



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

FRANCISCO ANGELIM DE FIGUEIRÊDO FILHO

GAJA:

RESPONSABILIDADE SOCIAL É O QUE SE ESPERA DE TODOS

CAMPINA GRANDE - PB

2021

FRANCISCO ANGELIM

GAJA:

RESPONSABILIDADE SOCIAL É O QUE SE ESPERA DE TODOS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Carlos Eduardo Santos Pires

CAMPINA GRANDE - PB

2021



F475g Figueirêdo Filho, Francisco Angelim de.

GAJA: responsabilidade social é o que se espera de todos. / Francisco Angelim de Figueirêdo Filho. - 2021.

10 f.

Orientador: Prof. Dr. Carlos Eduardo Santos Pires.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Desenvolvimento de aplicação. 2. Desenvolvimento de software. 3. Trabalho voluntário. 4. GAJA - sistema de trabalho voluntário. 5. Responsabilidade social. 6. Usabilidade de sistema. 7. Atividades voluntárias - cadastramento. 8. Questionário PSSUQ. 9. Post-Study System Usability Questionnaire. I. Pires, Carlos Eduardo Santos. II. Título.

CDU:004.415.2(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

FRANCISCO ANGELIM DE FIGUEIRÊDO FILHO

GAJA:

responsabilidade social é o que se espera de todos

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Carlos Eduardo Santos Pires

Orientador – UASC/CEEI/UFCG

Professor Pedro Sergio Nicolletti

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 25 de Maio de 2021.

CAMPINA GRANDE - PB

RESUMO (ABSTRACT)¹

Trabalhos voluntários são atividades prestadas à sociedade sem recebimento de qualquer remuneração ou lucro. Este tipo de atividade ainda é pouco praticada pela população brasileira. Logo, uma aplicação (sistema) com o objetivo de facilitar e incentivar a busca destas atividades pode ser considerada uma solução para amenizar este problema. Neste trabalho, propomos uma aplicação na qual usuários podem cadastrar atividades voluntárias. Como resultado, essas atividades ficam disponíveis para que voluntários possam se engajar. As atividades podem ser encontradas mediante buscas (consultas) por trabalhos próximos à localização do voluntário e compatíveis com as suas habilidades e interesses. Por fim, realizamos um experimento com o intuito de analisar a usabilidade e qualidade da solução proposta. A partir do resultado deste, observamos um nível de satisfação positivo quanto ao uso da aplicação.

¹ Voluntary jobs are activities provided to society without receiving any remuneration or profit. This type of activity is still little practiced by the Brazilian population. Therefore, an application (system) with the objective of facilitating and encouraging the search for these activities can be considered a solution to alleviate this problem. In this work, we propose an application in which users can register voluntary activities. As a result, these activities are available for volunteers to engage with. Activities can be found through searches (consultations) for jobs close to the volunteer's location and compatible with their skills and interests. Finally, we carried out an experiment in order to analyze the usability and quality of the proposed solution. From the result of this, we observed a level of positive satisfaction with the use of the application.

GAJA: responsabilidade social é o que se espera de todos

Francisco Angelim de Figueirêdo Filho

Universidade Federal de Campina Grande
Campina Grande, Paraíba
francisco.filho@ccc.ufcg.edu.br

Carlos Eduardo Santos Pires

Universidade Federal de Campina Grande
Campina Grande, Paraíba
cesp@dsc.ufcg.edu.br

RESUMO

Trabalhos voluntários são atividades prestadas à sociedade sem recebimento de qualquer remuneração ou lucro. Este tipo de atividade ainda é pouco praticada pela população brasileira. Logo, uma aplicação (sistema) com o objetivo de facilitar e incentivar a busca destas atividades pode ser considerada uma solução para amenizar este problema. Neste trabalho, propomos uma aplicação na qual usuários podem cadastrar atividades voluntárias. Como resultado, essas atividades ficam disponíveis para que voluntários possam se engajar. As atividades podem ser encontradas mediante buscas (consultas) por trabalhos próximos à localização do voluntário e compatíveis com as suas habilidades e interesses. Por fim, realizamos um experimento com o intuito de analisar a usabilidade e qualidade da solução proposta. A partir do resultado deste, observamos um nível de satisfação positivo quanto ao uso da aplicação.

Palavras-chave

Trabalhos voluntários; sociedade; aplicação; população brasileira.

1. INTRODUÇÃO

O trabalho voluntário é uma forma de contribuir com a sociedade. Aqueles que participam não recebem um valor monetário em troca do seu tempo, porém estas atividades podem ter algum outro valor para aqueles que realizam.

A motivação é um processo psicológico complexo que resulta de uma interação entre o indivíduo e o ambiente que o rodeia (FERREIRA et. al, 2008). Logo, os benefícios associados ao voluntariado podem estar relacionados a diversos motivos como ao reconhecimento social ou ao próprio ego.

Existem dois tipos de voluntariado, o presencial e o digital. O voluntariado presencial pode ocorrer de forma parcial ou contínua. A primeira forma consiste em eventos pontuais, normalmente organizados em forma de mutirões como, por exemplo, mutirões de limpeza e saúde. Já a segunda forma, é necessária maior assiduidade do voluntário, pois as atividades são mais frequentes; por exemplo, de uma a duas vezes por semana.

O voluntariado digital normalmente requer menos tempo do voluntário e pode ser feito de várias maneiras como, por exemplo, através da divulgação de eventos em redes sociais.

Com o objetivo de valorizar e incentivar essas práticas, a ONU declarou o ano de 2001 como o "Ano Internacional do Voluntariado". Ainda assim, o voluntariado foi praticado por apenas 4,3% da população com 14 anos ou mais no país em 2018

[2]. Neste contexto, são necessários mecanismos para aumentar o voluntariado. Existem diversas organizações que realizam trabalhos voluntários no Brasil, porém não há uma plataforma que faça a unificação delas e permita que pessoas que nunca participaram deste tipo de atividade possam encontrar com facilidade um trabalho que os motivem a começar a participar. Portanto, uma aplicação que unifique as informações sobre os trabalhos voluntários disponíveis, seria uma solução para esse problema.

2. TRABALHOS RELACIONADOS

Existem algumas aplicações que possuem a premissa de conectar pessoas que desejam colaborar com quem necessita de colaboração; por exemplo, a ONGS Brasil[3].

Porém, estas e outras aplicações possuem um fluxo burocrático no cadastro de novos usuários e/ou eventos. Neste contexto, a presença de dificuldades para pessoas que utilizam sistemas online para começar no voluntariado, pode ser um fator desmotivante, que tem como consequência a perda de novos voluntários.

Por isso, um dos diferenciais da solução proposta, é a facilidade de engajamento dos novos usuários, além de oferecer uma interface simples e objetiva.

3. SOLUÇÃO PROPOSTA

Nesta seção será detalhada a solução para o problema apresentado anteriormente.

3.1 Visão geral

O GAJA é um sistema online que permite aos usuários a busca e divulgação de trabalhos voluntários de forma facilitada. O sistema traz vantagens tanto para quem está à procura de voluntários, quanto para quem tem interesse em realizar este tipo de trabalho.

A aplicação contém informações sobre pessoas e organizações que realizam trabalhos voluntários.

3.2 Funcionalidades

Na Figura 1, é possível observar a página inicial da aplicação. Nela, estão presentes todos os eventos cadastrados no GAJA, ordenados de forma que o primeiro evento seja aquele que está mais próximo de acontecer.

Os eventos que já ocorreram também são listados, porém eles só aparecem após a exibição do último evento ainda ativo. A data destes eventos é mostrada em cor cinza para indicar que o evento já aconteceu.

Também é possível pesquisar eventos pelo nome ou pela localização utilizando a barra de pesquisa abaixo da imagem principal.



Figura 1: Página inicial da aplicação

Participação em eventos

Na página de um evento, são descritas as informações necessárias para um usuário decidir, de acordo com seus interesses, se deseja ou não participar daquele evento. Caso opte por participar, ele deverá seguir as instruções descritas na página e escolher uma das vagas listadas para realizar sua inscrição, como é mostrado na Figura 2. Para facilitar o uso da aplicação, não é necessário estar cadastrado para engajar-se com os eventos criados pelos usuários. Entretanto, é necessário se cadastrar no sistema para oferecer trabalhos voluntários.

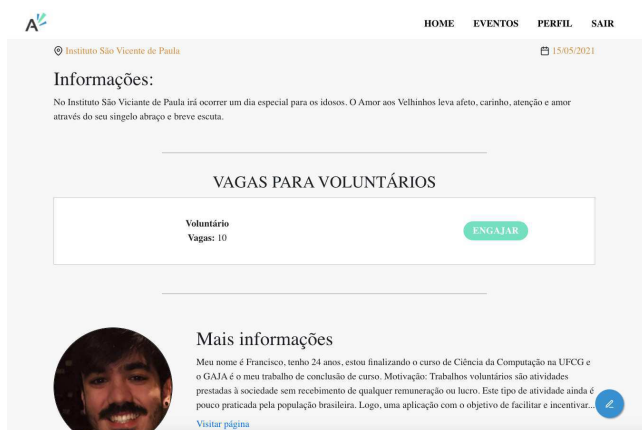


Figura 2: Página de um evento

Cadastro na aplicação

Qualquer usuário pode se cadastrar no GAJA para criar novos eventos voluntários. Além disso, ao finalizar o cadastro, o usuário terá uma página pública do seu perfil, onde outras pessoas poderão visualizar informações gerais e os eventos aos quais ele está relacionado, como mostrado na Figura 3.

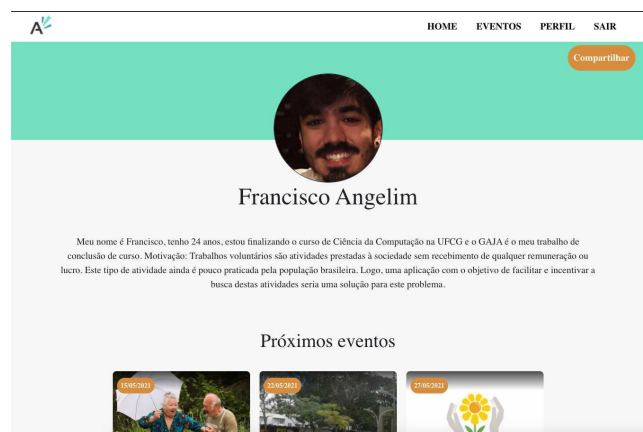


Figura 3: Perfil do usuário

Criação de novos eventos

Após cadastrar-se no sistema, o usuário torna-se apto a criar seu próprio evento. Para isto, basta conectar-se ao sistema utilizando as informações de conexão e acessar a opção "Eventos" na barra de navegação. Na página aberta, no canto inferior direito está o botão (+) que redireciona para a criação de um novo evento. Ainda nesta página, é possível visualizar duas listagens, a primeira se refere aos eventos cadastrados pelo usuário e a segunda contém os eventos que o usuário se engajou, como é possível observar na Figura 4.

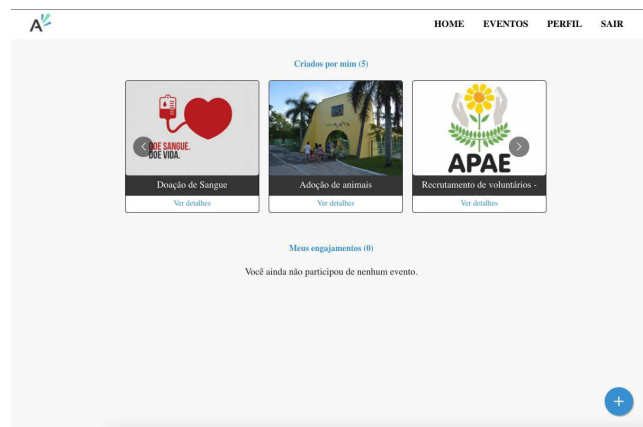


Figura 4: Eventos do usuário

Algumas informações são obrigatórias na criação de um evento, a citar:

- Tipo: Presencial ou Remoto;
- Nome do evento: título para identificação;
- Descrição: contém os detalhes do evento e informações relevantes para o engajamento do voluntário;
- Data: representa a data em que acontecerá o evento;
- Vagas disponíveis: opções de vagas para escolha do voluntário.

Após preencher estas informações, é possível ter uma pré-visualização do evento antes de o mesmo ser criado, utilizando a opção "Pré-visualizar" como mostra a Figura 5. Ao clicar em "Finalizar", o evento é criado e fará parte dos eventos presentes na página inicial do GAJA.

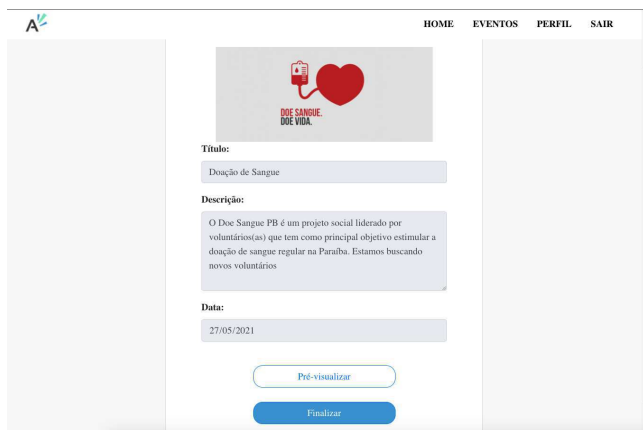


Figura 5: Tela final da criação de um evento

3.3 Arquitetura

Na arquitetura utilizada, o sistema foi dividido em dois projetos, um projeto contém o cliente (front-end) e o outro o servidor e banco de dados (back-end). Ambos os projetos estão disponíveis no GitHub².

O servidor contém as regras de negócio e é responsável por receber as requisições do cliente e realizar o tratamento dos dados como, por exemplo, formatações e validações. Já o cliente, é responsável por exibir a interface que o usuário irá interagir, para isso ele se comunica constantemente com o servidor.

A conexão entre esses projetos segue o modelo REST (*REpresentational State Transfer*). Neste modelo, a comunicação é *stateless*, ou seja, cada requisição é independente das outras pois não é armazenada nenhum tipo de informação entre elas. A comunicação é realizada utilizando o protocolo HTTP e os dados trocados são do tipo JSON (*JavaScript Object Notation*) que é um formato leve e de fácil leitura para humanos. Este fluxo pode ser visualizado na Figura 6.

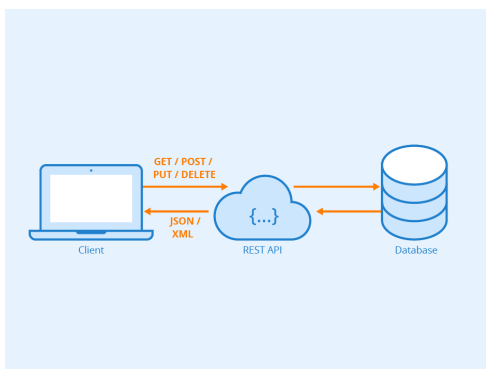


Figura 6: Fluxo da comunicação na arquitetura utilizada

3.4 Servidor

O servidor foi desenvolvido utilizando o ambiente de execução JavaScript chamado NodeJS [7] em conjunto com o framework Express.js.

Com o intuito de gerar coesão e organização ao código, a estrutura do servidor foi dividida em 4 grupos principais: *Controllers*,

² Repositórios:

<https://github.com/frnangelim/frontend-tcc>

<https://github.com/frnangelim/backend-tcc>

Services, *Models* e *Routes*. A Figura 7 mostra como foi organizado o código no lado do servidor.

As requisições que chegam ao servidor, são recebidas pelos *Routers*, estes irão direcionar a chamada para o *Controller* responsável. Em alguns casos, *Middlewares* podem ser utilizados para realizar tarefas como autenticação ou tratamento de arquivos.

A autenticação está sendo realizada através de JSON Web Tokens (JWT); caso ele esteja presente no cabeçalho *Authorization* da requisição, será realizada a lógica para recuperar o usuário a qual pertence aquele token.

Não foram inseridas validações levando em consideração o *role* do usuário, já que as funcionalidades desenvolvidas neste primeiro momento não necessitam dessa distinção, porém o sistema ficou pronto para a inserção deste critério.

Seguindo o fluxo, nos *Controllers*, são selecionados os dados relevantes para aquela chamada, para em seguida serem repassados aos *Services*. Estes serão responsáveis em manipular os dados e aplicar as regras de negócio da aplicação, além de fazer chamadas para realizar a persistência dos dados, quando necessário. Por fim, o *Service* retorna uma resposta para o *Controller* que irá finalizar a requisição enviando a resposta para o cliente.

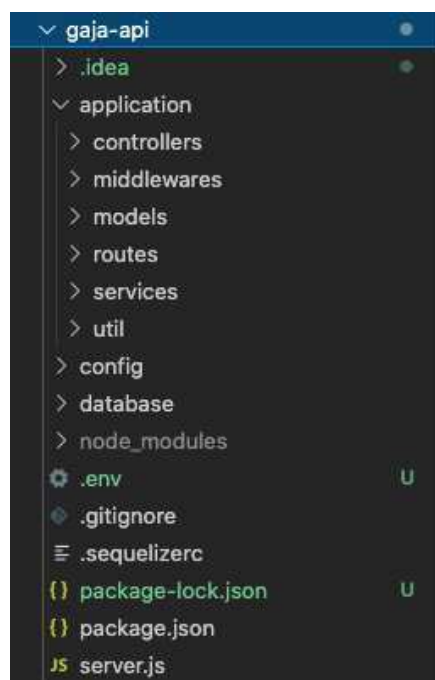


Figura 7: Estrutura de arquivos do servidor

3.5 Banco de dados

Para a persistência dos dados foi utilizado o SGBD (Sistema de Gerenciamento de Banco de Dados) MySQL [8]. Foi escolhido um banco de dados SQL devido à estrutura dos dados que serão armazenados ser bem definida e a necessidade de uma maior consistência nos dados.

Também foi utilizado o Sequelize ORM (*Object Relational Mapping*) para fazer uma abstração entre os modelos criados na aplicação e os modelos do banco de dados. Desta forma, não é necessário escrever consultas SQL manualmente, aumentando a produtividade do desenvolvimento.

Foram utilizadas as tabelas a seguir:

- *events*: contém os eventos do sistema;
- *event_slots*: representa as vagas para engajamento nos eventos;
- *event_enrollments*: armazena o engajamento dos usuários com os eventos;
- *users*: contém os usuários do sistema;
- *roles*: contém os papéis *roles* do sistema;
- *user_roles*: armazena os papéis dos usuários.

Estas tabelas e seus relacionamentos podem ser visualizados no diagrama presente na Figura 8.

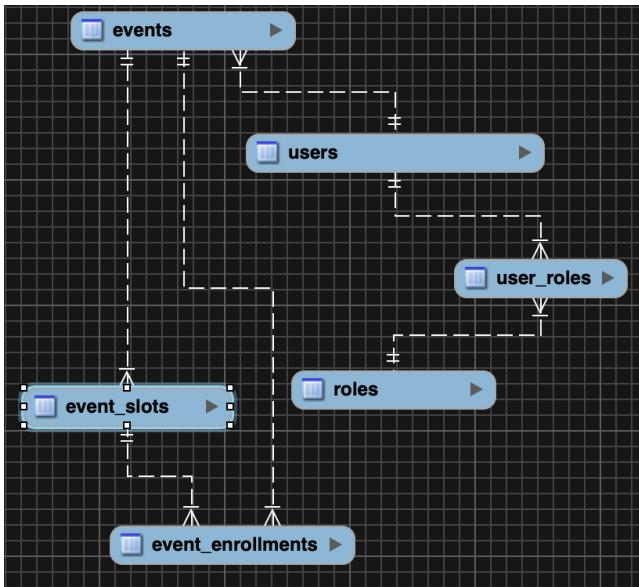


Figura 8: Diagrama Entidade Relacionamento do banco de dados do sistema

3.6 Cliente

O cliente foi desenvolvido utilizando a biblioteca Javascript chamada ReactJS [9]. A Figura 9 mostra como foi organizado o código no lado do cliente.

Para realizar o sistema de roteamento da aplicação, foi utilizada a biblioteca *react-router-dom*, que permite um roteamento dinâmico dentro da própria aplicação, sendo possível criar páginas públicas e privadas de forma facilitada.

Desta forma, criou-se um arquivo de roteamento que chama as páginas da aplicação de acordo com o endereço acessado. Tais componentes são chamados de *Pages*, eles utilizam outros componentes reutilizáveis que foram classificados como *Components*. Além disso, as páginas se comunicam com o servidor para obter os dados necessários. Para isso, foram utilizados os arquivos chamados de *Services*, os quais fazem requisições utilizando o protocolo HTTP e retornam os resultados para as *Pages*.

Com o intuito de auxiliar na estilização e reutilização de componentes, foi utilizada a biblioteca *Styled Components* [10] que viabiliza a escrita de CSS em Javascript. Também foi utilizada a biblioteca *reactstrap* [11] que fornece alguns componentes estilizados com *Bootstrap*.

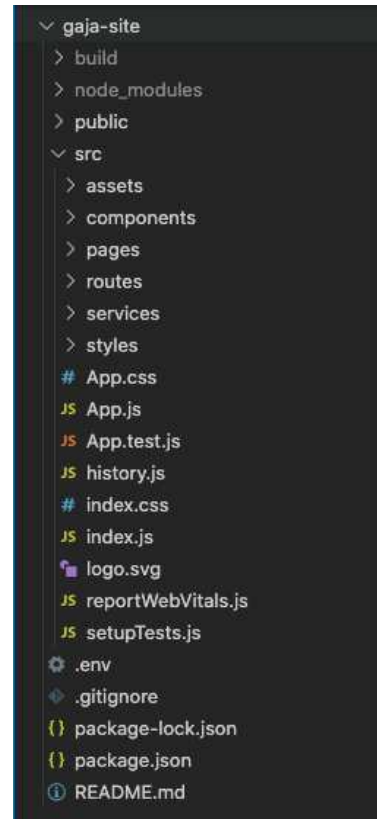


Figura 9: Estrutura de arquivos do cliente

3.7 Infraestrutura

Para disponibilizar o sistema aos usuários, foi feita a hospedagem da aplicação na plataforma *Heroku* [12]. Esta plataforma oferece serviços na nuvem no modelo PaaS (*Platform as a Service*).

Ao utilizar este serviço, foi obtida não só a infraestrutura necessária mas também o gerenciamento dela, retirando esta responsabilidade do desenvolvedor.

Tanto o cliente, quanto o servidor e banco de dados foram hospedados nesta mesma plataforma.

4. SISTEMA EM USO

Visto que a premissa da aplicação é facilitar e incentivar o engajamento de novos voluntários, é necessário que o sistema agrade aos usuários. Levando em consideração que a satisfação de um usuário é subjetiva, é necessário um método de avaliação quantitativo com foco na usabilidade do sistema e outros pontos relacionados.

Nesta seção, será descrito o processo de avaliação utilizado para medir o nível de satisfação dos usuários em relação à aplicação proposta.

4.1 Metodologia

Para fazer a avaliação do GAJA, o método escolhido para medir o nível de satisfação dos usuários foi a aplicação do questionário PSSUQ (*Post-Study System Usability Questionnaire*). As questões presentes nele abordam pontos importantes na utilização de um sistema como o GAJA.

O PSSUQ está disponível em diferentes versões, a versão que foi utilizada possui 16 questões (Figura 10), divididas em 3 categorias:

- *System Usefulness* (SYSUSE): questões 1 a 6, representam as afirmações relacionadas à facilidade de se utilizar o sistema;
- *Information Quality* (INFOQUAL): questões 7 a 12, representam as afirmações relacionadas à qualidade das informações do sistema e tratamento de erros;
- *Interface Quality* (INTERQUAL): questões 13 a 15, representam afirmações relacionadas à interface da aplicação.

A última questão (questão 16) não está contida nestas categorias, pois ela representa um indicador geral da satisfação do sistema.

1. No geral, estou satisfeito com a facilidade de uso deste sistema.
2. Foi simples usar este sistema.
3. Consegui completar as tarefas e cenários rapidamente usando este sistema.
4. Eu me senti confortável usando este sistema.
5. Foi fácil aprender a usar este sistema.
6. Acredito que poderia me tornar produtivo rapidamente usando este sistema.
7. O sistema deu mensagens de erro que me disseram claramente como resolver os problemas.
8. Sempre que cometia um erro ao usar o sistema, poderia recuperar fácil e rapidamente.
9. As informações (como ajuda online, mensagens na tela e outra documentação) fornecida com este sistema eram claras.
10. Foi fácil encontrar as informações de que precisava.
11. A informação foi eficaz para me ajudar a completar as tarefas e cenários.
12. A organização das informações nas telas do sistema foi clara.
13. A interface deste sistema era agradável.
14. Gostei de usar a interface deste sistema
15. Este sistema tem todas as funções e capacidades que eu espero ter.
16. No geral, estou satisfeito com este sistema.

Figura 10: Lista com os títulos das questões utilizadas no PSSUQ.

As respostas para todas essas questões são objetivas. As alternativas variam de 1 a 7, onde 1 significa concordância total com a afirmação e 7 representa discordância total.

O questionário foi criado utilizando a plataforma Google Forms. Além das questões do PSSUQ, ao final do formulário, foi adicionado um campo aberto para que os participantes pudessem deixar sugestões, melhorias ou qualquer outro tipo de comentário.

Para garantir uma maior acurácia na avaliação, os participantes foram orientados a engajar com pelo menos um evento e/ou a criar um novo evento, antes de responderem ao questionário.

4.2 Resultados

O questionário foi aplicado com um total de 9 usuários na faixa etária de 20 a 35 anos e em seguida foi realizada a análise das respostas. Para representar da melhor maneira o resultado do teste, foram calculadas as médias das categorias do PSSUQ apresentadas anteriormente bem como a média individual por questão:

- Média SYSUSE: 1,64
- Média INFOQUAL: 1,90
- Média INTERQUAL: 1,67

A Figura 11 mostra o gráfico que contém a média de cada questão separadamente.



Figura 11: Média dos resultados por questão

A média de todas as categorias ficou abaixo de 2, representando uma avaliação positiva da aplicação.

É possível observar também que a pior avaliação foi na questão 7 que se refere às mensagens de erro que o sistema apresentou ao usuário. Apesar de ter sido a média mais alta, essa média ainda assim representa uma avaliação positiva levando em consideração a escala do PSSUQ que varia de 1 a 7.

Portanto, podemos concluir que o nível de satisfação dos usuários com a aplicação foi positivo.

5. EXPERIÊNCIA

Nesta seção será relatado como se deu o processo de desenvolvimento da aplicação, desafios e os trabalhos futuros.

5.1 Processo de desenvolvimento

Primeiramente, foram definidas quais tecnologias seriam utilizadas. Elas foram escolhidas levando em conta as necessidades da aplicação e a experiência do desenvolvedor com a tecnologia.

Para melhor planejar as funcionalidades do sistema, o desenvolvedor realizou uma pesquisa empírica participando de uma ONG que realiza trabalhos voluntários na causa animal em sua cidade. Com esta experiência, foi possível analisar as rotinas e necessidades presentes naquele contexto e por fim planejar as funcionalidades.

O desenvolvimento foi iniciado pelo servidor, os modelos e *endpoints* foram preparados de forma que após a integração com o cliente, seria necessário apenas pequenos ajustes. Então, com o servidor parcialmente pronto, iniciou-se o desenvolvimento do cliente.

Primeiramente, foram criados os códigos HTML (HyperText Markup Language) e CSS (*Cascading Style Sheets*) da aplicação para em seguida ser adicionado o comportamento dos componentes. Após as telas do sistema estarem prontas, realizou-se a conexão entre o cliente e o servidor. Nesta etapa, foram feitos os ajustes comentados anteriormente como, por exemplo, a adição de paginação em algumas chamadas.

Por último, foi realizado o *deploy* da aplicação na plataforma *Heroku*.

5.2 Desafios

O desenvolvimento de uma interface intuitiva e agradável para os usuários foi um pré-requisito para este projeto. Porém, esta tarefa costuma não ser simples para desenvolvedores.

Com o intuito de contornar este problema, ao final do desenvolvimento, foi solicitado a uma Designer um trabalho de UI/UX (*User Interface/User Experience*) para a aplicação.

Foram feitas reuniões para discutir as funcionalidades e os fluxos presentes no sistema e aos poucos melhorias foram sendo

implementadas. Além disso, a aplicação obteve uma identidade visual através das cores, ícones e imagens personalizadas que foram inseridas.

5.3 Trabalhos futuros

Existem necessidades e melhorias que não foram incluídas neste primeiro momento na solução desenvolvida como, por exemplo, a arrecadação de recursos para as instituições dentro da própria aplicação. Além disso, outras funcionalidades podem ser adicionadas, como:

- Adição de categorias para os eventos;
- Sistema de recomendação de eventos com base nos interesses do usuário;
- Chat integrado a aplicação;
- Gamificação de acordo com as participações dos usuários;
- Inserir sistema de cache para evitar requisições em excesso para o servidor;
- Melhor tratamento das mensagens de erro.

6. REFERÊNCIAS

- [1] FERREIRA, M.; PROENÇA, Teresa.; PROENÇA, J. F. As motivações no trabalho voluntário - 2008. Disponível em: <https://core.ac.uk/download/pdf/47136954.pdf>. Acesso em 26 jan. 2021.
- [2] IBGE - Instituto Brasileiro de Geografia e Estatística, Outras Formas de Trabalho - 2018. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101650_informativo.pdf. Acesso em 26 jan. 2021.
- [3] ONGS Brasil. Disponível em: <https://www.ongsbrasil.com.br/>
- [4] AU-YEUNG J.; DONOVAN R. Best practices for REST API design - 2020. Disponível em: <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/>. Acesso em 30 abril 2021.
- [5] WILL, T. PSSUQ Post-Study System Usability Questionnaire - s.d. Disponível em: <https://uiuxtrend.com/pssuq-post-study-system-usability-questionnaire/>. Acesso em 02 maio 2021.
- [6] HACKER, R. Platform as a Service: o que é PaaS? - 2020. Disponível em: <https://blog.vindi.com.br/platform-as-a-service-paas/>
- [7] Node.js. Disponível em: <https://nodejs.org/en/>
- [8] MySQL. Disponível em: <https://www.mysql.com/>
- [9] React, uma biblioteca JavaScript para criar interfaces de usuário. Disponível em: <https://pt-br.reactjs.org/>
- [10] Styled components, visual primitives for the component age. Disponível em: <https://styled-components.com/>
- [11] Reactstrap, Easy to use React Bootstrap 4 components. Disponível em: <https://reactstrap.github.io/>
- [12] Heroku. Disponível em: <https://www.heroku.com/home>