candidatura é barrada para os candidatos que não estão logados na plataforma, indicando a tela de login como sugestão de ação. Porém, mesmo sem ter realizado o login, o usuário consegue visualizar os detalhes da vaga selecionada, informações sobre qual ONG cadastrou a vaga e o título da vaga, bem como seus requisitos e endereço, no caso de vaga presencial.

Figura 11 - Tela inicial



Fonte: Os autores

Após o login do voluntário, é possível se candidatar em novas vagas, visualizar seus detalhes e acessar as informações de cadastro do usuário, conforme figura 12. O usuário tem como possibilidade a edição de seu cadastro e alteração das informações relacionadas ao seu perfil. Por questões de privacidade de dados, foram ocultados o CPF e o número de telefone do perfil do candidato utilizado como exemplo, indicado na figura 12. Simultaneamente, as ONGs cadastradas também conseguem visualizar e editar suas informações, conforme demonstrado na figura 13.

Figura 12 - Perfil do Candidato

The screenshot shows the 'Perfil' page. The header includes the WeOng logo and navigation links: Home, Mapa de ONGs, Alcance do Projeto, Minhas Candidaturas, and Logout. The main heading is 'Perfil'. Below it, a sub-header reads 'Informações Editáveis'. The form contains the following fields:

- CPF: 4637 [redacted]
- Data de nascimento: March 3, 2002
- Status: Ativo
- Nome Completo: Ana
- Usuário: ana
- E-mail: ana@gmail.com
- Telefone: 11940 [redacted]
- CEP: 09910550
- Logradouro: Avenida Senador Vitorino Freire
- Bairro: Centro
- Cidade: Diadema
- Estado: São Paulo

Fonte: Os autores.

Figura 13 – Perfil da ONG

**Perfil**

ONG	
CNPJ: 05108918000172 - <a href="#">Revalidar CNPJ</a>	Situação cadastral do CNPJ: ATIVA
Nome da ONG: AMIGOS DO BEM	Razão social: AMIGOS DO BEM INSTITUICAO NACIONAL CONTRA A FOME E A MISERIA
Endereço: RUA DOUTOR GABRIEL DE RESENDE, 122, - VILA INVERNADA, SAO PAULO - SP, 03350005	Status: Ativa

Informações Editáveis	
Usuário amigosdobem	E-mail informacoes@amigosdobem.org
Telefone 1130190107	Site https://www.amigosdobem.org/

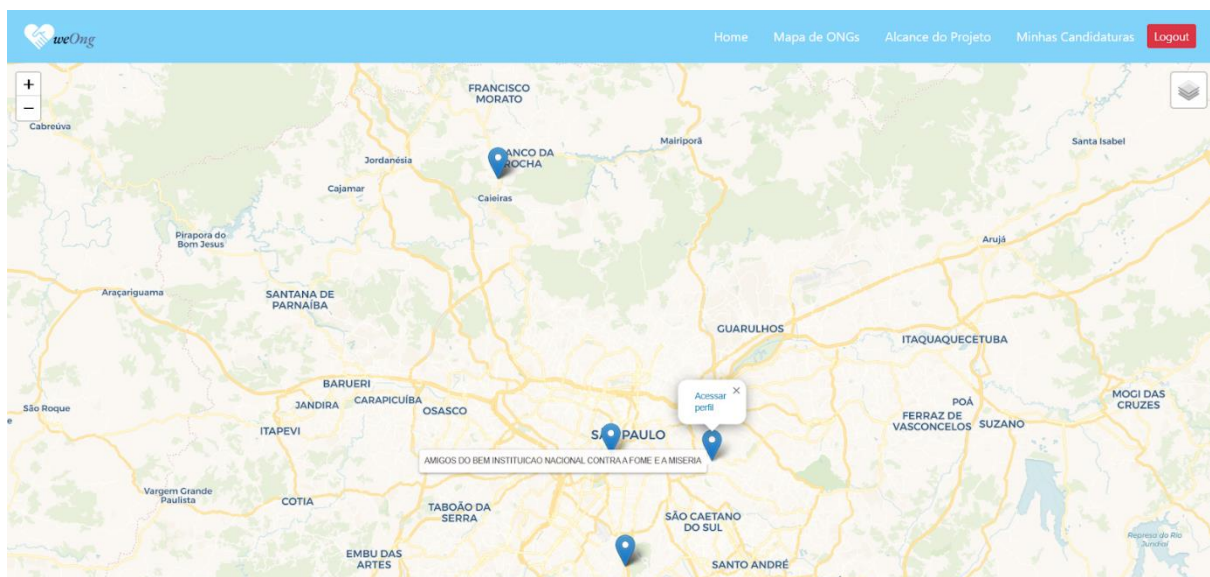
[Salvar Alterações](#) [Cancelar](#)

Fonte: Os autores

Um destaque importante para a interface do sistema é o mapa de ONGs, o qual disponibiliza para o usuário ONGs que da cidade que ele se encontra. O *back-end* envia o mapa pronto em HTML para o *template* e com os recursos do Django o mapa é carregado tendo o cuidado de não imprimir o HTML como texto, mas sim como as *tags* HTML devem funcionar na página.

O mapa possui três estilos de visualização: OpenStreetMap, visual clássico e aberto, CartoDB Positron, que é um mapa claro e minimalista, e CartoDB Voyager, estilo intermediário com mais detalhes urbanos. Esses estilos são disponibilizados pela biblioteca Folium e já são enviados pelo *back-end*. Além de ser possível visualizar os marcadores com a localização das ONGs, também é possível ver o nome da ONG ao passar o mouse em cima do marcador, e ao clicar é possível acessar um link que redireciona o usuário para o perfil da ONG, conforme Figura 14.

Figura 14 - Mapa de ONGs

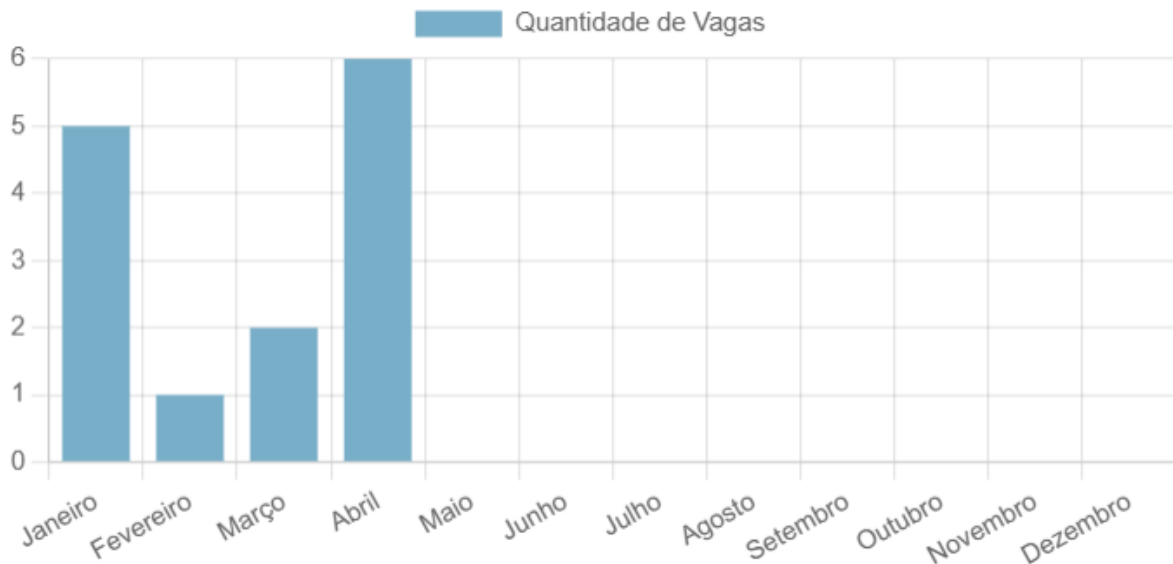


Fonte: Os autores.

Outro destaque desse sistema é a disponibilização de informações sobre o alcance do projeto através de gráficos interativos. Os gráficos foram criados utilizando a biblioteca Chart.js, que ao receber os dados enviados do *back-end* cria diferentes tipos de *charts*. Os dados são enviados através de requisições HTTP GET, que são disparadas para cada informação disponível, sempre que o usuário escolher um ano para obter as informações ou quando carrega a página e retorna dados do ano atual automaticamente.

O gráfico em barras chamado Quantidade de Vagas, Figura 15, divulga a quantidade de vagas criadas por mês no ano que está sendo solicitado, permitindo acompanhar a evolução da oferta de oportunidades voluntárias ao longo do tempo identificando possíveis sazonalidades, picos de demanda ou períodos com baixa criação de vagas.

Figura 15 - Quantidade de Vagas

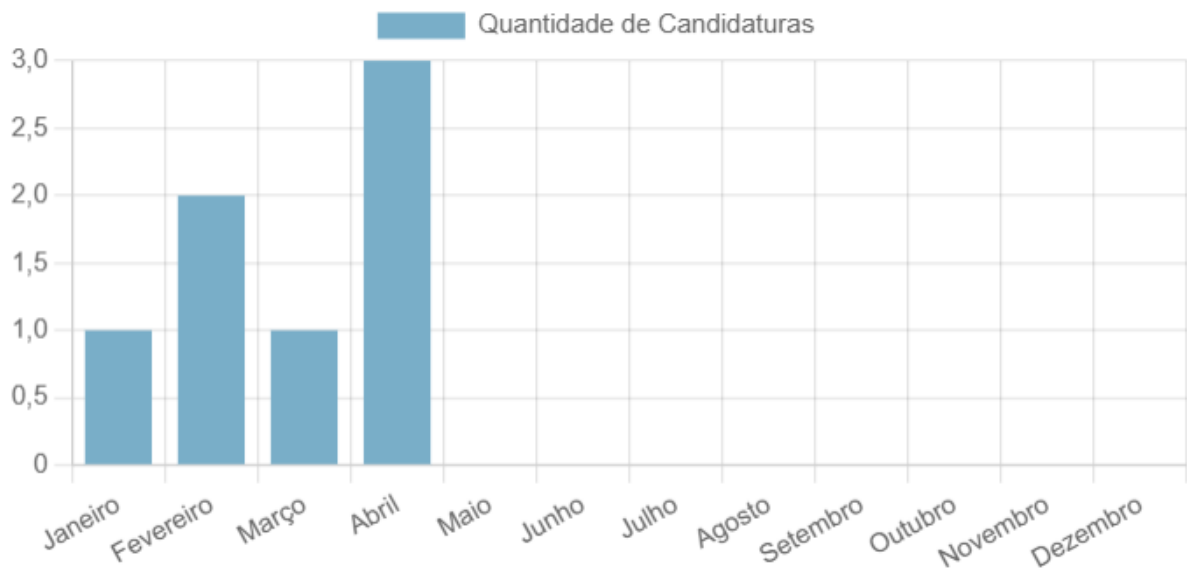


Fonte: Os Autores

O gráfico em barras de quantidade de candidaturas, Figura 16, apresenta a quantidade de candidaturas realizadas por mês, permitindo avaliar o engajamento dos voluntários ao longo do tempo, identificando os meses com maior participação, além de ajudar a verificar se campanhas de divulgação estão surtindo efeito, contribuindo para o planejamento estratégico de ações futuras.

Figura 16 - Quantidade de Candidaturas

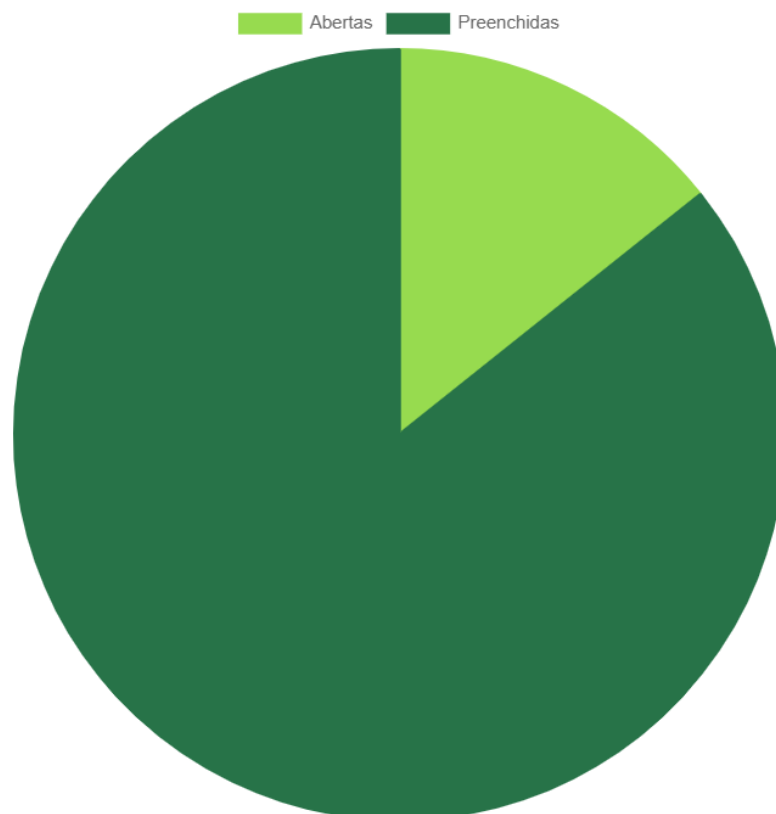




Fonte: Os Autores.

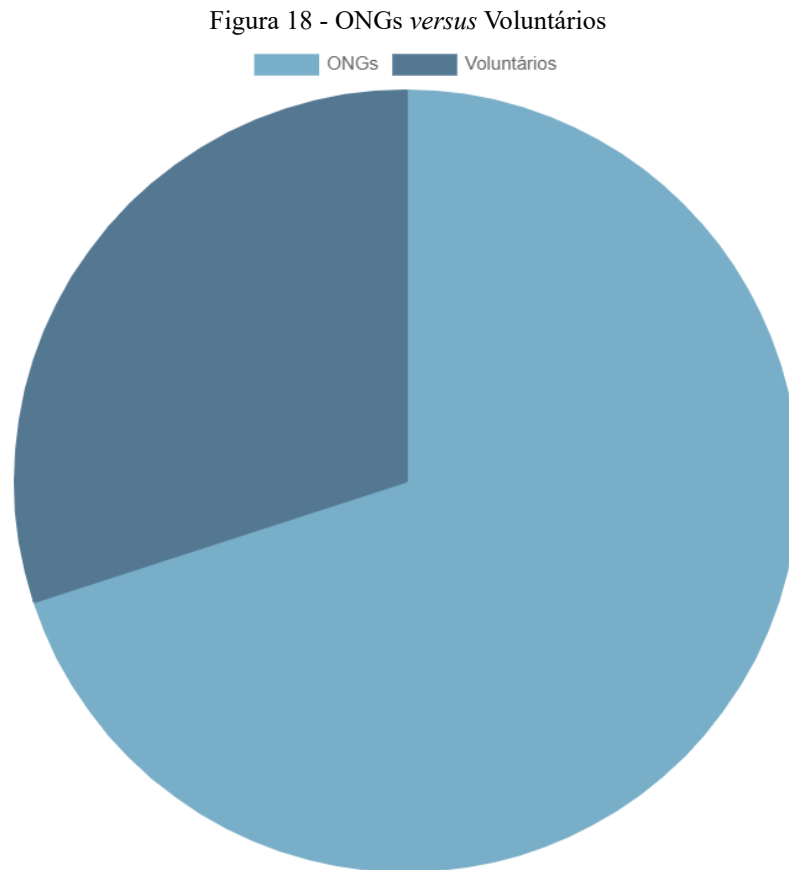
O gráfico de pizza que apresenta número de vagas abertas e número de vagas preenchidas, conforme Figura 17, permite visualizar a eficiência do sistema de recrutamento. Um número elevado de vagas abertas em relação às preenchidas pode indicar dificuldades no preenchimento ou baixa atratividade, enquanto o equilíbrio sugere um bom desempenho do sistema e alcance dos objetivos.

Figura 17 - Vagas Preenchidas *versus* Vagas Abertas



Fonte: Os Autores.

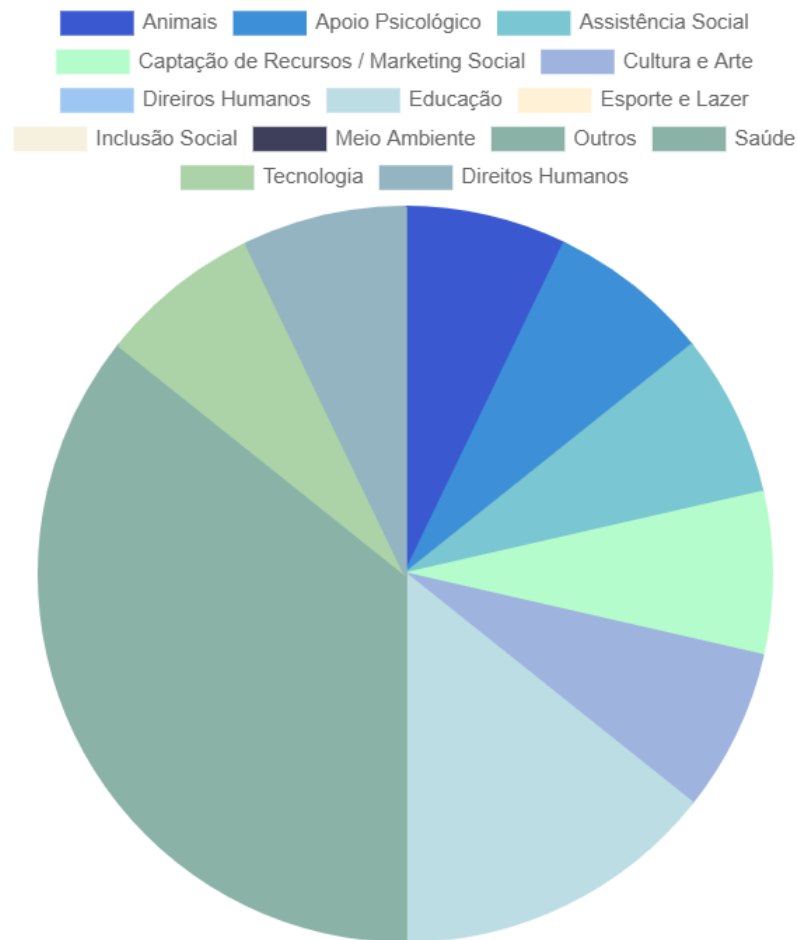
O gráfico de pizza que apresenta o número de ONGs e voluntários ativos no sistema, representado na figura 18, permite avaliar se há equilíbrio entre oferta e demanda, sendo essa métrica essencial para a sustentabilidade do ecossistema de colaboração voluntária. Essa divulgação é importante para todos os usuários, mas principalmente para os administradores do sistema terem uma ideia do que pode estar faltando ou não para atingir um equilíbrio e então tomar ações.



Fonte: Os Autores.

O gráfico de pizza de vagas por área, Figura 19, permite identificar quais áreas têm maior número de vagas disponibilizadas, revelando tendências de atuação das ONGs e permitindo que voluntários e gestores entendam melhor o perfil das demandas no sistema.

Figura 19 - Vagas por Área



Fonte: Os autores.

### 6.3 DESENVOLVIMENTO DA LÓGICA DE NEGÓCIO - BACK-END

A lógica de negócio do sistema proposto ficou centralizada no *back-end* desenvolvido com a linguagem de programação Python, o framework Django e o banco de dados MySQL, além de utilizar a API chamada Brasil API, para a validação de CNPJ das ONGs e CEPs de voluntários, bibliotecas úteis como validate-docbr, Subcapítulo 3.11 Validate-docbr, para validação de CPF, OpenCage GeoCoder, Subcapítulo 3.12 OpenCage GeoCoder, para a busca de coordenadas geográficas, Folium, Subcapítulo 3.13 Folium, para a criação de um mapa de ONGs.

O Django disponibiliza quase que de forma automática um painel de administração que possibilita o gerenciamento de todos os dados pelos administradores do sistema, fornece também mecanismos de autenticação, encerramento de sessão e gerenciamento de permissões entre usuários e grupos de usuários, facilitando assim a implementação de algumas funcionalidades essenciais.

Os modelos de dados para esse projeto foram criados utilizando Django, seguindo a sua documentação, a partir deles foram geradas as migrações e então a estrutura do banco de dados. Conforme o desenvolvimento se estendia, esses modelos foram sendo alterados devido à necessidade de novos campos, refatoração dos existentes ou exclusão do que não fazia sentido, por fim gerando a estrutura final do banco de dados, conforme o Diagrama Entidade Relacionamento. Durante todo o processo de desenvolvimento o banco de dados foi sendo construído e utilizado localmente, e compartilhado entre os membros da equipe por exportações realizadas, além de ser possível criar toda a estrutura com o histórico de migrações que foram gerados.

É com o Django que as URLs do sistema Web são criadas e elas então chamam o código que deve ser executado e retornar uma resposta para o usuário conforme solicitado. Este código é desenvolvido com base nas regras de negócio e funcionalidades propostas para esse sistema, realizando validações dos dados, consumindo bibliotecas e APIs externas gratuitas, buscando, salvando, atualizando e excluindo dados do banco de dados MySQL.

O *back-end* do sistema consome gratuitamente a Brasil API para validar o CNPJ das ONGs, conseguindo obter o nome fantasia, razão social, situação cadastral do CNPJ e endereço da ONG, e para validar o endereço dos voluntários utilizando o CEP informado pelo usuário. A busca dessas informações é realizada através de requisições HTTP do tipo GET nos endereços responsáveis por cada dado. Além de validar esses dados, o *back-end* também verifica se o voluntário está inserindo um número de CPF válido através da biblioteca validate-docbr. Por fim, o sistema não permite o cadastro de usuário caso um desses dados não sejam aprovados nas validações.

A biblioteca OpenCage GeoCode é utilizada para buscar as coordenadas geográficas dos endereços das ONGs e voluntários sempre que realizam o seu cadastro no sistema. Para consumir esta biblioteca é necessário utilizar uma chave, a qual é obtida gratuitamente no painel de administração do OpenCage, e com ela o sistema realiza a busca das coordenadas.

O mapa de ONGs é criado utilizando a biblioteca Folium, a qual necessita das coordenadas geográficas dos endereços e permite inserir legendas em cada ponto de ONG que está no mapa. O sistema primeiro busca a localização do usuário utilizando a API ipwhois.io, para então criar os marcadores de ONGs da cidade em que o usuário se encontra, mas caso não encontre a localização dele, o sistema utiliza como padrão a cidade de São Paulo, o que pode ser aprimorado no futuro. Para complementar são adicionados três estilos de mapas: OpenStreetMap, CartoDB Positron e CartoDB Voyager, os quais são estilos suportados pela

biblioteca Folium. Por fim, o *back-end* retorna uma representação do mapa em HTML e o *front-end* renderiza esse mapa para o usuário.

## 6.4 TESTES E RESULTADOS

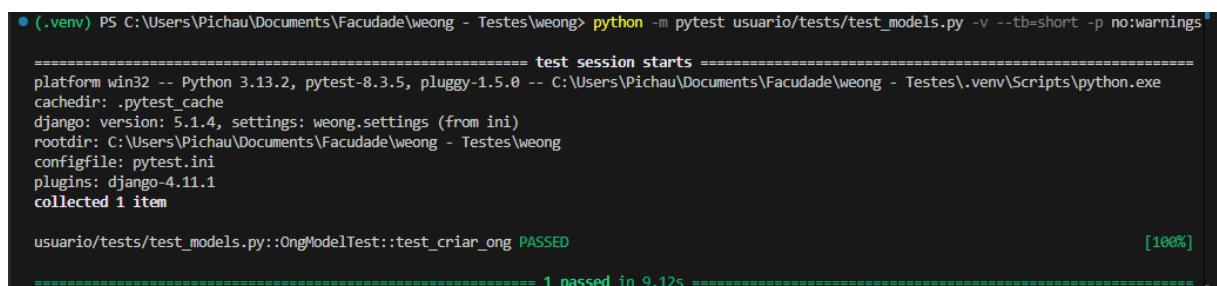
A validação da qualidade do sistema foi realizada por meio de testes automatizados e manuais, assegurando que as principais funcionalidades do projeto operem corretamente em diferentes cenários. Entre as abordagens adotadas, destacam-se os testes unitários, responsáveis por verificar o comportamento isolado de componentes essenciais, como os modelos de ONGs, voluntários e vagas de voluntariado.

A execução dos testes automatizados foi realizada utilizando o framework de testes do Django, com apoio do Pytest, que proporciona uma apresentação clara dos resultados. Abaixo, apresenta-se o comando utilizado na raiz do projeto para iniciar os testes:

```
python -m pytest usuario/tests/test_models.py -v --tb=short -p no:warnings
```

Esse procedimento gerou uma saída informativa no terminal, indicando o sucesso na execução dos testes unitários. A Figura 20 ilustra o resultado obtido, confirmando que o sistema está validado quanto ao correto armazenamento e gerenciamento dos dados no banco de dados.

Figura 20 – Resultado da Execução do Teste de Criação de ONG



```
(.venv) PS C:\Users\Pichau\Documents\Facudade\weong - Testes\weong> python -m pytest usuario/tests/test_models.py -v --tb=short -p no:warnings
===== test session starts =====
platform win32 -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0 -- C:\Users\Pichau\Documents\Facudade\weong - Testes\.venv\Scripts\python.exe
cachedir: .pytest_cache
django: version: 5.1.4, settings: weong.settings (from ini)
rootdir: C:\Users\Pichau\Documents\Facudade\weong - Testes\weong
configfile: pytest.ini
plugins: django-4.11.1
collected 1 item

usuario/tests/test_models.py::OngModelTest::test_criar_ong PASSED [100%]

===== 1 passed in 9.12s =====
```

Fonte: Os autores.

Os testes de permissão e acesso foram desenvolvidos com o objetivo de garantir a segurança e o cumprimento das regras de negócio do sistema, assegurando que apenas ONGs autenticadas possam realizar o cadastro de vagas, enquanto voluntários possuem acesso restrito a funcionalidades específicas, como a visualização e a inscrição em vagas. Esses testes reforçam a conformidade do sistema em relação aos diferentes perfis de usuários.

A execução seguiu o mesmo procedimento adotado nos testes unitários, com o seguinte comando:

```
python -m pytest usuario/tests/test_permissions.py -v --tb=short -p no:warnings
```

Esse processo permitiu validar o comportamento esperado para cada cenário de acesso. A Figura 21 apresenta o resultado obtido, demonstrando que o sistema gerencia corretamente as permissões de acordo com o perfil do usuário.

Figura 21 – Resultado da execução dos testes de permissão de usuários

```
(.venv) PS C:\Users\Pichau\Documents\Facudade\weong - Testes\weong> python -m pytest usuario/tests/test_permissions.py -v --tb=short -p no:warnings
===== test session starts =====
platform win32 -- Python 3.13.2, pytest-8.3.5, pluggy-1.5.0 -- C:\Users\Pichau\Documents\Facudade\weong - Testes\.venv\Scripts\python.exe
cachedir: .pytest_cache
django: version: 5.1.4, settings: weong.settings (from ini)
rootdir: C:\Users\Pichau\Documents\Facudade\weong - Testes\weong
configfile: pytest.ini
plugins: django-4.11.1
collected 3 items

usuario/tests/test_permissions.py::PermissionTest::test_nao_autenticado_nao_pode_criar_vaga PASSED [ 33%]
usuario/tests/test_permissions.py::PermissionTest::test_ong_pode_criar_vaga PASSED [ 66%]
usuario/tests/test_permissions.py::PermissionTest::test_voluntario_nao_pode_criar_vaga PASSED [100%]

===== 3 passed in 14.08s =====
```

Fonte: Os autores.

#### 6.4.1 Testes no Back-end

Durante o desenvolvimento do *back-end* da aplicação foi conduzido um processo rigoroso de testes manuais, com o intuito de validar não apenas o correto funcionamento dos *endpoints*, mas também a integridade das regras de negócio, o fluxo entre módulos e a consistência dos dados no banco. Inicialmente, foram realizadas interações diretas com as rotas HTTP por meio de navegadores com o intuito de simular diferentes cenários de uso. Essas simulações incluíram acessos às funcionalidades de cadastro de ONGs, registro de voluntários, criação de vagas e edição de informações, permitindo verificar as respostas do sistema em situações diversas, como a entrega correta dos códigos de *status* (HTTP 200, 302, 403) e a realização de redirecionamentos esperados. Por exemplo, ao tentar acessar a rota de cadastro de vagas sem autenticação, confirmou-se que o sistema redirecionava o usuário para a página de login, atendendo ao comportamento esperado para proteger áreas restritas.

Além dessa camada de validação dos *endpoints*, foram testados diferentes perfis de usuários em cenários variados, assegurando que as regras de negócio fossem devidamente aplicadas. Verificou-se que apenas ONGs autenticadas conseguiam acessar a funcionalidade de cadastro de vagas, enquanto os voluntários tinham acesso restrito às ações de visualização e inscrição nas vagas publicadas. Para isso, foi fundamental simular perfis variados, como ONGs

ativas, voluntários ativos e usuários não autenticados, assegurando que as permissões estivessem corretamente configuradas para cada nível de acesso.

Os testes também contemplaram a integração entre os módulos, conduzindo-se fluxos completos que envolviam o cadastro de uma ONG, o cadastro de uma vaga por essa ONG e, por fim, a visualização dessa vaga por um voluntário. Essa abordagem possibilitou validar que os módulos do sistema estavam devidamente integrados e que os dados fluíam corretamente entre as diferentes camadas da aplicação. Após a realização dessas operações, foram feitas verificações diretas no banco de dados, garantindo que as informações fossem corretamente armazenadas e refletissem as ações realizadas no sistema. Consultas às tabelas de ONGs, voluntários e vagas permitiram confirmar, por exemplo, que as vagas criadas estavam devidamente associadas às ONGs responsáveis por sua publicação.

Por fim, foram testados cenários de tratamento de erros, como a tentativa de cadastrar uma ONG com um CNPJ duplicado ou com campos obrigatórios em branco. Nessas situações, confirmou-se que o sistema respondia de maneira adequada, emitindo mensagens de erro informativas e impedindo a persistência de dados inválidos. Esse conjunto de testes manuais permitiu assegurar que o *back-end* do sistema estivesse preparado para lidar com as diferentes interações previstas, reforçando a confiabilidade das operações.

#### 6.4.2 Testes no Front-end

Inicialmente, devido à necessidade de garantir uma base visual sólida e funcional, os testes relacionados à interface do usuário foram focados no design gráfico, na apresentação dos componentes do sistema, utilizando Bootstrap, e na usabilidade geral da navegação. No entanto, considerando que o *front-end* é a camada final de interação com ONGs e voluntários, foram priorizados os testes de fluxo de interação do usuário, como os processos de cadastro, login, busca e visualização de vagas, e candidatura a oportunidades, conforme proposto durante o projeto.

Além da priorização das funcionalidades essenciais para voluntários e ONGs e do design de interface simples para melhor entendimento do usuário, foi garantido que a conexão entre os status apresentados no *front-end* (como status de vagas ou de candidaturas) e as saídas geradas no *back-end* (Django) estejam devidamente alinhadas e sincronizadas. Dessa forma, as informações exibidas em relação às vagas disponíveis, aos detalhes das ONGs ou aos perfis dos voluntários, refletem em tempo real os resultados processados pela lógica de negócio desenvolvida no *back-end* do projeto. A interface do usuário foi estruturada de maneira a ter

como principal objetivo listar de forma clara as diferentes vagas, informações de ONGs e opções de gerenciamento para os usuários, assegurando que tanto voluntários quanto ONGs possam acompanhar as informações e realizar ações necessárias facilmente.

O detalhamento completo sobre como está organizado o fluxo de informações, especialmente a comunicação entre as diferentes camadas do sistema (*Front-end* e *Back-end* Django), foi abordado na seção 7.3.1, onde foi explicado o fluxo de integração utilizando os serviços de hospedagem (Render) e banco de dados.

## 6.5 ANÁLISE FINANCEIRA

Para assegurar a entrega de uma solução digital eficaz, a equipe técnica foi montada com foco nas funções vitais para o desenvolvimento completo do sistema. Cada membro foi escolhido levando em conta as tarefas específicas, o alcance do projeto e um período de 6 meses.

Formação da Equipe:

- Profissional de Back-End (Infraestrutura)

Encarrega-se de configurar e manter a infraestrutura de servidores, assegurando a escalabilidade, o desempenho e a segurança do aplicativo.

Remuneração mensal: R\$ 5.500,00, totalizando em 6 meses: R\$ 33.000,00

- Coordenadora de Qualidade (QA)

Trabalha diretamente com testes, tanto automatizados quanto manuais, garantindo que o produto final entregue seja altamente confiável e livre de erros.

Remuneração mensal: R\$ 4.500,00, totalizando em 6 meses: R\$ 27.000,00

- Especialista em Banco de Dados

Responsável pelo design, criação e manutenção das bases de dados, além de assegurar a segurança e integridade das informações.

Remuneração mensal: R\$ 5.500,00, totalizando em 6 meses: R\$ 33.000,00

- Programadores de Back-End

Atuam na implementação da lógica de negócios da aplicação, integração com APIs, autenticação e asseguram a eficiência da camada do servidor.

Remuneração mensal (cada): R\$ 5.000,00, totalizando em 6 meses: R\$ 60.000,00



- Programadores de Front-End

Encargados de desenvolver a interface visual para o usuário, garantindo acessibilidade, responsividade e uma experiência de uso positiva.

Remuneração mensal (cada): R\$ 4.500,00, totalizando em 6 meses: R\$ 81.000,00

Custo total com a equipe técnica: R\$ 234.000,00

Infraestrutura Tecnológica (Serviços em Nuvem) – R\$ 1.200,00

Para assegurar que a plataforma funcione de maneira contínua e estável, foram considerados custos mensais com serviços em nuvem para suportar a hospedagem do aplicativo e dos bancos de dados.

Render: Serviço de hospedagem destinado ao ambiente de produção, permitindo a escalabilidade do aplicativo conforme a demanda dos usuários.

AWS RDS: Solução administrada para o banco de dados, que proporciona backup automático, segurança e alta disponibilidade.

Estimativa mensal: R\$ 200,00

Total para 6 meses: R\$ 1.200,00

Custo Operacional Total Previsto – R\$ 235.200,00

A soma dos gastos com mão de obra especializada e infraestrutura tecnológica, conforme indicado na Figura 22, representa o investimento requerido para a implementação e manutenção do projeto weOng durante os primeiros seis meses. Esses valores refletem o compromisso com a qualidade, segurança e escalabilidade do sistema, garantindo sua sustentabilidade e impacto social positivo.

Figura 22 - Consolidado financeiro

			
Cargos	Qtde	Salário Mensal (R\$)	Custo 6 meses (R)
Desenvolvedor Back-End (Infraestrutura)	1	5.500	33.000
Gestora de Qualidade (QA)	1	4.500	27.000
Desenvolvedor Banco de Dados	1	5.500	33.000
Desenvolvedores Back-End	2	5.000	60.000
Desenvolvedores Front-End	3	4.500	81.000
<b>Total Geral</b>	<b>8</b>		<b>R\$234.00</b>

Custo Operacional Total Estimado (6 meses)	
Categoria	Custo Estimado (R\$)
Infraestrutura (Render + AWS RDS)	1.200
Equipe Técnica (mão de obra)	234.000
<b>Total Geral (6 meses)</b>	<b>R\$235.200</b>

Fonte: Os autores.

## 7 TRABALHOS FUTUROS

O sistema web desenvolvido, tem como principal premissa otimizar o processo de recrutamento e gestão de voluntários para ONGs. Para isso, oferece uma interface intuitiva e amigável que permite o cadastro detalhado de vagas. A plataforma visa transformar as práticas gerenciais e de avaliação de desempenho das ONGs, tanto interna quanto externamente. Ao criar um ponto único para centralizar as oportunidades de voluntariado, o projeto alinha-se com as conformidades e requisitos de diversos órgãos reguladores, evidenciando o potencial da adoção da ciência da computação para gerar impacto positivo na sociedade.

Durante o desenvolvimento do sistema, foram realizadas diversas pesquisas tanto de benchmark com iniciativas similares que já apresentam os impactos deste tipo de projeto, quanto diretamente com o público-alvo, identificando e adaptando as melhores práticas para o contexto brasileiro. A solução foi concebida para ser prática e acessível, utilizando a tecnologia como um meio de transformação social. Entre as funcionalidades implementadas, destacam-se interfaces distintas para usuários e ONGs, a exibição detalhada de vagas, a criação de análises de inscritos nas vagas.

Muitas ONGs ainda dependem de métodos tradicionais, mecânicos e pouco eficientes para divulgar suas vagas de voluntariado, limitando seu alcance. O sistema proposto visa solucionar a dificuldade de divulgar vagas, encontrar e reter voluntários com perfis adequados, suprimindo a ausência de ferramentas eficientes no mercado brasileiro para esse processo. Os voluntários, por sua vez, preservam e fortalecem a malha social através de suas ações, e a plataforma facilitará sua conexão com oportunidades relevantes.

O refinamento da ideia proposta durante o desenvolvimento do projeto é uma tarefa essencial, visando a melhoria contínua e a aplicação coerente dos conceitos propostos durante o decorrer do projeto. Para a ampla aplicação no mercado, o projeto deve obedecer, conforme mencionado anteriormente no decorrer do texto, a Lei Geral de Proteção de Dados (LGPD), por lidar diretamente com informações pessoais dos voluntários candidatos e com as informações das ONGs que se cadastrarem no site.

Por fim, o desenvolvimento do projeto pode ser aprimorado através de novas funcionalidades relacionadas a filtragem das vagas e a indicação automática de vagas para um candidato que resida próximo de uma ONG que cadastrou uma vaga de trabalho. Simultaneamente, as ONGs poderiam receber como priorização na fila de inscritos para uma vaga, os candidatos que residam mais perto da localização indicada pela ONG no caso de vagas presenciais. Demais melhorias e funcionalidades podem ser indicadas posteriormente com o decorrer e uso do

projeto, sempre levantando novas alternativas para exibição de dados, melhoria de layout e interface do usuário e a experiência geral dos usuários do projeto.

## 8 CONCLUSÃO

Conforme apresentado no presente estudo, seja por motivações religiosas, a busca por prestígio ou pelo simples gesto de caridade, o voluntariado sempre fez parte da sociedade, desde os seus registros mais remotos. No cenário atual, as instituições não governamentais representam uma função vital na organização e manutenção da sociedade, como ações de reflorestamento, campanhas de alfabetização e apoio a comunidades em vulnerabilidade, estas primordiais para acompanhar o volátil cenário mundial.

Para sanar as dores experienciadas pelas Organizações Não Governamentais (ONGs), este projeto, seguindo uma metodologia incremental, propõe um sistema web inovador como resposta às necessidades, oferecendo uma plataforma centralizada e segura para conectar voluntários e ONGs, que utiliza as tecnologias web mais atuais; o que por sua vez, permite além da criação de uma interface amigável e com recursos de acessibilidade, um canal para que as organizações encontrem candidatos compatíveis com seus projetos.

Retifica-se que antes de iniciar as atividades em ambiente produtivo e com dados reais, foram analisadas e utilizadas todas as ferramentas disponíveis para aferir o aprofundamento em relação às obrigações legais, sempre com o intuito de preservar, tanto os direitos relacionados à liberdade quanto nos atrelados a privacidade das pessoas; no que abrange as atividades de coleta, produção, recepção, classificação, uso, transmissão, armazenamento e processamento de dados, como estabelecido na Lei Geral de Proteção de Dados (LGPD).

Uma vez realizado o levantamento de dados por meio de pesquisas com ênfase em ONGs para entender profundamente os desafios e as necessidades, os desenvolvimentos de código foram direcionados para tornar todas as ações executadas dentro da plataforma o mais fluidas e simples possível. Coletando ao longo do processo feedbacks e identificando oportunidades de otimização.

As ferramentas escolhidas para este projeto tiveram entre seus critérios de escolha a capacidade de reduzir o tempo necessário para desenvolvimento e implementação de funcionalidades complexas, desenvolvimento seguro, alinhamento com o mercado contemporâneo, eficiência computacional, compatibilidade com múltiplas plataformas e legibilidade, se destacando frente às demais, as que demandam a menor quantidade de código e maior facilidade de manutenção. Alinhada à arquitetura de microsserviços, foi possível escalar e manter módulos do sistema de forma independente, realizando a comunicação entre estes, através de APIs que integram a biblioteca Brasilapy para acesso a dados públicos, como a

OpenCage Geocoder API utilizada para análise da distribuição geográfica das ONGs, facilitando a identificação de clusters e facilitando a tomada de ação com maior impacto social.

Buscando atingir de forma segura seus objetivos, este projeto foi desenvolvido em conformidade também com o Regulamento Geral de Proteção de Dados (GDPR) europeu, com normas como ISO/IEC 25010, 27001 e 9001 demonstrando uma preocupação dos autores com a privacidade e o tratamento seguro das informações pessoais, sendo a transparência no uso dos dados, segurança contra acessos não autorizados pilares fundamentais do sistema. A conformidade com a qualidade e confiabilidade do software, não apenas garante a proteção dos dados dos usuários, mas também fomenta a confiança na plataforma, incentivando uma maior participação, engajamento e colaboração entre as partes interessadas.

Conclui-se que este sistema web representa uma ferramenta poderosa para impulsionar o engajamento cívico e fortalecer o terceiro setor no Brasil. Ao simplificar o processo de conexão entre voluntários motivados e instituições com ação em áreas sensíveis, a plataforma tem o potencial de catalisar um impacto social significativo. A otimização do processo de recrutamento de voluntários permitirá que as ONGs possam expandir suas operações e alcancem um número maior de beneficiários, representando a automatização de atividades complexas antes totalmente manuais e repetitivas, contribuindo com os avanços no desenvolvimento sustentável, redução das desigualdades e a inclusão social em diversas áreas como educação, saúde e meio ambiente.

Aproveitando do cenário favorável, os autores deste estudo ambicionam, com crescente aumento no interesse em realizar ações com propósito, munir as ONGs com as tecnologias necessárias para transformar as práticas gerenciais e de mensuração de desempenho interno e externo, em um meio transformador e acessível para expandir o impacto social já realizado, permitindo juntamente mais colaboração, conexões verdadeiramente humanas e amparar o Brasil com uma nova alternativa dentre as plataformas digitais para permitir a colaboração em projetos humanitários e ambientais.

## REFERÊNCIAS

BLISCHAK John D.; DAVENPORT, Emily R.; WILSON Greg. **A Quick Introduction to Version Control with Git and GitHub**. 2016. PLOS Computational Biology 12(1): e1004668. Disponível em: <https://doi.org/10.1371/journal.pcbi.1004668>. Acesso em: 29 abr 2025.

BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, Grady. **UML: Guia do Usuário**. 2. ed. Rio de Janeiro: Editora Elsevier, 2006.

BRASIL API. **Transformando o Brasil em uma API**. Brasil API. Disponível em: <https://brasilapi.com.br/>. Acesso em: 15 mar 2025.

BRASIL. **Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência)**. Lei nº 13.146, de 6 de julho de 2015. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2015/lei/113146.htm](http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/113146.htm). Acesso em: 20 abr 2025

BRASIL. **Lei nº 13.709, de 14 de agosto de 2018**. Lei Geral de Proteção de Dados Pessoais (LGPD). Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/113709.htm](http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm). Acesso em: 20 jan 2025.

BYLIEVA, D. **The Language of Human-Machine Communication**. 2020. Disponível em: <https://soctech.spbstu.ru/en/article/2020.1.4/>. Acesso em: 29 nov 2024.

COAD, Peter. **Object-Oriented Patterns**. Communications of the ACM, V. 35. nº 9. p. 152-159. [S.l.: s.n.]. 1992. Acesso em: 28 nov 2024.

DJANGO SOFTWARE FOUNDATION. **Django Documentation**. Disponível em: <https://docs.djangoproject.com/>. Acesso em: 20 jan 2025.

ERICH, Gamma et al. **Padrões de projeto**. Soluções Reutilizáveis de Software, 2000. Disponível em: [https://books.google.com.br/books?hl=pt-BR&lr=&id=U91CYCqTCgkC&oi=fnd&pg=PA9&dq=Padr%C3%B5es+de+Projetos:+Solu%C3%A7%C3%B5es+Reutiliz%C3%A1veis,+Gamma&ots=P\\_i36xgNJS&sig=ah5gP804w8](https://books.google.com.br/books?hl=pt-BR&lr=&id=U91CYCqTCgkC&oi=fnd&pg=PA9&dq=Padr%C3%B5es+de+Projetos:+Solu%C3%A7%C3%B5es+Reutiliz%C3%A1veis,+Gamma&ots=P_i36xgNJS&sig=ah5gP804w8)

HCKJQaqsNf90Z5qtI#v=onepage&q=Padr%C3%B5es%20de%20Projetos%3A%20Solu%C3%A7%C3%B5es%20Reutiliz%C3%A1veis%2C%20Gamma&f=false. Acesso em: 27 abr 2025.

FAYAD, M.E.; SCHMIDT, D.C. (eds) **Object-Oriented Application Frameworks**. Communications of the ACM, V. 40, n° 10, p. 32-38, 1997. Acesso em: 28 nov 2024.

FOWLER, Susan J. **Microserviços prontos para a produção**: Construindo sistemas padronizados em uma organização de engenharia de software. [S. l.]. Novatec. 2017.

FURNIEL, Igor. **ISO 9001: Tudo o que você precisa saber sobre a certificação**. Disponível em: <https://certificacaoiso.com.br/iso-9001/>. Acesso em: 15 abr 2025.

GitHub Docs. **Sobre o GitHub e o Git**. Disponível em: <https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git>. Acesso em: 29 abr 2025.

GOYAL, Amit; et al. **Django – The High-Level Python Web Framework**. Disponível em: <https://docs.djangoproject.com/>.

GOYAL, Arpit e col. **Django Introduction**. MDN Web Docs. 2024. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>. Acesso em: 08 dez 2024.

HOLVANDER, Anna. **The Django Way**: Build Secure and Scalable Web Applications. 1. ed. New York: Packt Publishing, 2020.

Ipwhois.io. **API Documentation**. Disponível em: <<https://ipwhois.io/documentation>>. Acesso em: 29 abr 2025.

JIawei, Han; MICHELINE, Kamber; JIAN, Pei. **Data Mining: Concepts and Techniques**. 3rd. Morgan kaufmann. 2012. Disponível em: <https://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf>. Acesso em: 27 abr 2025.



JOHNSON, Ralph E.; FOOTE B. **Designing Reusable Classes**. [S.l.]: Journal of Object Oriented Programming – JOOP, 1988.

KRAUSE, Jeffery. **Modern Web Development**. 1. ed. Birmingham: Packt Publishing, 2018.

KRUG, Steve. **Não me faça pensar**. Alta Books, 2014. Disponível em: <https://previewer.homologaprojeto.com.br/Livros/N%C3%A3o%20Me%20Fa%C3%A7a%20Pensar%20-%20Steve%20Krug.pdf>. Acesso em: 30 mar 2025.

LOUZADA, Vinícius. **O que é Git e Github: Os Primeiros Passos Nessas Ferramentas**. 2024. Disponível em: [https://www.alura.com.br/artigos/o-que-e-git-github?srsId=AfmBOop2W8ScfA5WnJxi4odwuxx33TLMJsiLlBbj\\_wnwkUJS42IHuR29](https://www.alura.com.br/artigos/o-que-e-git-github?srsId=AfmBOop2W8ScfA5WnJxi4odwuxx33TLMJsiLlBbj_wnwkUJS42IHuR29). Acesso em: 30 abr 2025.

MACHADO, Maria Istela Cagnin. **Modelagem de Software**. Universidade Federal de Mato Grosso do Sul, 2015. Disponível em: [https://fernandommota.github.io/academy/disciplines/2015/analise\\_projeto\\_software/files/03\\_modelagem.pdf](https://fernandommota.github.io/academy/disciplines/2015/analise_projeto_software/files/03_modelagem.pdf). Acesso em: 29 nov. 2024.

MICROSOFT. **Normas de Gestão de Segurança da Informação ISO/IEC 27001:2013**. 2023. Disponível em: <https://learn.microsoft.com/en-us/compliance/regulatory/offering-iso-27001>. Acesso em: 14 abr 2025.

MOREIRA, Ana. **Armazenamento E Gerenciamento de Dados Em Uma Empresa**. Campo Grande, 2018. Disponível em: <https://repositorio.pgsscogna.com.br/bitstream/123456789/25059/1/Ana%20Carolina-%20TCC-Final.pdf>. Acesso em: 21 mar 2025.

MURTA, Leonardo Gresta Paulino. **Diagrama de Atividades**. Instituto de Computação. Disponível em: <https://leomurta.github.io/courses/es1/aula9.pdf>. Acesso em: 28 nov 2024.

OLIVEIRA, A. S.; CALLADO, A. A. C. **Tecnologia, tecnologia da informação e práticas gerenciais de mensuração de desempenho: um estudo em ONGs brasileiras**. Anais do Congresso Brasileiro de Custos, 2015. Disponível em:

<https://anaiscbc.emnuvens.com.br/anais/article/download/4385/4385/4495>. Acesso em: 20 jan 2025.

OLIVEIRA, Danielle. **MER e DER: Definições, Banco de Dados e Exemplos**. 2023. Disponível em: <https://www.alura.com.br/artigos/mer-e-der-funcoes?srsId=AfmBOooqA86wwwt64E-JxSOszG0ZUjeesQtGx3lhimLPr7jxhiCCPD9f>. Acesso em: 28 mar 2025.

PASQUALI, T. E. et al. **Criação de dashboards analíticos em Python para tomada de decisão**. Caderno Pedagógico, [S. l.], v. 21, n. 8, p. e6539, 2024. DOI: 10.54033/cadpedv21n8-084. Disponível em: <https://ojs.studiespublicacoes.com.br/ojs/index.php/cadped/article/view/6539>. Acesso em: 29 nov 2024.

PYTHON. **Python - Documentation**. Disponível em: <https://www.python.org/doc/>. Acesso em: 15 mar 2025.

RIES, Eric. **The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses**. Crown Currency, 2011. Disponível em: <https://dn790008.ca.archive.org/0/items/TheLeanStartupErickRies/The%20Lean%20Startup%20-%20Erick%20Ries.pdf>. Acesso em: 27 abr 2025.

ROSSUM, Guido Van; FRED L., Drake, Jr. **An Introduction to Python**. Network Theory Limited, 2003. Disponível em: <http://atk.fam.free.fr/fichiers/stage/Python/JF/site/pytut.pdf>. Acesso em: 29 nov 2024.

RUAS, Pedro. **O Guia Prático Do MySQL**. 2005. Disponível em: <https://www.centroatl.pt/titulos/tecnologias/imagens/excerto-e-book-caoguiapraticodomysql.pdf>. Acesso em: 15 mar 2025.

SILVA, Rodrigo. **ISO/IEC 25010 E SUA IMPORTÂNCIA PARA A QUALIDADE DE SOFTWARE**. Disponível em: <https://blog.onedaytesting.com.br/iso-iec-25010/>. Acesso em: 15 abr 2025.

SILVA, R. O.; SILVA, I. R. S. **Linguagem de Programação Python**. Tecnologias em Projeção. v. 10. n° 1. p. 55-71. [S.l.: s.n.]. 2019.

Software Freedom Conservancy. **Git**. Disponível em: <https://git-scm.com/>. Acesso em: 29 abr 2025.

SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo, SP: Pearson, 2018. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 28 nov 2024.

TWITTER. **Bootstrap – The most popular HTML, CSS, and JS library in the world**. Disponível em: <https://getbootstrap.com/docs/>. Acesso em: 26 nov. 2024.

VAMSI, K. Manikanta e col. **IOP Conference Series: Materials Science and Engineering**. 2021. Disponível em: <https://iopscience.iop.org/article/10.1088/1757-899X/1042/1/012019/pdf>. Acesso em: 06 dez 2024.

WANG, K.; YAN, M.; ZHANG, H.; HU, H. **Unified abstract syntax tree representation learning for cross-language program classification**. ArXiv (Cornell University). 2022. Disponível em: <https://dl.acm.org/doi/10.1145/3524610.3527915>. Acesso em: 02 dez 2024.

World Wide Web Consortium (W3C). **Web Content Accessibility Guidelines (WCAG)**. Disponível em: <https://www.w3.org/WAI/standards-guidelines/wcag/>. Acesso em: 30 mar 2025

W3C. **World Wide Web Consortium**. Disponível em: <https://www.w3.org/>. Acesso em: 26 nov 2024.

YUILL, Simon; HALPIN, Harry. **Python**. 2006. Disponível em: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97bfafd514ca2abb7e2c5c86bb>. Acesso em: 28 nov 2024.

## **APÊNDICE A – DOCUMENTAÇÃO DA VERSÃO 1**

### **Documentação do Projeto - Funcionalidades, Erros e Melhorias**

Este documento tem como objetivo detalhar as funcionalidades desenvolvidas no projeto, os erros encontrados durante a execução e as possíveis melhorias a serem implementadas. Ele serve como um guia para futuras referências e para o planejamento de próximas etapas de desenvolvimento.

Funcionalidades Desenvolvidas:

#### **1. Navegação Geral (Barra Superior)**

- Home: Retorna o usuário para a página inicial, onde há a lista de vagas abertas.
- Cadastro de ONG: Redireciona para o formulário de cadastro específico de ONGs.
- Cadastro de Voluntário: Redireciona para o formulário de cadastro específico de Voluntários
- Login: Redireciona para a página de autenticação, onde o usuário pode inserir suas credenciais.
- Barra de Pesquisa: Permite que o usuário pesquise informações dentro do sistema (vagas, ONGs, etc.), embora o comportamento exato ainda precise ser analisado com mais detalhes.

#### **2. Cadastro de ONG (Formulário de Cadastro (Tela “Cadastrar ONG”))**

- Nome da ONG: Nome legal ou oficial da organização
- CNPJ: Identificação fiscal da ONG
- CEP, Logradouro, Número, Bairro, Cidade, Estado: Campos relacionados ao endereço completo da ONG
- Telefone: Contato principal (fixo ou celular)
- E-mail: Endereço de e-mail institucional ou de contato
- Site: (Opcional/obrigatório, dependendo das regras) para organizações que possuem página na internet
- Usuário: Nome de usuário para login.
- Senha e Confirmar Senha: Campos para criação e confirmação de senha.
- Validação de Senha: Funcionalidade de segurança que emite um aviso indicando a verificação de complexidade de senha

- Botão “Enviar para análise”: Aciona o processo de criação de conta/ONG no sistema, realizando o envio de dados para aprovação manual

### **3. Cadastro de Voluntário (Formulário de Cadastro (Tela “Cadastrar Voluntário”))**

- Nome: Nome completo do(a) candidato(a)
- CPF: Cadastro de pessoa física completo do(a) candidato(a)
- CEP, Logradouro, Número, Bairro, Cidade, Estado: Campos relacionados ao endereço completo do(a) candidato(a)
- Telefone: Contato principal (fixo ou celular)
- E-mail: Endereço de e-mail institucional ou de contato
- Usuário: Nome de usuário para login.
- Senha e Confirmar Senha: Campos para criação e confirmação de senha.
- Data de Nascimento: Campo para inserir a data de nascimento
- Validação de Senha: Funcionalidade de segurança qual emite um aviso indicando a verificação de complexidade de senha
- Botão “Enviar para análise”: Aciona o processo de criação de conta/voluntário no sistema, realizando o envio de dados para aprovação manual

### **4. Login**

- Campos de Credenciais
- Usuário: Identificação para login, podendo ser o usuário registrado tanto uma ONG quanto por voluntário.
- Senha: Campo de senha para autenticação
- Lembrar Credenciais: *Checkbox* que, ao ser marcado, pode manter o usuário logado ou lembrar seu nome de usuário/senha em logins futuros.
- Botão “Login”: Inicia a sessão do usuário, validando credenciais no banco de dados.
- Recuperar Senha / Redefinir Senha: Funcionalidade de recuperação ou redefinição de senha
- Cadastro Rápido: Link para direcionar o usuário para as telas de cadastro caso ainda não seja registrado

### **5. Visão de Permissões e Fluxos Possíveis**

- Cadastro e Aprovação: O envio de formulário (“Enviar para análise”) sugere que pode haver um fluxo de aprovação ou ativação manual de contas de ONG ou Voluntário.

- Níveis de Acesso: ONG, Voluntário, e administradores tem permissões adicionais para gerenciar e aprovar cadastros e/ou criar vagas

- Lista de Vagas: Funcionalidade para que as ONGs criem vagas e os voluntários se candidatem

- Painel de Administração: Local para gerenciar permissões, usuários, vagas e demais dados.

- Gerenciamento de Grupos e Permissões: Painel para controlar o que cada grupo (ONGs ou Voluntários) pode fazer no sistema.

#### **Erros Encontrados:**

<b>Erro</b>	<b>Descrição</b>	<b>Solução (se aplicável)</b>	<b>Prioridade</b>
Inserir valores errados no CEP	Ao digitar caracteres de texto, números além do permitido	Restringir as opções de entrada de dados do usuário	Média-alta

## APÊNDICE B – FORMULÁRIO DE PESQUISA

1. E-mail:

---

2. Qual sua idade?

☐ 14 - 17 anos

☐ 18 - 24 anos

☐ 25 – 39 anos

☐ Mais de 40 anos

3. Você já realiza trabalho voluntário?

☐ Sim, já realizo

☐ Não

4. Se não realiza, já desejou realizar trabalho voluntário?

☐ Sim, já quis/quero

☐ Não tenho interesse

5. Se deseja realizar, qual o maior empecilho encontrado? Selecione as opções que se aplicam

☐ Não encontro ONGs com que eu me identifique

☐ Falta de Tempo

☐ Não sei onde encontrar vagas de trabalho voluntário

☐ Outros

6. Qual tipo de trabalho voluntário você realiza ou gostaria de realizar? Selecione as opções que se aplicam.

☐ Trabalho relacionado a crianças (leitura, alfabetização, etc)

☐ Trabalho relacionado a animais e pets

☐ Trabalho relacionado a idosos

☐ Trabalho relacionado a comunidades carentes/ socialmente vulneráveis

☐ Outros

7. Por quanto tempo você gostaria de realizar trabalho voluntário?

- ☐ 1 – 3 semanas
- ☐ 1 – 6 meses
- ☐ 1 ano ou mais
- ☐ Não tenho certeza

8. Qual a sua disponibilidade para realizar trabalho voluntário?

- ☐ Durante a semana
- ☐ Durante sábados e/ou domingos
- ☐ Apenas alguns minutos por dia
- ☐ Não tenho certeza/depende do tipo de trabalho a ser realizado.

9. Caso deseje compartilhar, pode escrever sobre o motivo que te levou a querer realizar trabalho voluntário?



## APÊNDICE C – RELATÓRIO DE TESTES DA APLICAÇÃO

### Relatório de Testes da Aplicação

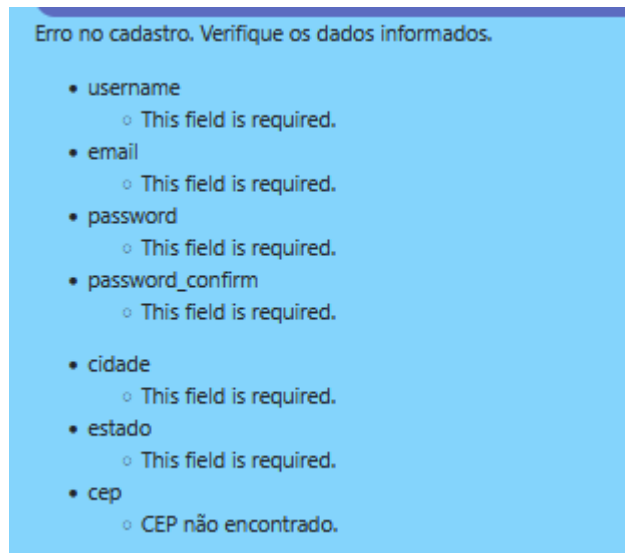
Este documento tem como principal objetivo apresentar os resultados obtidos dos testes realizados na aplicação WeOng, incluindo capturas de tela dos resultados obtidos em cada teste.

#### Testes Realizados:

##### 1. Testes de Cadastro com Dados Inválidos

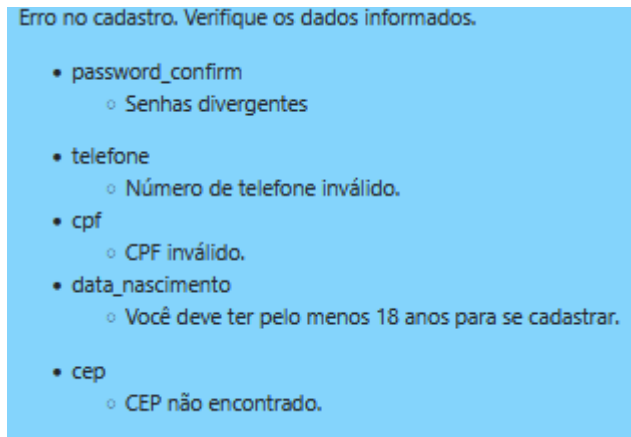
##### 1.1. Cadastrar ONG com CNPJ inválido

Ao inserir letras ou quantidade incorreta de dígitos no campo de CNPJ, o sistema recusa o cadastro e exibir uma mensagem de erro apropriada.



##### 1.2. Cadastrar Voluntário com CPF inválido

Ao digitar um CPF com quantidade de dígitos incorreta ou caracteres diferentes de números a aplicação impede o prosseguimento e informar que o CPF é inválido.



### 1.3. Cadastrar qualquer usuário (ONG ou Voluntário) com campos obrigatórios vazios

Ao deixar em branco campos como "Nome" ou "E-mail" e tentar enviar, o sistema apresenta uma mensagem de erro solicitando o preenchimento dos campos.

**Cadastro Voluntário**

Nome Completo  CPF

**Preencha este campo.** Bairro

Logradouro  Número

Cidade  Estado

Telefone  E-mail

Usuário  Data de nascimento

Senha  Confirmar Senha

Atenção! Para sua segurança, a senha deve conter:

- Números
- Letras
- Caractere Especial

## 2. Testes de Conflito/Colisão de Dados

### 2.1. Usuário duplicado

Ao tentar criar um usuário (ONG ou Voluntário) com o mesmo "Usuário" (login) ou e-mail que já exista no banco, o sistema bloqueia o cadastro e apresentar mensagem indicando que já existe um usuário com aquele login/email.

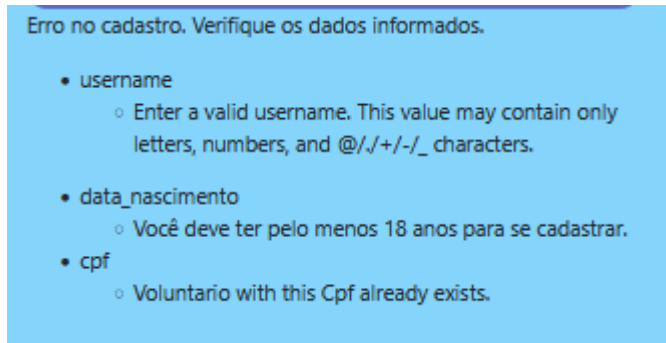
A imagem mostra a interface de um formulário de "Cadastro Voluntário" em um sistema web. O formulário possui campos para: Nome Completo, CPF, CEP, Bairro, Logradouro, Número, Cidade, Estado (menu suspenso), Telefone, E-mail, Usuário, Data de nascimento (formato dd/mm/aaaa), Senha e Confirmar Senha. Abaixo dos campos, há uma seção de "Atenção!" com requisitos para a senha: conter números, letras e caracteres especiais. Um botão azul "Enviar para análise" está presente. Na base do formulário, uma mensagem de erro vermelha indica: "Erro no cadastro. Verifique os dados informados." com um detalhe específico: "• username: A user with that username already exists."

## 2.2. Submissão de CEP já cadastrado com outro formato de endereço

Ao tentar inserir o mesmo CEP com dados de "Logradouro/Bairro/Cidade" diferentes, o sistema não apresentou erro

## 2.3. Idade inválida e CPF já utilizado

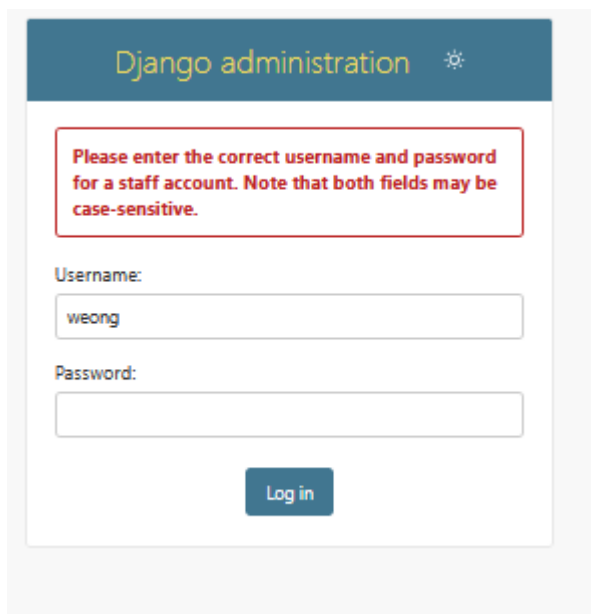
Ao tentar inserir uma idade fora do permitido (9/9/9999) e utilizar o mesmo CPF cadastrado previamente, o sistema apresentou uma mensagem solicitando o ajuste das informações



### 3. Testes de Permissões e Acessos

#### 3.1. Acessar URL de administração (Django Admin) sem ser superusuário

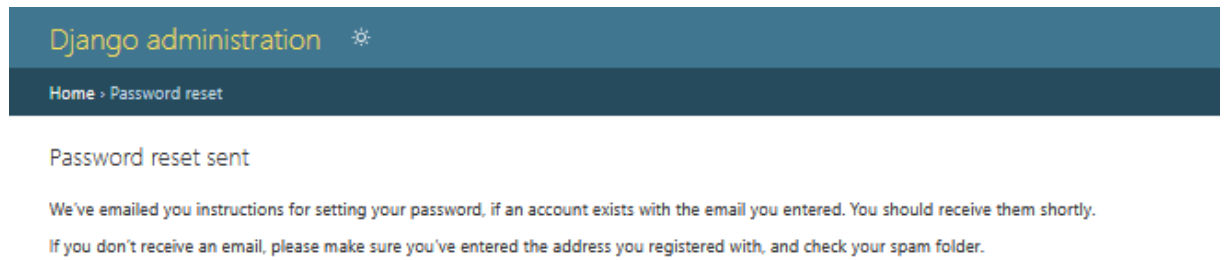
Ao tentar entrar em <http://localhost:7000/admin> com um usuário comum (ONG ou Voluntário), o sistema bloquear o acesso com mensagem de erro de permissão.



### 4. Testes de Autenticação/Recuperação de Senha

#### 4.1. Esqueci minha senha – E-mail inválido

Ao selecionar a opção ‘Redefinir senha’ na tela de login, é aberta uma nova tela com "Password reset" do Django Admin. Ao inserir um e-mail que não existe no sistema ou que esteja incorreto o sistema apresenta a mensagem que caso encontre o e-mail informado será realizado o envio do e-mail de redefinição.



## 5. Testes de Integração e Navegação

### 5.1. Inserir dados incorretos no formulário de busca

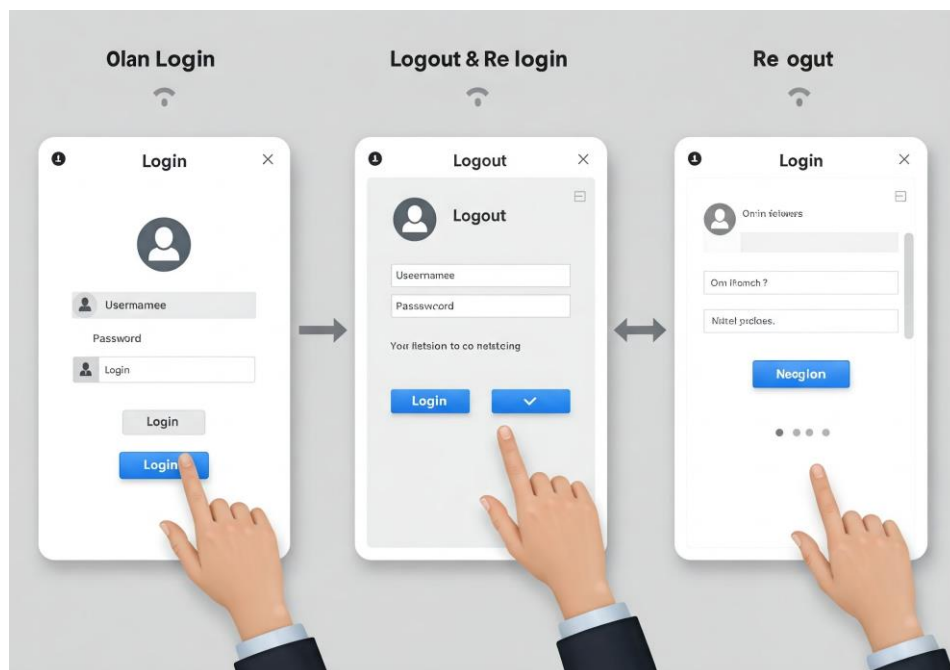
Ao digitar diversos caracteres especiais, números e credenciais na barra de pesquisa e verificar a aplicação, não foram retornados erros, vulnerabilidades ou expostas informações.

### Testes Não-Realizados

#### 1. Fluxo de "Login" seguido de "Logout" e re-login

Se faz pendente a realização de testes de realizar o login e logout e tentar, após sair da conta, logar novamente, possivelmente com dados errados ou outro usuário.

Este teste tem como objetivo garantir que a sessão anterior foi finalizada corretamente e que seja possível fazer o novo login sem problemas.



#### 2. Testes de Desempenho e Conexão

##### 2.1. Múltiplos Cadastros Simultâneos

Se faz pendente a realização de diversas requisições de cadastro simultaneamente.

Este teste tem como objetivo verificar se o servidor Django consegue lidar com as requisições sem falhar ou corromper dados no banco.

## 2.2. Acesso concorrente à criação de vagas

Se faz necessário testar a funcionalidade de criar vagas, realizando a criação de múltiplas vagas ao mesmo tempo com o mesmo usuário e/ou diferentes usuários.

Este teste tem como objetivo avaliar se o sistema deve manter a consistência dos dados, sem duplicar registros indevidamente.

## Apêndice D – Termo de Privacidade

Esta Política de Privacidade ("Política") explica de maneira clara, transparente e objetiva como são tratados os dados pessoais dos usuários ("Titular") ao acessarem os conteúdos disponibilizados no site desenvolvido e/ou fornecido pela ASSOCIAÇÃO WeONG, em conformidade com a Lei 13.709/2018 – Lei Geral de Proteção de Dados ("LGPD").

### **Através desta Política, o Titular será informado:**

1. Quais dos seus dados pessoais tratamos;
2. Como usamos seus dados pessoais;
3. Como protegemos seus dados pessoais;
4. Com quem os seus dados pessoais podem ser compartilhados;
5. Quais são os direitos dos Titulares em relação ao tratamento de seus dados pessoais e
6. O canal de comunicação a ser utilizado pelos Titulares para o exercício de seus direitos.

## **GLOSSÁRIO**

Para facilitar a compreensão dos termos técnicos presentes nesta Política, explicamos alguns conceitos básicos utilizados:

- **Agentes de tratamento:** Termo que abrange os conceitos de controlador e operador de dados pessoais, conforme definido na Lei 13.709/2018.
- **Autoridade Nacional de Proteção de Dados (ANPD):** Órgão da administração pública federal, vinculado à Presidência da República, responsável por fiscalizar e regulamentar a aplicação da Lei 13.709/2018.
- **Bloqueio:** Suspensão temporária de qualquer operação de tratamento, mediante a guarda do dado pessoal ou do banco de dados.

- **Compartilhamento de dados:** Comunicação, difusão, transferência internacional, interconexão de dados pessoais ou tratamento compartilhado de bancos de dados pessoais, mediante autorização específica entre entidades públicas ou privadas.
- **Consentimento:** Manifestação livre, informada e inequívoca pela qual o titular concorda com o tratamento de seus dados pessoais para uma finalidade determinada.
- **Conteúdo:** Qualquer informação, dado, comunicação, software, foto, vídeo, gráfico, música, som e outros materiais e serviços que podem ser visualizados pelos titulares no Site, incluindo mensagens, conversas, bate-papo e outros conteúdos originais.
- **Controlador de dados:** Pessoa natural ou jurídica, de direito público ou privado, responsável pelas decisões relativas ao tratamento de dados pessoais.
- **Cookies:** Arquivos pequenos salvos no computador do usuário que armazenam preferências e outras informações utilizadas nas páginas da internet que visitam.
- **Dados manifestamente públicos:** Dados publicados e compartilhados abertamente pelo Titular.
- **Dados pessoais:** Todos os dados relacionados a uma pessoa natural, identificada ou identificável.
- **Dados pessoais anonimizados:** Dados relativos a titulares que não possam ser identificados.
- **Dados pessoais sensíveis:** Dados pessoais sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculados a uma pessoa natural.
- **Eliminação:** Exclusão de dado ou de conjunto de dados armazenados em banco de dados, independentemente do procedimento empregado.
- **Encarregado de proteção de dados/Data Protection Officer (DPO):** Pessoa indicada pelo controlador e operador para atuar como canal de comunicação entre o controlador, os titulares de dados e a Autoridade Nacional de Proteção de Dados (ANPD).
- **Finalidade:** Princípio da Lei 13.709/2018 que prevê a realização do tratamento para propósitos legítimos, específicos, explícitos e informados ao Titular, sem possibilidade de tratamento posterior de forma incompatível com essas finalidades.

- Incidente de segurança: Evento adverso, confirmado ou sob suspeita, relacionado à segurança de sistemas de computação ou serviços de computadores.
- Legítimo interesse: Base legal que permite a realização de atividades de tratamento de dados para atender interesses legítimos do controlador ou de terceiros.
- Operador de dados: Pessoa natural ou jurídica, de direito público ou privado, que realiza o tratamento de dados pessoais em nome do controlador.
- Tratamento de dados: Qualquer operação realizada com dados pessoais, como coleta, produção, recepção, classificação, utilização, acesso, reprodução, transmissão, distribuição, processamento, arquivamento, armazenamento, eliminação, avaliação ou controle da informação, modificação, comunicação, transferência, difusão ou extração.

## **PRINCÍPIOS DAS ATIVIDADES DE TRATAMENTO DE DADOS PESSOAIS**

Esta Política foi criada com o objetivo de reiterar o compromisso da WeONG com a segurança, privacidade, acessibilidade e transparência nas atividades de tratamento dos dados pessoais a partir de nosso canal: Site. Toda e qualquer atividade de tratamento de dados pessoais descrita será regida pelos princípios da finalidade, adequação, necessidade, livre acesso, não discriminação, prevenção, qualidade dos dados, segurança, transparência, responsabilização e prestação de contas.

## **OBJETO DE TRATAMENTO DE DADOS PESSOAIS**

O acesso ao site é aberto a todos, mas pode ser necessário que o usuário forneça dados pessoais para acessar alguns de nossos conteúdos ou serviços. Nesse contexto, podemos coletar os seguintes dados:

- Nome
- E-mail
- Empresa/organização
- Áreas de interesse do usuário

É fundamental que os dados pessoais fornecidos sejam precisos, verdadeiros e completos, sendo responsabilidade do usuário garantir que as informações fornecidas são corretas e completas.



## **FINALIDADE DE TRATAMENTO DE DADOS PESSOAIS**

A ASSOCIAÇÃO WeONG realiza o tratamento de dados pessoais dos Titulares de forma transparente, limitando-se ao mínimo necessário para o cumprimento das finalidades lícitas e específicas, descritas a seguir:

- Permitir que a ASSOCIAÇÃO WeONG responda aos questionamentos ou dúvidas enviadas pelo Titular;
- Permitir que a ASSOCIAÇÃO WeONG entre em contato com o Titular interessado em ser voluntário ou parceiro;
- Permitir que a ASSOCIAÇÃO WeONG entre em contato com o Titular interessado em saber mais sobre o nosso trabalho, informando periodicamente sobre nossas atividades e solicitando doações.

## **COLETA DE DADOS PESSOAIS**

A ASSOCIAÇÃO WeONG coleta dados pessoais dos Titulares das seguintes formas:

- Formulários de contato, dúvidas, sugestões, consultas, voluntariado ou parceria.
- Cadastro de usuários para receber a newsletter.

Além disso, a ASSOCIAÇÃO WeONG obtém dados pessoais automaticamente do Titular por meio da utilização de "cookies" e outras tecnologias similares, com o intuito de garantir a melhor experiência para os Titulares, bem como para a análise estatística de tendências, administração das páginas na web e mídias, e rastreamento do comportamento do usuário.

Os tipos de cookies utilizados são os seguintes:

### **Cookies**

Cookies são arquivos enviados pelo servidor da plataforma para o computador ou celular do usuário, com a finalidade de identificar o dispositivo e obter dados de acesso. Isso permite personalizar a utilização do nosso site ou redes sociais de acordo com o perfil do usuário, além de garantir maior segurança para os usuários da plataforma. Quando você utiliza os Serviços, nós, juntamente com nossos prestadores de serviço terceirizados, podemos usar as Informações de Utilização para entender mais efetivamente como os Serviços são utilizados e, assim, melhorar a usabilidade de nossos Serviços em razão de nosso legítimo interesse para isso.

É importante destacar que não utilizamos cookies de comportamento/marketing que se integram com plataformas de redes sociais para controle da exibição de anúncios. Esses cookies

são de terceiros (plataformas utilizadas para campanhas de marketing, como Facebook, Google, LinkedIn) e podem cruzar informações dos Usuários com seus respectivos perfis nessas plataformas. Caso não queira fornecer essas informações, você pode desativar o uso de cookies em seu navegador. A seguir, disponibilizamos links para instruções de como fazer isso nos diferentes navegadores: Google Chrome, Internet Explorer e Safari.

### **ARMAZENAMENTO DE DADOS PESSOAIS**

Após o término da relação que levou ao fornecimento dos seus dados à ASSOCIAÇÃO WeONG, a manutenção de suas informações ocorrerá apenas para fins legítimos e essenciais, como a tomada de decisões estratégicas relacionadas aos nossos projetos, o cumprimento de obrigações legais e fiscais, resolução de litígios, prevenção à fraude e manutenção da sua segurança.

No caso de solicitação de exclusão de informações, reservamo-nos o direito de preservar quaisquer dados necessários para:

- Cumprimento de obrigações legais, fiscais e auditorias, sendo necessário armazenar os dados até o fim do período exigido por lei.
- Atender a interesses legítimos, realizando o balanceamento dos impactos desse interesse nos direitos do titular dos dados. Por exemplo, quando necessitamos dos dados para comprovar o impacto do nosso projeto ou para implementar mecanismos de prevenção à fraude e manutenção da segurança do usuário.

### **SEGURANÇA DE DADOS PESSOAIS**

A ASSOCIAÇÃO WeONG se responsabiliza pela adoção e manutenção de medidas razoáveis de segurança, técnicas e administrativas, com o objetivo de proteger os dados pessoais contra situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito, bem como contra o processamento discriminatório de informações, conforme os melhores padrões de mercado e a legislação aplicável.

As medidas de segurança adotadas incluem:

- Emprego de softwares de alta tecnologia para impedir acessos não autorizados aos nossos sistemas;
- Utilização de métodos de criptografia;
- Adoção de mecanismos de autenticação de acesso aos registros, capazes de individualizar o responsável pelo tratamento de dados;
- Limitação de acesso aos locais de armazenamento de dados, entre outras.

O acesso às informações armazenadas pela ASSOCIAÇÃO WeONG é restrito aos profissionais autorizados e qualificados, dentro dos limites da necessidade para desempenharem suas funções, e que estão sujeitos às obrigações de confidencialidade. A ASSOCIAÇÃO WeONG se compromete a informar ao Titular e aos órgãos de proteção de dados pessoais sobre qualquer incidente de segurança que possa acarretar risco ou prejuízo relevante aos Titulares, em um prazo razoável, justificando eventual atraso na comunicação.

### **COMPARTILHAMENTO DE DADOS PESSOAIS**

Como regra, a ASSOCIAÇÃO WeONG não realiza o compartilhamento de dados pessoais, tampouco a transferência internacional desses dados sem seu prévio e expresso consentimento, considerando que os dados pessoais são coletados e tratados de acordo com a legislação brasileira.

Entretanto, alguns provedores de serviços essenciais para a realização dos nossos serviços estão sediados em outros países. Nesses casos, transferimos os dados estritamente para o cumprimento de contratos que dependam do uso de tais serviços. Além disso, a ASSOCIAÇÃO WeONG opta por parceiros que possuam Políticas de Privacidade claras e compromissos com o cumprimento da LGPD e do GDPR (Regulamento Geral de Proteção de Dados da União Europeia), não autorizando o uso indevido dos dados por esses fornecedores.

### **DIREITOS DO TITULAR**

O Titular dos dados pode exercer, a qualquer momento, através dos canais de atendimento da ASSOCIAÇÃO WeONG, todos os direitos garantidos pela Lei 13.709/2018 e outras legislações aplicáveis, tais como:

- Confirmação da existência do tratamento de dados pessoais e acesso aos mesmos;
  - Correção de dados pessoais incompletos, inexatos ou desatualizados;
  - Portabilidade dos dados pessoais a outro fornecedor de serviço ou produto, observados os segredos comercial e industrial;
  - Eliminação dos dados pessoais tratados com o consentimento do Titular;
- Informação das entidades públicas e privadas com as quais a instituição realizou uso compartilhado de dados;
- Informação sobre a possibilidade de revogar ou não fornecer consentimento, bem como as consequências dessa negativa;
  - Revisão de decisões tomadas unicamente com base em tratamento automatizado de dados pessoais que afetem seus interesses;

- Requerer a aplicação das normas de defesa do consumidor na tutela da proteção de seus dados, quando aplicável;
- Requisitar a exclusão de nossas listas de comunicação ou marketing, assim como o cancelamento do envio de mensagens e emails, caso não deseje mais receber.

### **EXERCÍCIO DE DIREITOS DO TITULAR**

A ASSOCIAÇÃO WeONG facilitará o exercício de direitos dos titulares de dados, sempre que possível e nos termos da lei.

Em caso de requisições, dúvidas, sugestões ou reclamações relacionadas ao tratamento de dados pessoais por parte da ASSOCIAÇÃO WeONG, pedimos gentilmente que entre em contato pelo e-mail [fundacaoweong@gmail.com](mailto:fundacaoweong@gmail.com).

Quando o Titular entrar em contato com a ASSOCIAÇÃO WeONG para o exercício de seus direitos, a instituição poderá tratar dados pessoais para fins de confirmação da identidade do titular de dados, na medida do seu interesse e nos limites da sua segurança.

### **ATUALIZAÇÕES**

Esta Política de Privacidade está sujeita a eventuais atualizações, a qualquer tempo e a critério da ASSOCIAÇÃO WeONG, sendo que tais alterações terão aplicação e vigência imediata.

Atualizações podem ocorrer, especialmente quando a Autoridade Nacional de Proteção de Dados (ANPD) emitir novas diretivas sobre os procedimentos descritos neste documento.

Esta é a primeira versão da Política de Privacidade da ASSOCIAÇÃO WeONG.