

APLICAÇÃO WEB PARA AUXÍLIO NA DIVULGAÇÃO DE CASOS DE DESAPARECIMENTOS NO BRASIL

Luiz Henrique Saydt¹,
Kevyn Phillipe Gusmão²,
Fabio Goldner².

¹ Discente do Curso de Engenharia de Computação do Centro Universitário MULTIVIX Vitória

² Docentes, Mestres, do Centro Universitário MULTIVIX Vitória

RESUMO

O tema de desaparecimentos de pessoas é pouco abordado no Brasil, os materiais disponíveis acerca do tema são escassos e não tem o alcance que deveriam e, a cada ano, surgem no Brasil, milhares de novos casos de desaparecidos, dos quais menos da metade são solucionados. Como possível forma de ampliar o alcance da informação sobre os desaparecimentos no Brasil, de modo que mais pessoas possam se ajudar, esse trabalho propõe o desenvolvimento de um protótipo de uma aplicação WEB de fácil acesso para toda população, onde seria possível cadastrar eventos de desaparecimentos e aqueles que foram afetados, incluindo suas características e foto, permitindo assim, que mais pessoas possam estar envolvidas ao saberem onde, quando e como ocorreu determinado evento.

PALAVRAS-CHAVE

Desaparecimentos; Aplicação Web; Divulgação; Acessibilidade; Tecnologia.

ABSTRACT

The issue of missing persons is rarely addressed in Brazil. The available materials on the subject are scarce and lack the reach they should have, and each year, thousands of new cases of missing persons emerge in Brazil, with less than half being resolved. As a possible way to expand the dissemination of information about disappearances in Brazil, allowing more people to help each other, this work proposes the development of a prototype for an easily accessible web application for the entire population, where it would be possible to register disappearance events and those affected, including their characteristics and photos, thus enabling more people to get involved by knowing where, when, and how a particular event occurred.

KEYWORDS

Disappearances; Web Application; Dissemination; Accessibility; Technology.

INTRODUÇÃO

De acordo com o relatório “Ainda? Essa é a palavra que mais dói”, do Comitê Internacional da Cruz Vermelha (CICV), o Brasil, entre 2016 a 2019, registrou cerca de 80 mil casos de desaparecimentos por ano. Uma pesquisa do Fórum Brasileiro de Segurança Pública (FBSP), junto com o Instituto Datafolha, realizada em 2017, apontou que cerca de 17% dos brasileiros já declaram o desaparecimento de algum parente, amigo ou conhecido.

Segundo dados fornecidos pelo Anuário Brasileiro de Segurança Pública, observa-se que a quantidade de pessoas encontradas, em certos anos, não chega à metade dos desaparecidos. Em 2016, foram registradas 81.176 ocorrências de

desaparecimentos, porém, nesse mesmo ano, o registro de pessoas encontradas foi de apenas 28.121.

Além das consequências para a vítima, o desaparecimento pode causar um dano psicológico em familiares e amigos, às vezes irreparável, provocados pelo sentimento de incerteza da morte e a esperança de reencontrar a vítima com vida.

A dor da morte de um filho pode ser insuperável. Mas não saber se ele está vivo ou morto em decorrência de um desaparecimento e, portanto, não saber se algum dia ele será encontrado pode ser infinitamente mais angustiante. Muitas famílias de crianças e adolescentes desaparecidos parecem experimentar um luto sem um corpo mantendo a esperança de que um dia vão encontrar seus filhos (FIGARO-GARCIA, 2010, p. 40)

Diversas são as etapas a serem seguidas para se resolver um desaparecimento, um deles, citado pela Cartilha de Enfrentamento ao Desaparecimento, é a divulgação para a sociedade, onde é sugerido a divulgação em redes sociais. O que pode não ser muito efetivo, visto que as publicações podem não repercutir ou se limitar apenas a amigos e familiares.

O maior empecilho dessa etapa está no alcance da divulgação, principalmente em casos onde o desaparecido não possui muitos contatos próximos. Assim, esse trabalho tem como proposta o desenvolvimento de um protótipo de um sistema informatizado, que auxilie no cadastro e divulgação de desaparecidos, de forma que o mesmo proporcione fácil acesso e interatividade aos utilizadores, em outras palavras, a possibilidade de acessarem a ferramenta por qualquer dispositivo e que o mesmo seja de fácil utilização.

Para atender a esses requisitos, foi escolhida a plataforma WEB, que proporciona ao utilizador acessar determinada aplicação de qualquer lugar e de qualquer dispositivo, desde que esses possuam acesso à internet e um navegador WEB, como o Chrome, Firefox, Opera, entre outros. Dessa forma, seria possível utilizar o sistema em um computador *desktop*, *notebook* e até mesmo em *smartphones*.

2 REFERENCIAL TEORICO

2.1 Desaparecimento de Pessoas

O desaparecimento de uma pessoa é definido pela Cartilha de Enfrentamento ao Desaparecimento como “o sumiço repentino de alguém, sem aviso prévio a familiares ou a terceiros”, ainda segundo a cartilha, uma pessoa passa a ser considerada desaparecida quando “não pode ser localizada nos lugares que costuma frequentar,

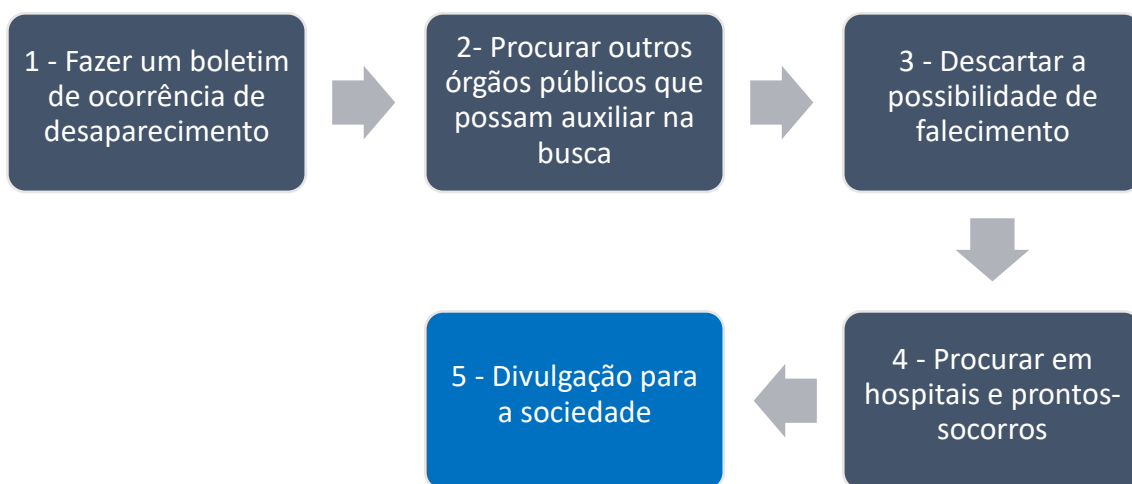
nem encontrada de qualquer outra forma.”, e a partir desse evento, a pessoa é considerada desaparecida de imediato, não sendo necessário aguardar qualquer período de tempo antes de tomar as devidas providências. Outros autores, como Dijaci David de Oliveira, utilizam o termo “desaparecimento civil”, definido por ele como

Uma pessoa que saiu de um ambiente de convivência familiar, ou de algum grupo de referência emocional-afetiva – como uma roda de amigos –, para realizar qualquer atividade cotidiana, não anunciou a sua intenção de partir daquele lugar e jamais retornou (OLIVEIRA, 2014, p. 30).

Quanto ao tipo de desaparecimento, a Cartilha de Enfrentamento ao Desaparecimento define três, sendo eles: Voluntário: Quando a pessoa se afasta por vontade própria e sem avisar. Involuntário: Quando a pessoa é afastada do cotidiano por um evento sobre o qual não tem controle, como um acidente, um problema de saúde, um desastre natural e etc. Forçado: quando outras pessoas provocam o afastamento, sem a concordância da pessoa. Como em um sequestro.

Independentemente da natureza do desaparecimento, Dijaci David de Oliveira aponta três problemas que devem ser resolvidos de imediato: “Saber o que ocorreu, saber o que fazer, saber a quem procurar”. A Cartilha de Enfrentamento ao Desaparecimento sugere cinco etapas a serem seguidas para maior eficiência no encontro ao desaparecido, as quais estão descritas na imagem a seguir.

Figura 1 - Passo a passo para otimizar as buscas por desaparecidos.



Fonte: Adaptado pelo autor.

O foco desse trabalho é propor uma ferramenta que possa auxiliar o cumprimento da 5ª etapa, a “Divulgação para a sociedade”. A cartilha explica que no processo de busca de um desaparecido, é muito importante avisar o máximo de pessoas possível, principalmente nos locais que ele(a) costumava frequentar. Vale a pena conversar com

familiares, amigos, vizinhos, colegas de trabalho e qualquer conhecido que possa auxiliar na busca. Você também pode fazer postagens nas redes sociais, com foto e dados gerais do desaparecido.

2.2 Desenvolvimento WEB

Segundo Conolly (2018), o desenvolvimento web pode ser entendido como um ecossistema, construído com base em tecnologias já existentes, como a internet, mas que também proporciona a criação de novas tecnologias e protocolos, como o HTTP, HTML e o Javascript. O autor também apresenta que, quando comparadas com aplicações para desktops, as aplicações WEB oferecem certas vantagens, sendo algumas: Acessibilidade: Podem ser acessadas por qualquer computador com acesso à internet; Usabilidade: Se bem desenvolvida, podem funcionar em diferentes sistemas operacionais, como Windows, Mac e Linux e também em diferentes navegadores, como Chrome, Firefox e Safari; Manutenção: Devido ao sistema estar integralmente em servidores, toda manutenção realizada irá ser aplicada para todos os utilizadores da aplicação, necessitando apenas o recarregamento da página no navegador.

Outra vantagem apresentada por ele que se destaca é, em tradução própria, “O armazenamento centralizado no servidor significa menos preocupações de segurança sobre o armazenamento local (o que é importante para informações confidenciais, como dados da área da saúde)”. Essas vantagens foram cruciais para que a escolha do tipo de protótipo que se busca construir fosse WEB, além disso, com base em pesquisas prévias, pretende-se utilizar o *framework* Vue.js para o *frontend* da aplicação, junto às tecnologias NodeJS e MySQL, que compõe o *backend*, tecnologias essas que serão descritas em maiores detalhes nos tópicos a seguir.

2.3 Frameworks

Conforme as tecnologias avançam, as aplicações WEB se tornam maiores, mais complexas e a dificuldade para mantê-las aumenta, assim, surge a necessidade por ferramentas para auxiliar o desenvolvedor na sua construção.

Uma das principais ferramentas utilizadas são os *frameworks* que, de acordo com Conolly (2018), em tradução própria, “Uma biblioteca ou framework é um ambiente de software reutilizável que você pode empregar em sua própria aplicação, a fim de prover funcionalidades específicas, que aprimorarão a velocidade e confiabilidade do

processo de desenvolvimento”. Já Wohlgethan (2018), diz que, em tradução própria, “Ao contrário das bibliotecas, frameworks oferecem de fato um conjunto completo de funções úteis e assume responsabilidade por decisões que, de outra forma, o desenvolvedor teria que tomar antes de realmente escrever o código do aplicativo. ”

Ambos os autores concordam que frameworks, quando bem utilizados e selecionados, visto que há uma quantidade vasta deles para diversas áreas do desenvolvimento, oferecem ao desenvolvedor agilidade e praticidade nas tarefas do *backend* e *frontend*, que basicamente, são as duas áreas que compõe o desenvolvimento WEB (LUNA, 2018).

2.4 Backend e frontend

Abdullah (2014) define *frontend* como, em tradução própria, “a parte que o usuário vê e interage, como menus, formulários de contato, etc. [...] Todos sendo controlados pelo navegador”, ou seja, é a parte da aplicação responsável pelo contato do usuário com o sistema, todos os componentes visíveis e interativos estão sendo controlados por tecnologias que são executadas no navegador, como o HTML, CSS e JavaScript.

O *backend* de uma aplicação, por outro lado, é definido por Luna (2018) como, em tradução própria, “Quem trabalha do lado do servidor, utilizando linguagens como PHP [...] interagindo com bases de dados, como MySQL, verificando sessões de usuários e montando uma página no servidor”. O usuário final não possui contato direto com o *backend*, sendo assim, ideal para realizar tarefas como a comunicação com o banco de dados e rotinas de autenticação.

2.5 NodeJS e JavaScript

“JavaScript é uma linguagem de programação leve, interpretada e orientada a objetos, conhecida como a linguagem de *scripting* para páginas Web.” (MDN Web Docs), embora nos últimos anos tenham surgido tecnologias para a utilização do JavaScript no *backend*, seu uso atualmente é predominantemente no lado do cliente, isto é, *frontend*. Essa linguagem é utilizada em aplicações WEB para tornar uma página mais interativa e reativa às ações do usuário, Conolly (2018) diz que, em tradução própria JavaScript pode ser usado para acessar programaticamente e manipular dinamicamente qualquer aspecto da aparência ou conteúdo do documento HTML. Ele pode ser usado para animar, mover, fazer a transição, ocultar e carregar conteúdo em partes de uma página, em vez de atualizar uma página inteira do servidor.

Junto ao crescimento da linguagem também surgiram diversas bibliotecas e *frameworks* largamente utilizadas, desde ferramentas que auxiliam o desenvolvedor a se comunicar de forma assíncrona com APIs, como o Ajax, que permite aos elementos do *frontend* a serem atualizados sem necessidade de recarregar a página (MDN Web Docs), até *frameworks* maiores, voltados para o fornecimento de um ambiente completo de desenvolvimento de interfaces, sendo os mais conhecidos, React.js, do Facebook, Angular.js, liderado pela Google, além do Vue.js, criado e mantido por desenvolvedores independentes.

JavaScript foi, por vários anos, uma linguagem de programação que funcionava de forma exclusiva nos navegadores, isto é, no *frontend* das aplicações WEB. Porém, em 2009, era lançado o *software* Node.js, ou apenas Node, o qual utiliza o interpretador JavaScript V8, da Google, porém, fora do ambiente do navegador. Agora era possível a execução de JavaScript diretamente em uma máquina que possuía o Node.js. Em seu site oficial, Node é descrito como um ambiente de tempo de execução JavaScript, construído sobre a *engine* JavaScript V8, do Chrome, assíncrono, orientado a eventos, como projeto para construir aplicações escalonáveis de rede.

Mike Cantelon (2014) cita alguns benefícios da utilização do JavaScript ao lado do servidor, sendo alguns deles: Os desenvolvedores podem escrever aplicações para WEB em apenas uma linguagem de programação, permitindo assim, compartilhar os códigos entre cliente e servidor. Node usa uma máquina virtual (V8) que segue os padrões ECMAScript, dessa forma ambos o cliente e servidor, poderão utilizar novos recursos do JavaScript conforme eles são disponibilizados. JSON é um formato de escrita de dados muito popular, além de ser nativo do JavaScript, o que torna a comunicação entre o *frontend* e *backend* mais simples.

2.6 Vue.js

“Vue.js é uma biblioteca JavaScript leve, criada por Evan You. Comparado ao React, o Vue coloca mais ênfase na experiência do usuário, tornando-o fácil de entender, se o usuário souber o básico: HTML, JavaScript e CSS.” (Saks, 2019), quando comparado a outros projetos, Vue apresenta uma rápida curva de aprendizado, e devido a suas características simplistas, é ideal para projetos menores, mas, ainda assim, sua escalabilidade não deve ser desprezada, visto que há grandes empresas que já utilizam essa tecnologia, como a Alibaba, Nintendo e Xiaomi.

Embora alguns autores recomendam o *framework* que possua as melhores características de desempenho e usabilidades, outros, como Freeman (2018), recomendam apenas que se escolha aquele o qual o usuário mais se familiarizar, pois, embora cada ferramenta tenha suas vantagens e desvantagens, ambas poderão produzir resultados parecidos, a depender do nível de familiaridade do desenvolvedor com a ferramenta.

Uma aplicação Vue se baseia no uso de componentes reutilizáveis chamados de instâncias Vue (Saks, 2019), cada componente é responsável por uma pequena parte da aplicação, seja um menu, uma barra de navegação ou até mesmo um único botão pode fazer parte de um componente individual. Outra característica dominante é a chamada “*data binding*”, que permitem ao desenvolvedor manipular facilmente e de forma reativa os objetos do HTML da página, acelerando e simplificando o desenvolvimento do *frontend* da aplicação.

2.7 Nuxt.js

Embora o Vue.js seja um *framework* repleto de recursos, ainda há diversos outros não presentes, como por exemplo, uma estrutura de pastas padrão, que não está presente no *framework* e cabe ao desenvolvedor organizar arquivo à arquivo, além disso, aplicações Vue.js não podem ser construídas em modo SSR (*server-side rendering*, ou renderização no lado do servidor), esse recurso é definido como:

o processo de pegar todos os JavaScript e todos os CSS de um site que, geralmente é carregado no browser (client-side), e renderizá-los como estático do lado do servidor. Com isso podemos obter um site com um tempo de carregamento reduzido e totalmente indexável por SEO's (ROCHA, 2018, p. 5).

A partir desses, e outros problemas encontrados ao se desenvolver uma aplicação com Vue.js, surgiu o Nuxt.js, que é descrito por Lau Tiam Kok (2020), em tradução própria, como “Nuxt é um *framework* de desenvolvimento web Node.js de nível superior para a criação de aplicativos Vue que podem ser desenvolvidos e implantados em dois modos diferentes: universal (SSR) ou *single-page application* (SPA).”

Nuxt também fornece diversos outros recursos, como *middlewares*, *layouts*, módulos e *plugins*, além de contar com uma estrutura de pastas padrão, que não só garante mais organização ao projeto, como também fornecem diversas

funcionalidades, por exemplo, ao ser criado um novo arquivo no diretório 'pages', Nuxt automaticamente realiza o roteamento para essa nova página do projeto.

2.8 TailwindCSS

Desenvolver toda a parte visual de uma aplicação pode ser uma tarefa muito trabalhosa e repetitiva. É necessário adicionar até mesmo dezenas de estilos CSS para cada item, seja um botão, um menu ou uma tabela, de forma que estes possam adquirir as características visuais desejadas, como tamanho, cores e responsividade. Os *frameworks* CSS buscam trazer ao desenvolvedor maior facilidade na estilização das páginas WEB, geralmente por meio de classes CSS que trazem diversos estilos prontos. Alguns *frameworks* oferecem classes que geram componentes completos, como o Bootstrap e o Vuetify, porém, com a desvantagem de possuírem pouca customização. Por outro lado, tem-se o TailwindCSS:

Um *framework* CSS de baixo nível, baseado em utilidade e altamente customizável [...] Tailwind não é um framework que possui componentes pré-fabricados ou predefinidos, mas possui extensos recursos de classes de utilidades e conjuntos que permitem que as pessoas criem qualquer projeto que desejem ou construam seus próprios componentes (NGUYEN, 2019, p. 45).

Na figura 2 podemos ver o exemplo de um botão HTML sem nenhum estilo, já na figura 3 temos um botão com estilo aplicado utilizando apenas classes do TailwindCSS, para alterar suas cores, dimensões e bordas.

Figura 2 – Criação de um botão HTML sem estilos



Fonte: Produzido pelo autor.

Figura 3 – Criação de um botão HTML com classes do TailwindCSS



Fonte: Produzido pelo autor.

Outra característica importante do *framework* é a facilidade para criar designs responsivos, ou seja, páginas que se adaptam ao tamanho da tela ao qual estão localizadas, seja em um computador ou um dispositivo celular. O tamanho dos itens, sua disposição, quantidade e outras características se ajustam de forma automática para facilitar a sua visualização em outros tamanhos de tela. Como um dos objetivos do protótipo que se deseja criar para esse trabalho é a acessibilidade, é importante que a aplicação possa ser facilmente utilizada por usuários de qualquer dispositivo.

3 METODOLOGIA

Para realização deste trabalho foi feita uma pesquisa aplicada, com abordagem quantitativa e qualitativa, de cunho exploratório, que tem como finalidade a coleta de dados por meio de levantamentos bibliográficos e documentais, entrevistas e estudos de caso, a fim de se aprofundar em um tema, muitas vezes pouco explorado.

Ainda segundo Gil (2008), a pesquisa aplicada tem como característica fundamental o interesse na aplicação, utilização e consequências práticas dos conhecimentos. Sua preocupação está menos voltada para o desenvolvimento de teorias de valor universal que para a aplicação imediata numa realidade circunstancial.

De forma a se compreender melhor o problema de desaparecimentos de pessoas, realizou-se uma pesquisa bibliográfica quantitativa, buscando-se dados estatísticos que demonstram o alcance do problema na sociedade, e também, realizou-se uma pesquisa qualitativa, com o intuito de se entender a gravidade do problema e como é possível minimizá-lo ou resolvê-lo.

A partir dos conhecimentos adquiridos com as pesquisas descritas, percebeu-se que um dos problemas a serem resolvidos era a divulgação dos casos de desaparecimento para a sociedade, de forma que, naturalmente, mais pessoas estivessem envolvidas e pudessem ajudar.

Assim, propôs-se o desenvolvimento de um protótipo de uma aplicação WEB, com o intuito de fornecer um ambiente acessível, onde poderiam ser registrados e divulgados, casos de desaparecimentos de pessoas, a fim de trazer maior visibilidade para o problema.

Para o desenvolvimento do sistema foi utilizado um computador desktop com as especificações básicas definidas a seguir: Processador: AMD Ryzen 5 2600; Memória RAM: 32GB DDR4; Armazenamento: SDD 256 GB.

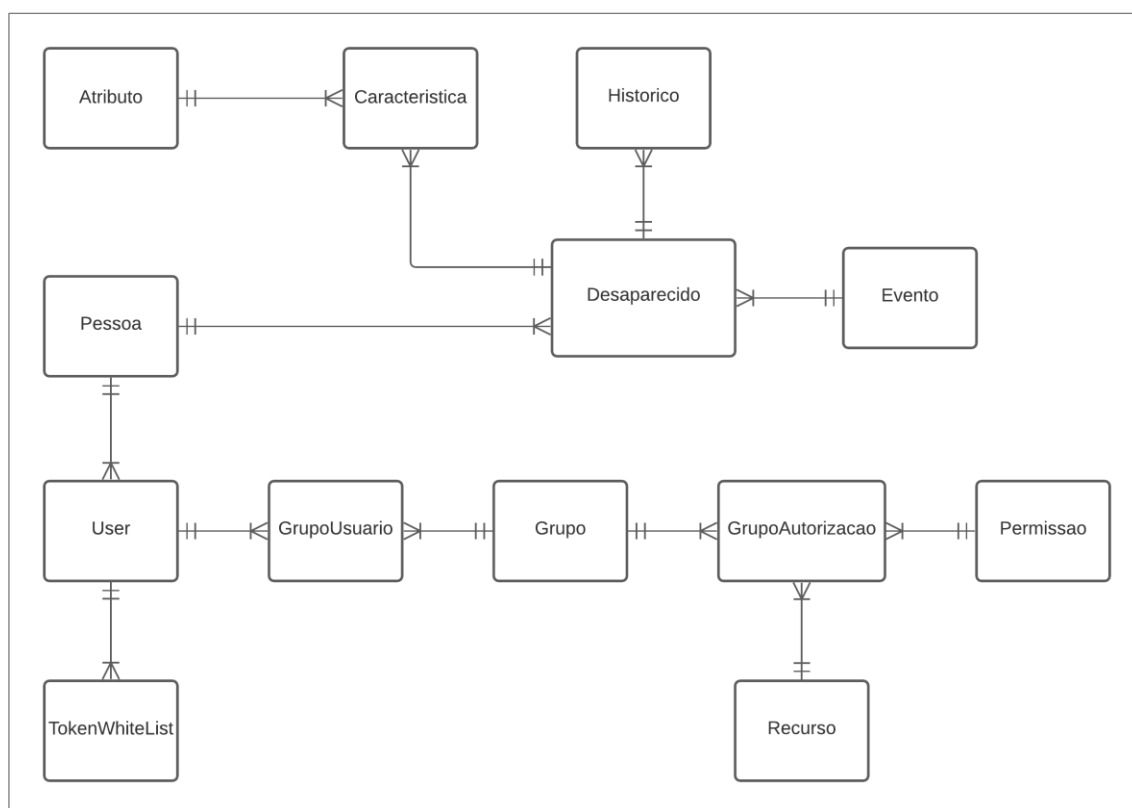
O *software* IDE utilizado foi o Visual Studio Code (v. 1.61). O gerenciador de banco de dados utilizado foi o MySQL, em conjunto com a ferramenta MySQL Workbench (v. 8.0) para gerenciamento dos dados e geração do modelo de entidade e relacionamento.

O *frontend* do sistema foi acompanhado com os navegadores Opera GX e Google Chrome, já o *backend*, foi acompanhado com o *software* Postman, para documentação e testes das rotas. Para o funcionamento das bibliotecas utilizadas pelo projeto, o ambiente de desenvolvimento contou com o *software* Node.js, em sua versão 14.

4 RESULTADOS E DISCUSSÃO

A primeira etapa do desenvolvimento do sistema foi a criação das entidades e seus relacionamentos no banco de dados, isto é, como os diferentes elementos (pessoa, usuário, etc...) interagem entre si. O banco de dados foi construído a partir do ORM (Mapeamento objeto-relacional) Prisma.js, onde a partir da definição de um *schema*, utilizando uma linguagem própria da biblioteca, é gerado de fato as entidades e relacionamentos no banco de dados, que podem ser observados no diagrama da Figura 4.

Figura 4 – Diagrama de entidade x relacionamento do projeto.



Fonte: Produzido pelo autor.

A próxima etapa do desenvolvimento se deu pela definição da API, a qual contém as rotas de acesso para a comunicação com o *frontend*, dois recursos importantes a serem implementados na API são a autenticação e a autorização. A autenticação é a etapa a qual deve-se confirmar a identidade do usuário, a pergunta que se deve fazer nessa etapa é “Quem é você? ”. Já na autorização, sabe-se quem é o usuário, mas deve ser verificado quais são os direitos de acesso dele, nessa etapa, faz-se a pergunta “O que você quer acessar?” (CONOLLY, 2018).

4.1 Autenticação

Para esse projeto foi escolhido o método de autenticação por tokens, utilizando o JSON Web Tokens, que segue os padrões da RFC 7519. Com esse método, a partir do momento em que usuário fornece suas credenciais de acesso e elas são validadas, o servidor criará um objeto do tipo JSON contendo os dados únicos que identificam esse usuário, além de uma data de expiração para invalidação do mesmo, esse objeto é então codificado com uma chave conhecida apenas pelo servidor e devolvido ao

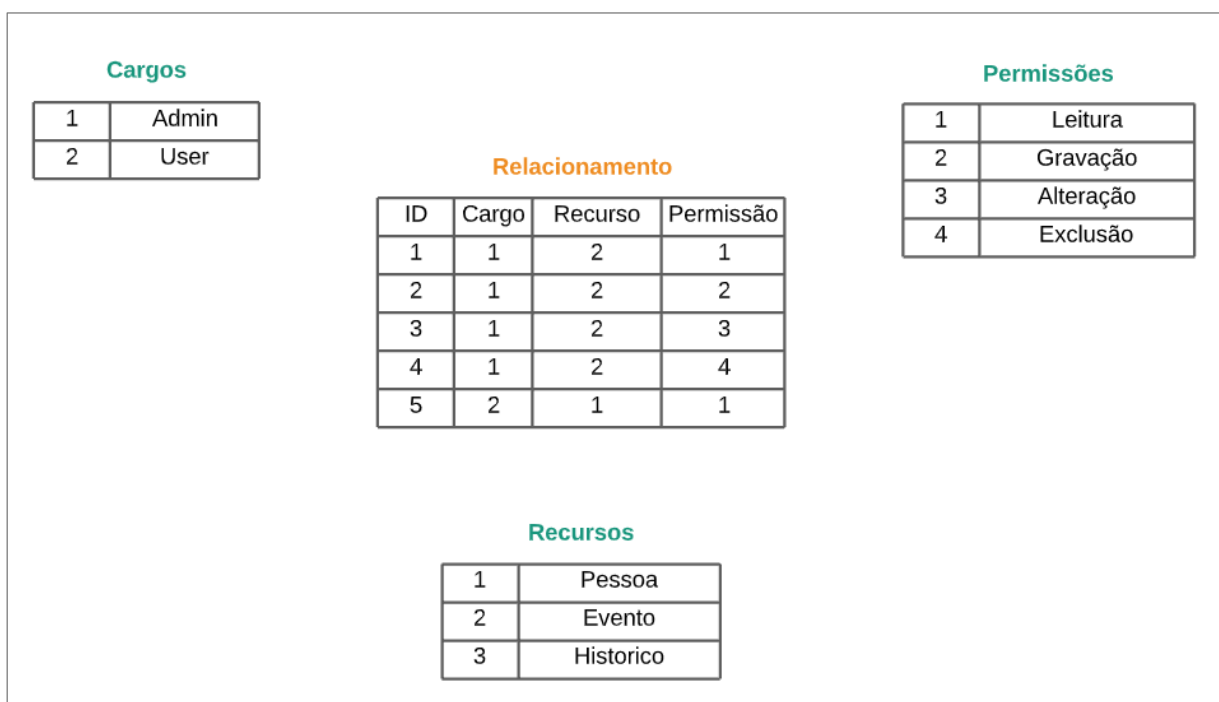
cliente, que a partir de agora, irá fornecer o token por meio do cabeçalho das próximas requisições realizadas.

De forma a garantir ainda mais a autenticidade, utilizou-se uma tabela no banco de dados ('tokenwhitelist'), para serem registrados todos os tokens que foram gerados pelo servidor, contendo a data de expiração e o identificador único do usuário, dessa forma, se torna mais difícil para um usuário mal intencionado se apropriar do token de outro usuário para acessar o sistema, visto que seria necessário o acesso ao banco de dados da aplicação para a inserção do token.

4.2 Autorização

De forma a se ter um método de autorização robusto, organizado e escalonável, utilizou-se o modelo de cargos e permissões, onde as rotas da API são divididas em recursos, por exemplo, o recurso pessoa, ou o recurso desaparecido, esses recursos são associados no banco de dados com os cargos que podem acessá-lo, como 'admin', 'user', entre outros. Além disso, os cargos e recursos são também associados com as permissões de acesso, que podem ser de escrita, gravação, exclusão e etc. A Figura 5 demonstra o relacionamento entre as entidades do método apresentado, com valores de exemplo.

Figura 5 - Relação entre tabelas para implementação do método de autorização.



Fonte: Produzido pelo autor

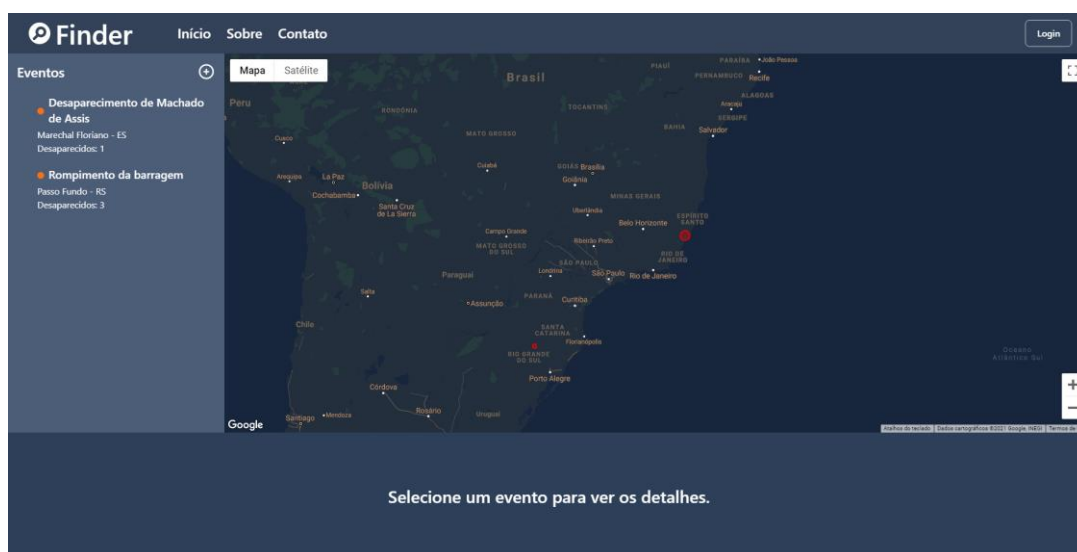
As permissões aos recursos são associadas à grupos, e não aos usuários diretamente. Caso um usuário necessite de certa permissão, é necessário adicioná-lo a um grupo correspondente que a possui. Dessa forma, as autorizações se tornam mais seguras e robustas. Com a lógica da autenticação e autorização concluídas, criou-se um *middleware* para ser utilizado pelas rotas da API, através da biblioteca *express.js*, de forma que, para toda requisição na rota, o *middleware* intercepta a comunicação e verifica a autenticidade e autoridade do cliente.

A API se comunica com o cliente através da notação JSON, tanto no recebimento de dados, no corpo da requisição, quanto no envio deles de volta ao cliente. Devido ao fato de que o *frontend* e o *backend* são ambos desenvolvidos em Node, utilizar JSON se torna especialmente simples, pois ele é a representação de um objeto JavaScript e, portanto, não é necessário fazer conversões entre os tipos de dados trocados pelo cliente e servidor.

4.3 Frontend

Ao acessar a página WEB da aplicação pela primeira vez, o usuário irá se deparar com a tela de eventos, visualizando a ocorrência deles em um mapa, podendo ver os detalhes e a lista de pessoas desaparecidas. Um evento é todo tipo de ocorrência de desaparecimento de uma ou mais pessoas, como o evento de alguém sair de casa e não retornar, ou um evento de um desastre natural como uma enchente, terremoto e etc.

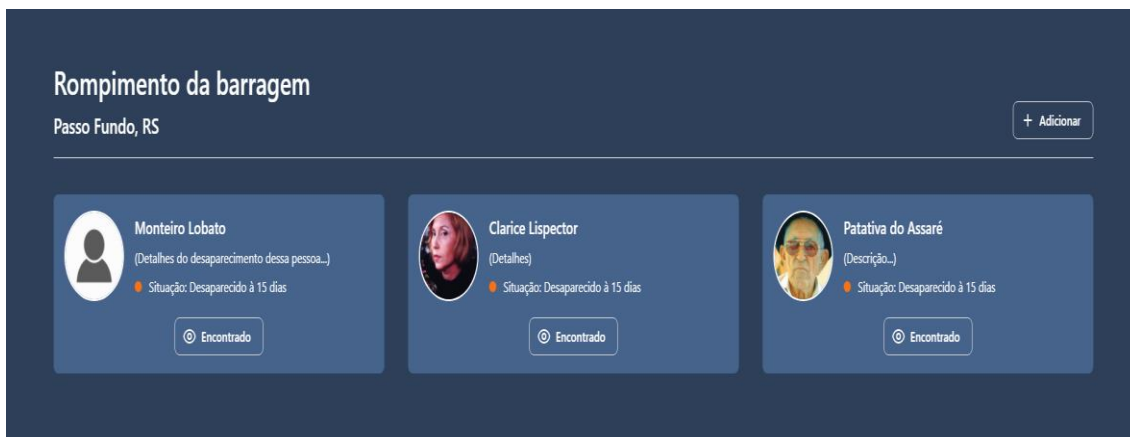
Figura 6 – Página inicial da aplicação.



Fonte: Produzido pelo autor.

Para visualizar um evento específico, basta clicar no mesmo na lista de eventos, assim, será exibido a lista de indivíduos considerados desaparecidos, contendo seus nomes, descrição, imagem, a quanto tempo estão desaparecidos e um botão para informar que a pessoa foi encontrada.

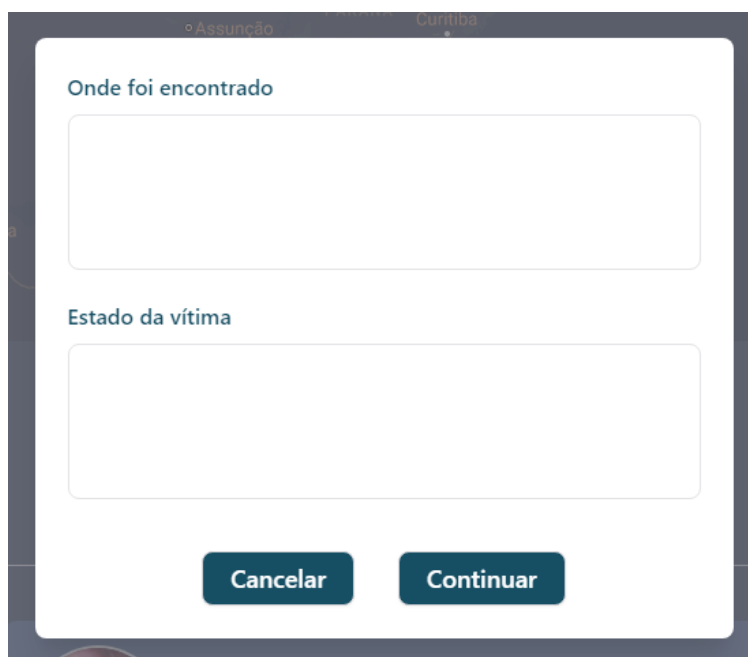
Figura 7 – Área de detalhes do evento selecionado.



Fonte: Produzido pelo autor.

Ao clicar na opção, lhe é exibido um modal, isto é, uma janela flutuante, sobrepondo às demais, onde o usuário deverá informar onde a vítima foi encontrada e qual seu estado físico ou mental. Após todos os desaparecidos de um evento serem encontrados, ele será considerado finalizado.

Figura 8 – Janela para relatar encontro da vítima selecionada.



Fonte: Produzido pelo autor.

Para que possa criar novos eventos, o usuário deverá realizar seu login, caso não possua um, poderá criá-lo. O registro requer o preenchimento de alguns dados do usuário, sendo eles, nome completo, *e-mail*, data de nascimento, nome de usuário e senha.

Figura 9 – Janela de cadastro

The image shows two side-by-side registration forms. The left form contains the following fields: 'Nome Completo', 'E-mail', 'Data de Nascimento' (with a calendar icon), 'Nome de Usuário', 'Senha', and 'Repita a Senha'. The right form contains: 'Nome Completo', 'Data de Nascimento' (with a calendar icon), 'Nome de Usuário', 'Senha', and 'Repita a Senha'. Both forms have 'Cancelar' and 'Continuar' buttons at the bottom.

Fonte: Produzido pelo autor.

Caso os dados estejam corretamente preenchidos, o usuário será levado para a próxima janela, onde poderá escolher ou não uma foto de perfil.

Figura 10 – Janela de seleção de foto de perfil.

The image shows a profile photo selection screen titled 'Escolha sua foto de perfil'. It features a central image of a person with a crop overlay. At the bottom are 'Remover Foto' and 'Selecionar Foto' buttons. A right arrow icon is in the top right corner.

Fonte: Produzido pelo autor.

4.3.1 Criação de Eventos

Na lista de eventos, após realizado seu login, o usuário poderá criar novos eventos, clicando no botão com sinal de mais ('+'). Desta forma, lhe será exibido um formulário a ser preenchido com dados do evento.

Figura 11 – Formulário de criação de evento de desaparecimento



Nome do evento *

Ex: Desaparecimento do ...

Descrição do evento *

Data do ocorrido *

dd/mm/aaaa 

Local do corrido

Cidade *

Bairro

Logradouro

Selecione o Estado *

Clique no mapa para selecionar a área aproximada, em seguida, escolha o alcance do evento

Fonte: Produzido pelo autor.

O usuário deverá fornecer um nome simples e explicativo para o evento, além de uma descrição do que ocorreu, em seguida, a data do ocorrido, e então, deverão ser fornecidas as informações do local do ocorrido, nessa etapa, o usuário deve clicar no mapa disponível na página para marcar o local aproximado do evento, além de selecionar o raio de alcance do mesmo.

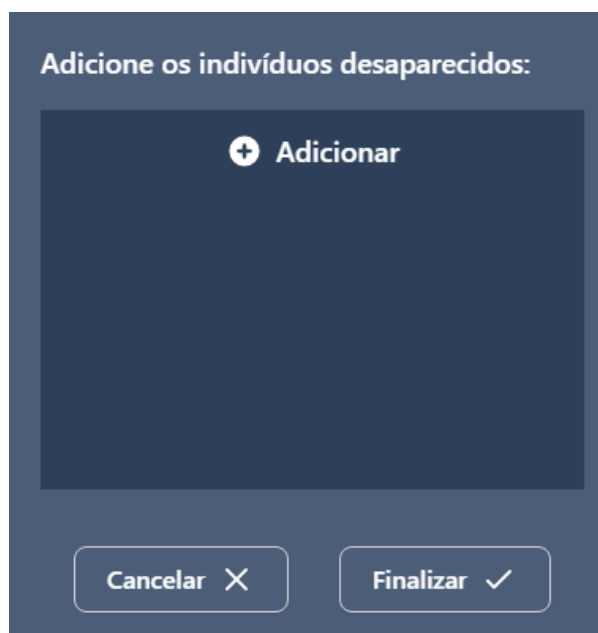
Figura 12 – Seleção da localidade e área de alcance do evento



Fonte: Produzido pelo autor.

Com as informações devidamente preenchidas, no próximo menu o usuário deverá inserir as pessoas consideradas desaparecidas devido a esse evento.


Figura 13 – Janela de listagem dos desaparecidos.



Fonte: Produzido pelo autor.

Clicando no botão 'adicionar' será aberto um modal para adicionar a pessoa, onde deverá ser informado seu nome, foto e detalhes que possam facilitar sua localização, como características físicas, onde foi visto por último, os lugares que costuma frequentar, entre outras informações que julgar relevante para facilitar a localização do mesmo.

Figura 14 – Formulário para adição de desaparecido.



O formulário para adição de desaparecido apresenta os seguintes elementos:

- Um campo de texto rotulado "Nome".
- Um campo de texto rotulado "Detalhes da pessoa".
- Um ícone de perfil de usuário (silhueta de cabeça e ombros).
- Dois botões de ação na base: "Cancelar X" e "Continuar →".

Fonte: Produzido pelo autor.

Ao terminar de adicionar todos os desaparecidos, o evento poderá ser finalizado e será exibido na lista de eventos, onde todos poderão ver os detalhes e informar o encontro de um desaparecido. Quando todos os desaparecidos de um evento forem encontrados, ele será considerado concluído.

5. CONSIDERAÇÕES FINAIS

O presente estudo buscou aprimorar o alcance das informações acerca dos eventos de desaparecimentos de pessoas, visto que, os métodos atuais de divulgação são escassos e comumente se limitam às redes sociais e outros meios jornalísticos, onde por vezes, passam despercebidos, não recebendo a devida atenção. Assim, desenvolveu-se um protótipo de uma aplicação WEB completa, onde é possível cadastrar ocorrências de desaparecimentos civis, informando detalhes do evento e localidade, além de informações da vítima, a qual pode ser adicionado detalhes e uma foto em seu cadastro para auxiliar no reconhecimento. A interface do projeto também fornece um mapa geográfico, onde é possível visualizar melhor as ocorrências em uma região alvo, por exemplo, a cidade a qual o usuário encontra-se.

Desta forma, tem-se uma plataforma digital, onde a qualquer momento, todo cidadão poderá consultar os diversos eventos que ocorrerem em sua proximidade, podendo ter conhecimento de ocorrências que, fora da aplicação, poderiam nunca tê-lo alcançado. O protótipo apresentado, poderia ser aprimorado, em trabalhos futuros, através da inclusão de um sistema de notificações, onde os usuários do sistema poderiam ser notificados, via e-mail por exemplo, a respeito de novas ocorrências de

desaparecimentos em sua região. Além disso, o projeto poderia ser adaptado para permitir o gerenciamento de não apenas desaparecidos civis, como também animais e veículos.

6. REFERÊNCIAS

ABDULLAH, Hanin M.; ZEKI, Ahmed M. **Frontend and backend web technologies in social networking sites: Facebook as an example**. In: 2014 3rd international conference on advanced computer science applications and technologies. IEEE, 2014. p. 85-89.

Ajax, Documentação. **MDN Web Docs**. 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/Guide/AJAX>>. Acesso em: 14 de nov. de 2021.

‘Ainda?’ Essa é a palavra que mais dói. **Comitê Internacional da Cruz Vermelha**, 06 de jul. de 2021. Disponível em: <<https://www.icrc.org/pt/publication/relatorio-ainda-essa-e-palavra-que-mais-doi>>. Acesso em: 14 de nov. de 2021.

Anuário Brasileiro de Segurança Pública 2017. **Fórum Brasileiro de Segurança Pública**, nov. 2017. Disponível em: <https://www.forumseguranca.org.br/wp-content/uploads/2017/12/ANUARIO_11_2017.pdf>. Acesso em: 14 de nov. de 2021.

CANTELON, Mike et al. **Node.js in Action**. Greenwich: Manning, 2014.

CONNOLLY, Randy; HOAR, Ricardo. **Fundamentals of web development**. 2. ed. Pearson Education, 2018.

DE OLIVEIRA, Dijaci David. **O desaparecimento de pessoas no Brasil**. Cânone Editora Ltda, 2014.

FREEMAN, Adam. **Pro Vue.js 2**. 1. ed. Apress, 2018.

FIGARO-GARCIA, Claudia. **Uma proposta de prática psicológica para casos de desaparecimento de crianças e adolescentes**. 2010. Tese de Doutorado. Universidade de São Paulo.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 6. ed. Editora Atlas SA, 2008.

KOK, Lau Tiam. **Hands-on Nuxt.js Web Development: Build universal and static-generated Vue.js applications using Nuxt.js**. Packt Publishing Ltd, 2020.

LUNA, Fernando; MILLAHUAL, Claudio Peña; IACONO, Matías. **PROGRAMACION WEB Full Stack 13-PHP: Desarrollo frontend y backend-Curso visual y práctico**. RedUsers, 2018.

NGUYEN, Linh. **Building e-commerce solutions with Woocommerce**. 2019. Tese de bacharelado. Centria University of Applied Sciences.

Prefeitura de São Paulo. **Cartilha de Enfrentamento ao Desaparecimento**. Ministério Público do Estado de São Paulo. São Paulo, 2016.

ROCHA, Fernando. O que é Server Side Rendering e como usar na prática. **Medium**, 14 de ago. de 2018. Disponível em: < <https://medium.com/techbloghotmart/o-que-é-server-side-rendering-e-como-usar-na-prática-a840d76a6dca>>. Acesso em: 14 de nov. de 2021

SAKS, Elar. **JavaScript Frameworks: Angular vs React vs Vue**. 2019. Tese de Bacharelado. University of Applied Sciences.

Sobre JavaScript. **MDN Web Docs**. 2021 Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript>. Acesso em: 14 de nov. de 2021.

WOHLGETHAN, Eric. **Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue. js**. 2018. Tese de Doutorado. Hochschule für Angewandte Wissenschaften Hamburg.