

CEU – uma infraestrutura para gestão de eventos acadêmicos da UESPI

Kleidimilson José I. de Sousa¹, Alcemir Rodrigues Santos¹

¹ Laboratório de Engenharia de Software
Universidade Estadual do Piauí (UESPI)
Piripiri – PI – Brazil

kleidimilson@aluno.uespi.br, alcemir@prp.uespi.br

Resumo. *A gestão de eventos científicos é um processo manual na maioria das vezes, dificultando a criação e participação. O presente trabalho apresenta o CEU, uma infraestrutura para gestão de eventos acadêmicos da Universidade Estadual do Piauí que tem como objetivo permitir a criação de eventos e atividades, a fim de tornar o processo de realização de eventos mais fácil. O trabalho utilizou-se da metodologia ágil Scrum e durante todo o desenvolvimento foram utilizadas ferramentas consolidadas no mercado de desenvolvimento de software, como Node JS, PostgreSQL, React native e TypeScript.*

1. Introdução

De acordo com o CEPEX (Conselho de Ensino, Pesquisa e Extensão) [UESPI 2021] na Resolução 002/2021, as atividades acadêmicas Científicos-culturais devem ser desenvolvidas pelos discentes a partir de seu ingresso na graduação. O objetivo dessas atividades é estimular o espírito científico na academia e o pensamento reflexivo dos estudantes, além de promover a teoria e prática.

As Atividades Acadêmico, Científico, Culturais (AACC's) são requisitos indispensáveis e podem ser realizadas fora da Universidade ou dentro da própria instituição. De acordo com [UESPI 2021], são consideradas AACC's a participação em cursos de extensão, congressos, seminários, jornadas, simpósios, *workshop* e conferências promovidas pela Universidade Estadual do Piauí (UESPI) ou outros órgãos de formação de forma online ou remota. Portanto, os Eventos Científicos possuem uma grande importância no processo da comunicação científica, visto que as informações sobre estudos e inovações chegam de forma mais fácil à comunidade científica. Desta forma, a criação de ferramentas que estimulem a criação de eventos Científicos torna-se necessária para a universidade.

A maioria dos softwares presentes no mercado são pagos ou não possuem as funcionalidades necessárias para atender a instituição tornando a gestão de eventos muito manual e custosa. Além disso, as informações de eventos disponíveis estão espalhadas em diversos meios tais como: Whatsapp, Instagram, Telegram e o site da universidade. Diante disso, o estudante não possui um lugar que centralize as informações dos eventos ofertados de forma que ele busque diretamente as informações caso esteja a procura de AACC's para cumprir as horas necessárias de atividades extracurriculares e os organizadores não possuem uma ferramenta que possibilite uma boa gestão desses eventos. Estes eventos são divulgados através dos canais de comunicação contendo informações sobre o evento e o *link* para a realização da inscrição em uma plataforma externa. Além

disso, há eventos específicos que ocorrem em determinados campus da instituição, como minicursos, que acabam não tendo uma boa divulgação e uma ferramenta de inscrições.

Uma das ferramentas mais utilizadas para a realização de inscrições de eventos é o Google Forms, uma ferramenta desenvolvida pela Google para a criação e gestão de formulários, tal ferramenta tem como propósito a criação de simples formulários e a gestão de eventos utilizando Google Forms torna o processo manual por parte dos organizadores e participantes visto que tal ferramenta não foi feita para este propósito. Outra ferramenta muito utilizada é o Event3 uma plataforma para o gerenciamento de eventos, apesar do sistema ser gratuito é cobrado uma taxa de 10% para eventos pagos e seu aplicativo móvel não possibilita a gestão dos eventos, apenas o acompanhamento das inscrições.

A proposta deste trabalho foi desenvolver um aplicativo móvel para o gerenciamento e a inscrição de eventos acadêmicos da instituição, tornando o processo de busca e disponibilização de eventos mais fácil. Com isso, buscou-se por meio desse trabalho facilitar a participação e a gestão de eventos criando um canal que conecta a instituição e os estudantes, possibilitando o aumento da oferta de eventos na Universidade. As contribuições do presente trabalho são:

- O desenvolvimento de um aplicativo utilizando uma tecnologia de desenvolvimento *cross-platform*;
- A facilitação na elaboração e inscrição em eventos acadêmicos;
- O desenvolvimento de uma API Rest;
- A modelagem de um banco de dados Postgres;
- A elaboração do design das telas do aplicativo;

O presente trabalho tem como alvo alunos e professores da Universidade Estadual do Piauí, no entanto, algumas atividades estão fora do escopo deste trabalho: (i) Definir políticas de eventos e atividades; (ii) Prover uma aplicação web para o software; (i) Definir uma política de atividades curriculares acadêmicos.

O restante do trabalho está organizado da seguinte maneira: A Seção 2 apresenta uma revisão bibliográfica. A Seção 3 exibe a metodologia do trabalho. A Seção 4 exibe informações acerca do produto desenvolvido. A Seção 6 exibe as considerações finais do trabalho e a Seção 7 apresenta os trabalhos futuros.

2. Revisão Bibliográfica

Esta Seção apresenta conceitos relacionados ao desenvolvimento deste trabalho: como o Scrum, Gestão de eventos e os sistemas de gestão de eventos acadêmicos existentes no mercado.

2.1. Scrum

De acordo com Valente [Valente et al. 2020] o Scrum é uma metodologia ágil, interativa, incremental para o gerenciamento de projetos. Tal metodologia foi proposta por Jeffrey Sutherland e Ken Schwaber em um artigo publicado em 1997 [Schwaber 1997]. No Scrum, o projeto é dividido em ciclos chamados de Sprints. As Sprints são um período de tempo geralmente de uma a três semanas com atividades que devem ser executadas e concluídas neste período. Segundo [Valente et al. 2020] os times scrum são compostos por

três papéis: *Product Owner*, *Scrum Master* e *Development Team*. e todas as funcionalidades deverão ser adicionadas a uma lista chamada *Product Backlog*. Após isso, é feita uma reunião chamada *Sprint Planning* onde é feito o planejamento e a priorização das atividades que serão executadas em uma Sprint. Todas as tarefas que devem ser implementadas na Sprint são retiradas do *Product Backlog* e transferidas para uma outra lista, chamada *Sprint Backlog*. A cada dia é feita uma rápida reunião chama *Daily Sprint* com o objetivo de responder três perguntas: O que eu fiz ontem? O que eu irei fazer hoje? Algo me impede?. Ao final de cada Sprint é feita uma revisão das tarefas que foram implementadas chamada de *Sprint Review Meeting*.

2.2. Gestão de eventos

De acordo com Silvers [Silvers 2005], gerenciar um evento é o processo em que o evento é planejado, preparado e produzido. Além disso, para Silvers [Silvers 2005] um evento deve possuir avaliação, definição, aquisição, atribuição de direção, controle e análise de tempo, finanças, pessoal, produtos, serviços e outros recursos afim de alcançar os objetivos. Desta forma sugere que a atividade de um organizador de evento seja definida como a pessoa, cujo trabalho é supervisionar e organizar todos os aspectos de um evento, que incluem planejamento, organização, implementação e reuniões de controle, convenções e outras atividades similares.

2.3. Trabalhos Relacionados

Algumas universidades disponibilizam um sistema para a divulgação e criação de eventos acadêmicos. A Universidade Federal do Ceará possui um sistema web para a divulgação de eventos da instituição [do Ceará 2022]. Entretanto, apenas discentes têm acesso ao sistema. A Universidade Federal de Sergipe (UFS) possui um sistema que permite aos discentes visualizar os eventos abertos e fazer a inscrição bem como o acompanhamento das inscrições realizadas.

O Even3 [em Tecnologia 2022] é um software desenvolvido pela empresa L3 Soluções em Tecnologia que permite a criação e gestão de eventos. O software é grátis para eventos gratuitos e é cobrado uma taxa de 10% sobre o valor de cada inscrição caso o evento seja pago. A ferramenta permite inscrições via boleto, cartão e pix além de emitir certificados e o credenciamento do evento. Dentre as instituições que utilizam este software é possível citar a FGV e o Instituto Federal de São Paulo. O even3 possui dois aplicativos presentes na Play Store: Even3 Eventos e o Even3 Credencia. O Aplicativo Even3 Eventos permite ao usuário a visualização de eventos disponíveis e o acompanhamento de suas inscrições. O aplicativo Even3 Credência tem a função de fazer o credenciamento de atividades, registro de apresentações de trabalhos e a emissão de certificados.

A Doity [Barbosa and Neves 2022] é uma plataforma desenvolvida por ex-universitários Uziel Barbosa e Geraldo Neves. Durante a organização de um evento no Instituto Federal de Alagoas (Ifal), perceberam a dificuldade em gerenciá-lo e decidiram criar o software. A plataforma permite a criação de eventos, credenciamento, inscrições em atividades, emissão de certificados e a avaliação e gerenciamento de trabalhos científicos. O software é gratuito para criar eventos de até 3 mil inscritos e para os eventos pagos é cobrada uma taxa de 10% sobre cada inscrição. A Doity possui o aplicativo Doity Organizador disponível na Play Store e permite ao usuário fazer o *check-in* dos eventos cadastrados na plataforma.

Ferreira [FERREIRA VIEIRA 2021] apresenta em seu trabalho de conclusão de curso, apresenta o Eventoservice: aplicação para auxílio à organização de eventos acadêmicos [Ferreira 2021]. O trabalho é um software que permite a criação de conteúdo para um evento científico. O Eventoservice disponibiliza suas funcionalidades em uma API (Application Programming Interface”), desse modo, pode ser integrada a outros sistemas.

3. Metodologia

Esta Seção aborda a metodologia utilizada no trabalho e as ferramentas que foram utilizadas durante o desenvolvimento do aplicativo.

3.1. Visão Geral

A gestão do projeto foi feita através da metodologia Scrum, tal metodologia foi escolhida porque torna o processo de gestão do projeto mais flexível e incremental. O Scrum não tem apenas como objetivo o gerenciamento de projetos de softwares e pode ser aplicada em diversos contextos. Tal metodologia foi escolhida pois segundo [Schwaber 1997] o Scrum ajuda indivíduos e empresas a gerarem valor por meio de soluções adaptáveis para problemas complexos. Além disso, permite o gerenciamento da codificação do projeto, bem como a construção do documento de conclusão de curso.

3.2. Descrição do processo

O processo de desenvolvimento do prezado trabalho foi dividido em 5 etapas: (i) levantamento de requisitos; (ii) projeto; (iii) desenvolvimento da API; (iv) construção do aplicativo; e (v) teste. Em seguida, discutimos cada uma delas.

Etapla 1: Levantamento de Requisitos Nesta etapa, foi levantado os requisitos do aplicativo por meio de histórias de usuários. Os requisitos foram criados tendo como base funcionalidades de outros softwares presentes no mercado como abordado na Tabela 6. Esta etapa permitiu descrever as necessidades dos usuários de uma forma mais simplificada. As Estórias de Usuário foram classificadas em níveis de complexidade seguindo os números da Sequência de Fibonacci, quanto maior o número escolhido maior será a complexidade da tarefa.

Etapla 2: Projeto Nesta etapa, iniciou-se o planejamento da arquitetura de toda a plataforma por meio da criação de Diagramas do sistema, *Wireframes* do aplicativo, além da modelagem do banco de dados. Para a elaboração do diagrama de arquitetura foi utilizado o software Astah Community, por meio da linguagem UML e a ferramenta utilizada para a criação telas detalhadas foi o Figma.

Etapla 3: Desenvolvimento da API Nesta etapa, iniciou-se o desenvolvimento do projeto. Esta etapa foi iniciada com a construção da API em Node JS. De acordo com [Masse 2011] uma API é um software que expõe dados e funções com o objetivo de facilitar as interações entre o servidor e outros sistemas. O backend, serve um JSON (*JavaScript Object Notation*) que é consumido pelo aplicativo.

Etapla 4: Construção do aplicativo Nesta etapa, foi feita a codificação do aplicativo. A ferramenta escolhida para o desenvolvimento do aplicativo foi o React Native. Para este aplicativo foi utilizado a versão EXPO do *framework*.

Etapla 5: Teste Nesta etapa, o aplicativo foi submetido a uma análise heurística com objetivo de detectar possíveis problemas de usabilidade.

3.3. Ferramentas e Frameworks

Esta Seção irá abordar as ferramentas de desenvolvimento e produtividade que foram utilizadas durante o andamento do projeto. Para a codificação da API foi utilizada a linguagem TypeScript. Tal linguagem que é baseada no JavaScript permite a utilização de tipagem estática no projeto. Além da linguagem foram utilizadas as seguintes ferramentas e *frameworks*.

3.3.1. Ferramentas

Trello: O gerenciamento das tarefas foi realizado através do Trello. Tal ferramenta permite o gerenciamento de qualquer tipo de projeto, fluxo de trabalho ou monitoramento de tarefas. Tal ferramenta permite o controle da produtividade.

GitHub: Para o versionamento de código, foi utilizado o Github. Esta tecnologia permite a hospedagem de códigos e o controle de versão utilizando o Git. Tal ferramenta será usada durante toda fase de desenvolvimento.

Docker: Para o *deploy* da API do projeto, foi utilizado o docker. Tal tecnologia permite a criação de contêineres e imagens. Por meio desta ferramenta é possível criar ambientes de desenvolvimento de uma forma facilitada.

Swagger: Para a documentação da API foi utilizada a ferramenta Swagger. Tal ferramenta é *open source* permite a descrição das rotas e parâmetros presentes no sistema.

PostgreSQL: O armazenamento de dados foi feito através do Postgres. De acordo com a documentação oficial [Postgres 1989] PostgreSQL é um sistema de gerenciamento de banco de dados (SGBD), desenvolvido por Michael Stonebraker na Universidade da Califórnia no Departamento de Ciência da Computação de Berkeley. Dentre as funcionalidades encontradas no postgres é possível citar: (i) consultas complexas; (ii) chaves estrangeiras; (iii) *integridade transacional*; e (iv) *controle de simultaneidade multiversão*.

Azure: O deploy da API e o upload de imagens e certificados foram feitos através da Azure utilizando o plano de estudante. A azure é uma plataforma de execução de aplicativos e serviços, baseada nos conceitos da computação em nuvem.

3.3.2. Frameworks

React Native

O aplicativo foi desenvolvido através do React Native. De acordo com o livro [Eisenman 2015], o React native é um *framework* Javascript para a construção de aplicativos móveis que possuem renderização nativa. Tal tecnologia é baseada em React, uma biblioteca desenvolvida pelo facebook para a criação de interfaces web. De acordo com a documentação do React Native [Meta 2015], é possível utiliza-lo em projetos Android, IOS. Todo o código da aplicação é feito com a linguagem de programação Javascript, esse fato permite que com um mesmo código uma equipe de desenvolvimento mantenha seu aplicativo para as duas plataformas.

O React Native foi desenvolvido em 2015 e desde então é amplamente utilizado por grandes empresas de tecnologia como: Twitter, Instagram, Tesla e Discord. De acordo

com o site oficial [Meta 2015], no ano de 2018, o React Native teve o 2º maior número de contribuidores no GitHub. Atualmente, o *framework* possui contribuições de empresas de todo o mundo, incluindo Callstack, Expo, Infinite Red, Microsoft e Software Mansion.

Node.js

A construção da API foi desenvolvida em node, um software de código aberto, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript no servidor. Inicialmente a linguagem Javascript foi desenvolvida para rodar em navegadores, a criação do node passou a permitir o desenvolvimento de uma aplicação completa utilizando Javascript. O Node.js é uma tecnologia essencial para várias empresas em todo o mundo. As empresas que utilizam tal tecnologia em seus projetos são Netflix, LinkedIn, Walmart, Trello.

4. CEU – Central de Eventos da UESPI

Esta seção descreve em mais detalhes o produto gerado com a execução deste Trabalho de Conclusão de Curso.

4.1. Arquitetura

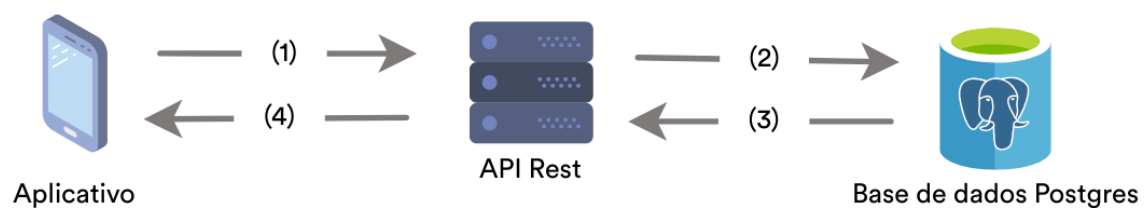


Figura 1. Arquitetura do sistema

Fonte: Autoria Própria

A Figura 1 descreve a arquitetura do projeto. É possível observar os passos para a comunicação entre o aplicativo, o servidor e a base de dados: (1) O aplicativo faz uma requisição HTTP para o servidor; (2) o servidor executa as consultas no banco de dados; (3) o banco de dados retorna o resultado das consultas para o servidor; (4) o servidor envia uma resposta para o aplicativo.

4.2. Casos de Uso

A Tabela 1 descreve as histórias de usuário definidas para o sistema. A coluna “Estória” descreve os requisitos em alto nível. A coluna “Ator” menciona dois tipos diferentes de atores: (i) Organizador; (ii) Participante. Cada um deles pode ser descrito como abaixo:

Organizador: O Organizador é o ator responsável pela criação e o gerenciamento de um determinado evento. Ele cadastrará todas as informações de um evento e fará a emissão dos certificados para os participantes.

Participante O Participante é o ator que se inscreverá nos eventos e atividades disponibilizados no aplicativo.

Tabela 1. Estórias de usuários definidas para o CEU.

#	Ator	Estória
1	Organizador	ES01: Eu como Organizador, quero criar um evento para que os alunos possam participar
2	Organizador	ES02: Eu como Organizador, quero configurar um evento criado anteriormente
3	Organizador	ES03: Eu como Organizador, quero poder cancelar um evento
4	Organizador	ES04: Eu como Organizador quero confirmar a presença dos participantes no evento
5	Organizador	ES05: Eu como Organizador quero avisar por email caso um evento seja cancelado
6	Organizador	ES06: Eu como Organizador, quero emitir certificados para os participantes para comprovar a participação no evento
7	Organizador	ES07: Eu como Organizador, quero emitir certificados para os palestrantes para comprovar a participação no evento
8	Participante	ES08: Eu como participante, quero poder pesquisar os eventos disponíveis no aplicativo
9	Participante	ES09: Eu como participante, quero ter acesso ao certificado para que eu possa baixá-lo
10	Participante	ES10: Eu como participante, quero me inscrever em um evento para eu participar
11	Participante	ES11: Eu como participante, quero filtrar os eventos disponíveis no aplicativo de acordo com seu tipo

ES01 O organizador ao entrar com seu e-mail institucional é capaz de criar novos eventos no sistema.

ES02 O organizador após ter criado um evento no sistema, pode configurá-lo e editar as informações deste evento.

ES03 O organizador após ter um evento cadastrado no sistema, pode excluí-lo do sistema

ES04 O organizador após possuir inscritos em um evento, pode selecionar os participantes presentes.

ES05 Após o organizador cancelar um evento, todos os inscritos recebem um e-mail avisando o seu cancelamento.

ES06 Após o organizador selecionar os inscritos de um evento, ele pode gerar os certificados.

ES07 Após o organizador selecionar os inscritos de um evento, ele pode gerar os certificados.

ES08 O participante pode pesquisar os eventos que estão disponíveis no sistema.

ES09 Após o organizador emitir os certificados, o participante pode visualizar seu certificado pelo aplicativo e pelo seu e-mail.

ES10 O participante pode fazer inscrições em eventos disponíveis no sistema.

ES11 O participante pode filtrar os eventos disponíveis na plataforma. Os filtros presentes no aplicativo são: Palestras, Simpósios, Conferências e Minicursos

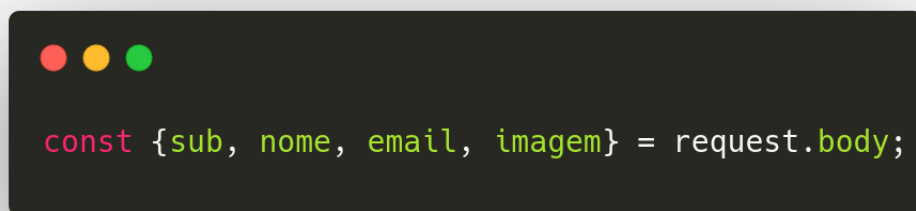
Tabela 2. Descrição dos endpoints de Usuário.

Método	Endpoint	Descrição
POST	/usuarios/	Rota para criar um usuário
GET	/usuario/:sub	Rota que busca um usuário pelo id
PUT	/usuarios/:sub	Rota para editar um usuário
DELETE	/usuarios/:sub	Rota para deletar um usuário

4.3. Backend

4.3.1. API

A Tabela 2 mostra os *Endpoints* presentes na API referentes ao usuário. Através dessas rotas, o sistema cria, edita, lista e deleta um usuário.



```
const {sub, nome, email, imagem} = request.body;
```

Figura 2. Parâmetros da rota de criação usuário.

A Figura 2 mostra os parâmetros recebidos pela API necessários para a criação de um usuário.

A Tabela 3 expõe os *Endpoints* referentes a eventos. Através dessas rotas, o sistema cria, lista, busca, deleta e recupera os eventos de um organizador.

A Figura 3 mostra os parâmetros recebidos pela API necessários para a criação de um evento.

A Tabela 4 mostra os *Endpoints* referentes às atividades. Através dessas rotas, o sistema cria, lista, busca, deleta e recupera as atividades do sistema.

A Figura 4 mostra os parâmetros recebidos pela API na criação de uma atividade.

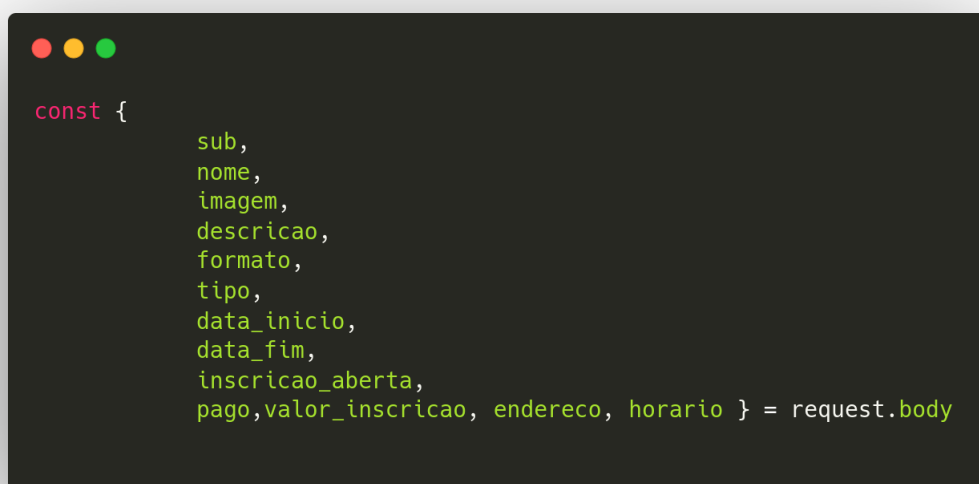
A Tabela 5 exhibe os *Endpoints* referentes as inscrições e certificados. Através dessas rotas, o sistema inscreve um participante em um evento, lista as inscrições de um evento, lista as inscrições de um usuário e gera um certificado.

A Figura 5 mostra os parâmetros recebidos pela API na inscrição em um evento.

A Figura 6 mostra os parâmetros recebidos pela API na criação de um certificado.

Tabela 3. Descrição dos endpoints de eventos.

Método	Endpoint	Descrição
POST	/eventos/	Rota para criar um evento
GET	/eventos/	Rota para listar todos os eventos
GET	/eventos/:nome	Rota para buscar um evento pelo nome
DELETE	/eventos/:id/:nome	Rota para deletar um evento
GET	/eventos/organizador/:nome	Rota para listar os eventos de um organizador



```
const {
  sub,
  nome,
  imagem,
  descricao,
  formato,
  tipo,
  data_inicio,
  data_fim,
  inscricao_aberta,
  pago, valor_inscricao, endereco, horario } = request.body
```

Figura 3. Parâmetros da rota de criação de um evento.

4.3.2. Banco de dados

A Figura 7, mostra o modelo do banco de dados desenvolvido. Na figura é possível observar as entidades e os relacionamentos entre as tabelas.

4.4. Front-end

4.4.1. Personas

De acordo com [Blomkvist 2002] uma persona é uma representação arquetípica de usuários reais ou potenciais. As personas representam padrões comportamentais, motivações e objetivos dos usuários reunidos em um usuário fictício.

Participante dos eventos Pedro Paulo tem 21 anos, é universitário. Pensa em participar de eventos acadêmicos para obter conhecimento; Está buscando eventos disponíveis em sua universidade para cumprir a carga horária de atividades

Tabela 4. Descrição dos endpoints de atividades.

Método	Endpoint	Descrição
POST	/atividades/	Rota para criar uma atividade
PUT	/atividades/:id	Rota para editar uma atividade
DELETAR	/atividades/:id	Rota para deletar uma atividade
GET	/atividades	Rota para listar todas as atividades
GET	/atividades/:id	Rota para buscar uma atividade

Tabela 5. Descrição dos endpoints de inscrição.

Método	Endpoint	Descrição
POST	/inscricao/	Rota para fazer uma inscrição
GET	/inscritos/:nome-evento	Rota para listar inscritos de um evento
GET	/inscricoes/:nome-usuario	Rota para listar inscrições de um usuário
POST	/certificado	Rota para gerar um certificado
GET	/certificados/:usuario-sub	Rota para listar os certificados de um usuário

acadêmicas científicas culturais. Em meio a tantos meios de divulgação Pedro Paulo fica confuso na hora de buscar eventos.

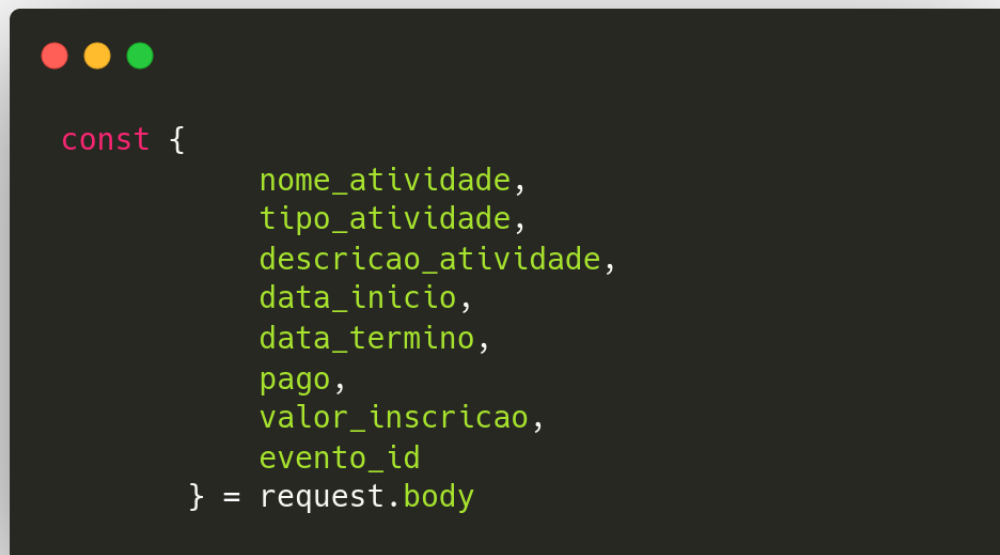
Organizador do evento Antônio tem 40 anos, é professor universitário. Pensa em organizar eventos acadêmicos para a comunidade acadêmica; Está buscando uma ferramenta disponível para criar e gerenciar seus eventos de um forma simples e gratuita.

4.4.2. Interface de Usuário

Esta Seção expõe as interfaces desenvolvidas no projeto e o fluxo de navegação das mesmas.

As Figuras 8, 9 e 10, mostram os diagramas de atividades do sistema. Na UML(Unified Modeling Language), um diagrama de atividade permite uma visualização do comportamento de um determinado sistema. Tal diagrama descreve uma sequência de ações. O diagrama de atividade possui uma semelhança com fluxogramas porque ambos exibem o fluxo das ações em uma atividade; entretanto, os diagramas de atividades também possuem algumas particularidades como a exibição de fluxos paralelos ou simultâneos e fluxos alternativos.

As Figuras 11 e 12 mostram as interfaces do Aplicativo. A Figura 11 possui todas as interfaces relacionadas ao organizador dos eventos e a Figura 12 possui as interfaces do participante.



```
const {
  nome_atividade,
  tipo_atividade,
  descricao_atividade,
  data_inicio,
  data_termino,
  pago,
  valor_inscricao,
  evento_id
} = request.body
```

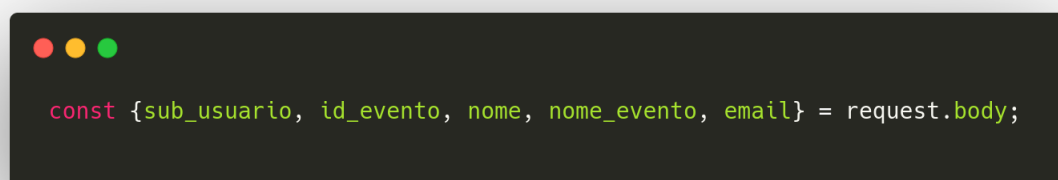
Figura 4. Parâmetros da rota de criação de uma atividade.

Tela de Login: As Figuras 11(a), 12(a) mostram a tela de login do aplicativo. O login é feito através do Google. De acordo com o tipo do e-mail o usuário é categorizado entre organizador e participante. Apenas usuários logados com o e-mail institucional da UESPI são capazes de registrar novos eventos.

Tela de início: As Figuras 11(b), 12(b) mostram a tela inicial do aplicativo que exibe todos os eventos disponíveis. O usuário pode filtrar esses eventos pelo tipo.

Tela de um evento A Figura 12(e) mostra a tela de um determinado evento permite que o usuário visualize todas as informações sobre um evento como: nome, descrição, horário, data e realizar sua inscrição no mesmo.

Tela de adicionar evento A Figura 11(d) mostra a tela de criação de eventos. Ela per-



```
const {sub_usuario, id_evento, nome, nome_evento, email} = request.body;
```

Figura 5. Parâmetros da rota de inscrição em um evento

```
const { nome_evento,
        nome_usuario,
        email_usuario,
        grade,
        id,
        usuario_sub,
        evento_id } = request.body;
```

Figura 6. Parâmetros da rota de criação de certificado

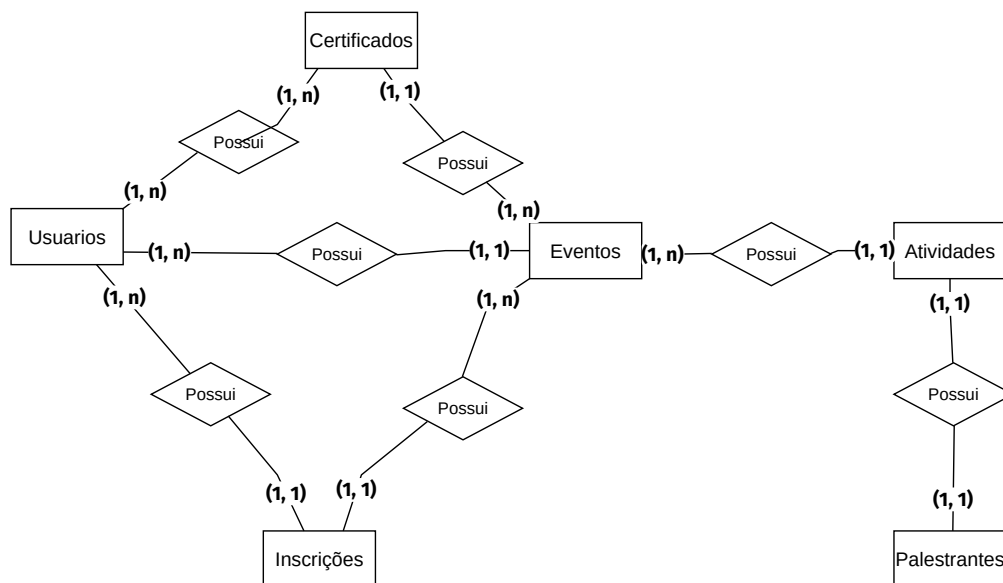


Figura 7. Modelo do Banco de Dados.

Fonte: Autoria Própria

mite que um usuário organizador cadastre um novo evento no aplicativo.

Tela de gerenciamento do evento A Figura 11(e) exibe a tela que permite a edição das informações de um evento, visualização dos inscritos, geração de certificados e cancelamento dos eventos.

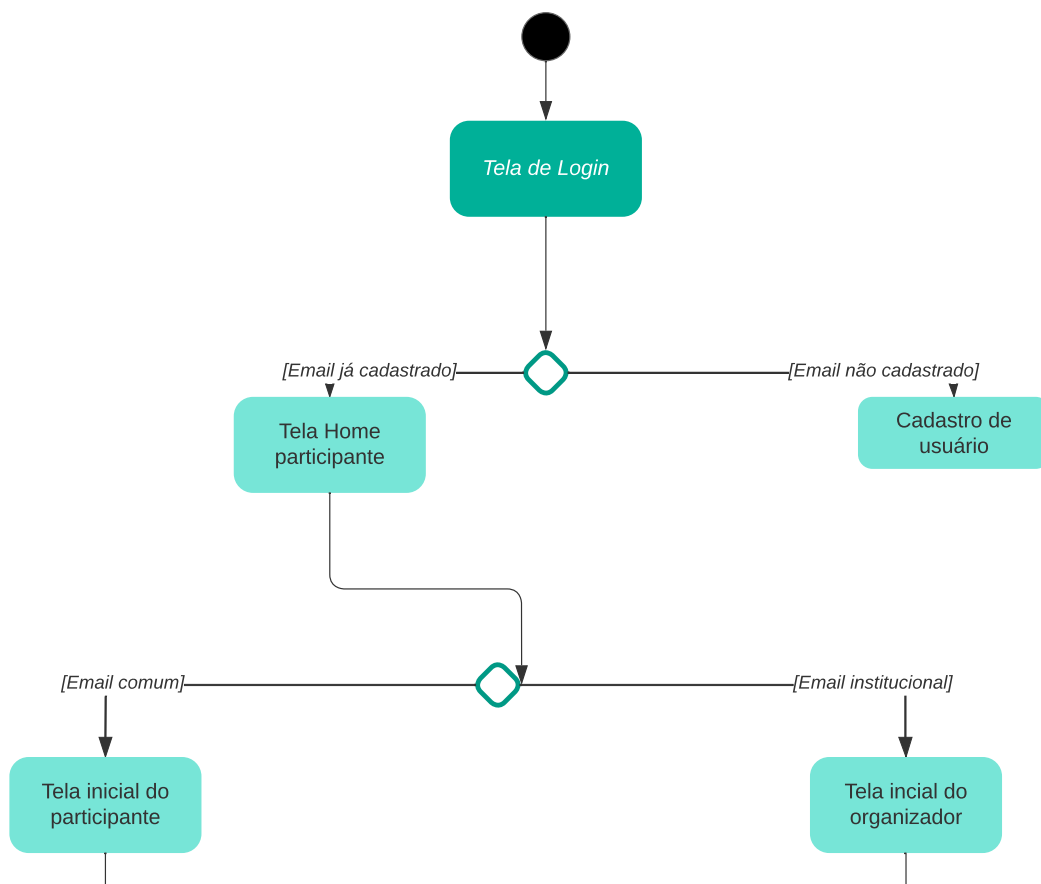


Figura 8. Diagrama de atividade do Login.

Fonte: Autoria Própria

Tela de minhas inscrições As Figuras 11(f), 12(d) exibem as telas que permitem a visualização das inscrições do usuário.

Tela de pesquisa de eventos As Figuras 11(c), 11(c) exibem as telas que permitem a pesquisa de eventos.

O sistema é capaz de gerar um certificado com base nas informações do evento e dos participante. A Figura 13 mostra o modelo do certificado gerado no CEU.

5. Avaliação

5.1. Comparação dos sistemas

A Tabela 6 compara as características dos softwares abordados anteriormente, com o aplicativo proposto no prezado trabalho. Através da tabela, é possível identificar as características dos sistemas presentes no mercado e compara-las com as do CEU (Central de Eventos da UESPI). As características comparadas foram: voltada para eventos acadêmicos, é totalmente gratuita, permite a emissão de certificados, permite o credenciamento, permite a inscrição em um evento por meio do aplicativo, permite o gerenciamento de um evento por meio do aplicativo.

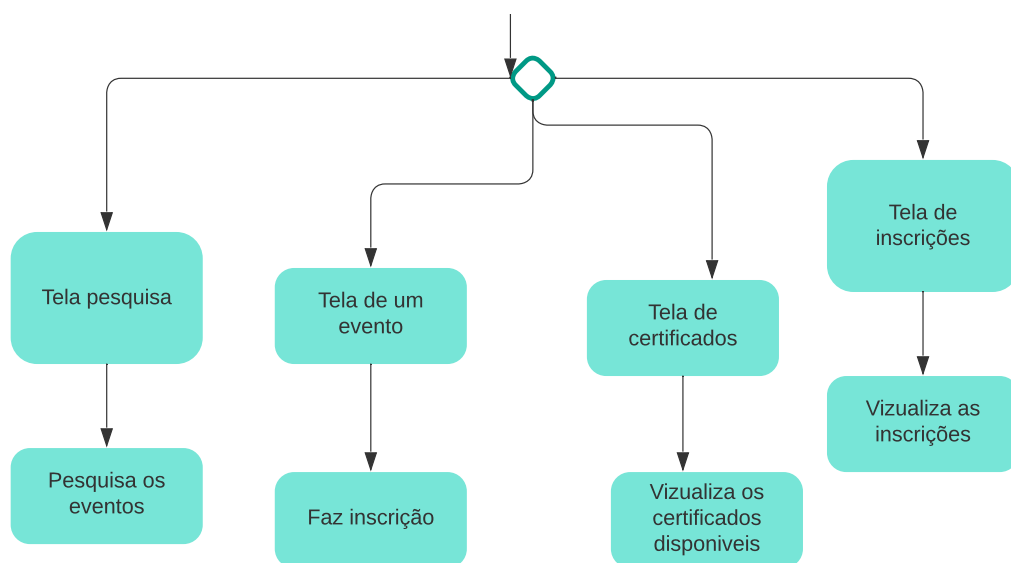


Figura 9. Diagrama de atividade do Participante.

Fonte: Autoria Própria

5.2. Avaliação Heurística

As avaliações das interfaces do aplicativo foram feitas utilizando as heurísticas propostas por [Nielsen 1994]. A análise heurística consiste em submeter as interfaces de um sistema à avaliação sobre um conjunto de heurísticas. Desse modo, é possível identificar problemas de usabilidade e posteriormente corrigi-los.

Após a análise heurística do aplicativo, foram identificados alguns problemas,

Tabela 6. Tabela de características das plataformas.

Plataformas				
Características	EVEN3	DOITY	EVENTOSERVICE	CEU
É voltada para eventos acadêmicos			✓	✓
É totalmente gratuito			✓	✓
Permite a emissão de certificados	✓	✓		✓
Permite o credenciamento	✓	✓	✓	✓
Permite a inscrição em um evento por meio do aplicativo	✓			✓
Permite o gerenciamento por meio do aplicativo				✓
Disponibiliza o certificado no aplicativo e no e-mail	✓	✓		✓

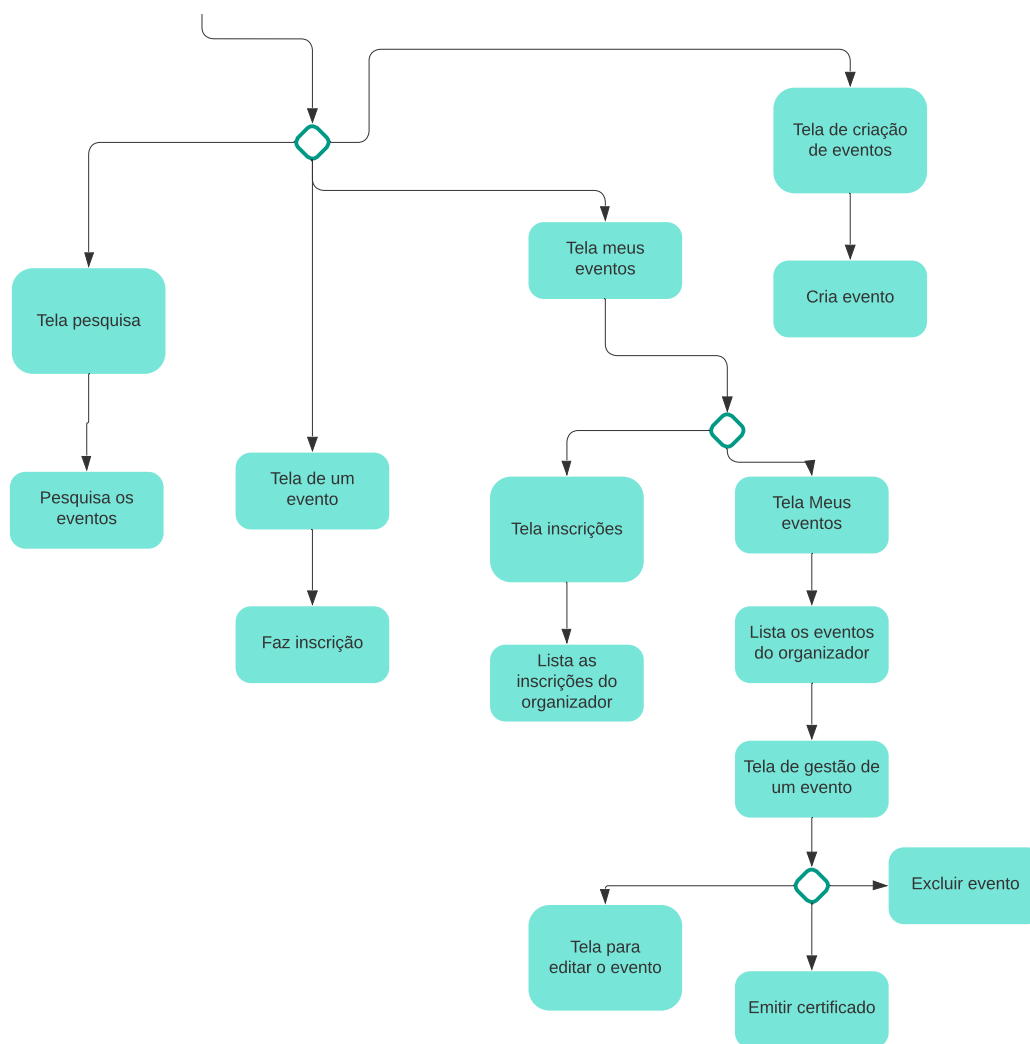


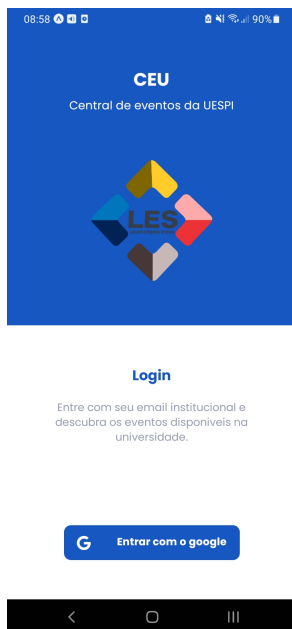
Figura 10. Diagrama de atividade do Organizador.

Fonte: Autoria Própria

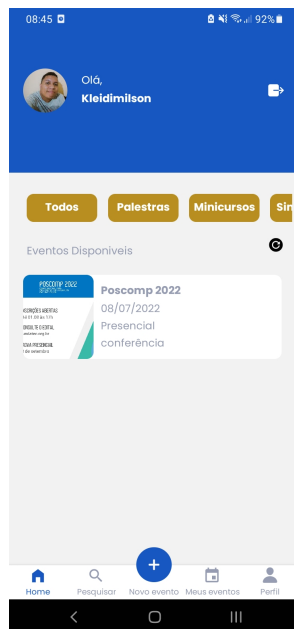
como mostrados na Tabela 9. As faltas de ícones de carregamento no aplicativo foram corrigidas em todas as telas que fazem requisições para a API. Além disso, foi corrigido a falta de um alerta sobre a ausência de eventos no sistema. Por fim, apesar da identificação do problema: Não mostra a data selecionada na tela de criação de eventos, a correção desse erro não foi possível devido a falta de tempo.

6. Considerações Finais

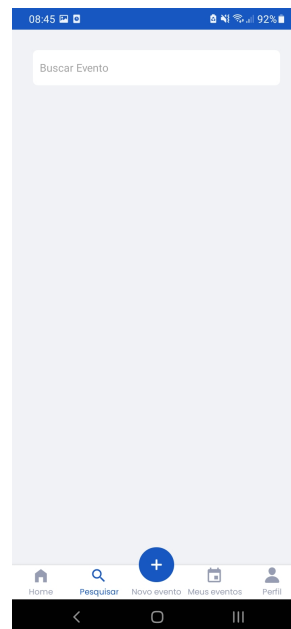
Este trabalho buscou construir um sistema para a gestão de eventos acadêmicos da UESPI. Durante a construção do sistema foi desenvolvida uma API Rest em Node Js e a construção do aplicativo foi feita utilizando o React Native uma tecnologia para a construção de aplicativos cross-platform. A modelagem do banco de Dados foi feita através no Postgres. Através das mencionadas foi possível construir uma aplicação completa que pode amplamente utilizada por professores e estudantes da Universidade Estadual do Piauí para a elaboração e participação de eventos.



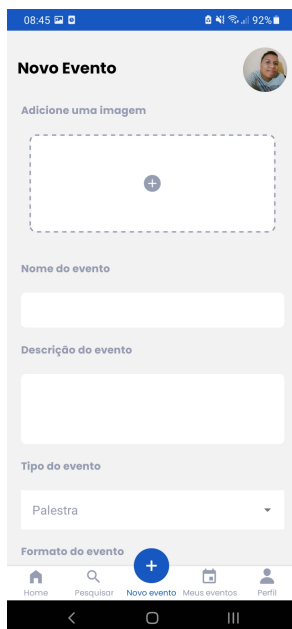
(a) Tela de Login.



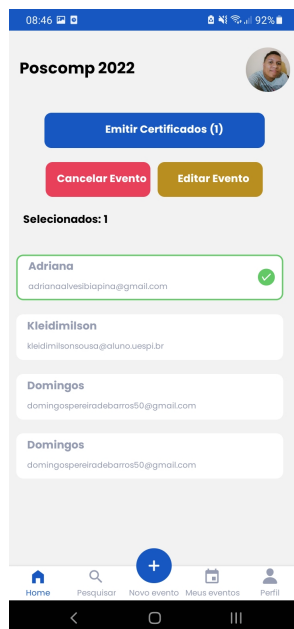
(b) Tela inicial do organizador.



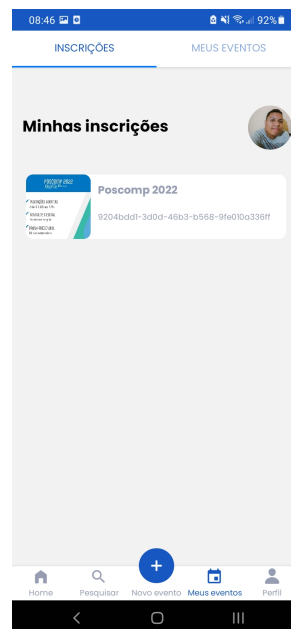
(c) Tela para pesquisar um evento.



(d) Tela para criar um evento.

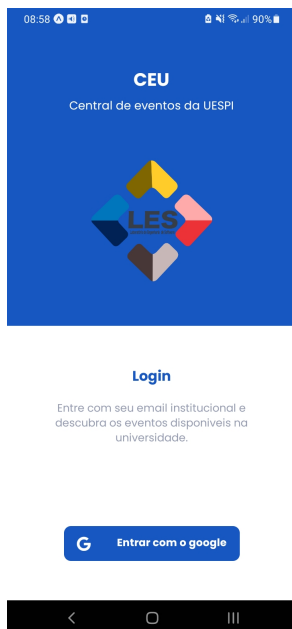


(e) Tela para gerenciar um evento.

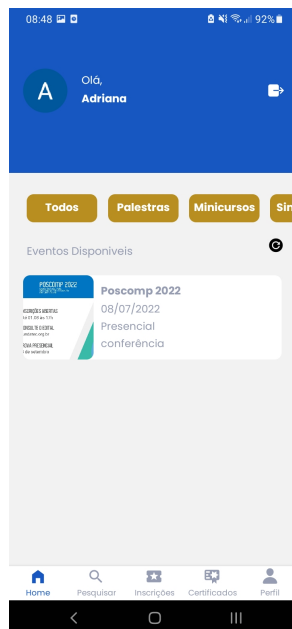


(f) Tela de inscrições do organizador.

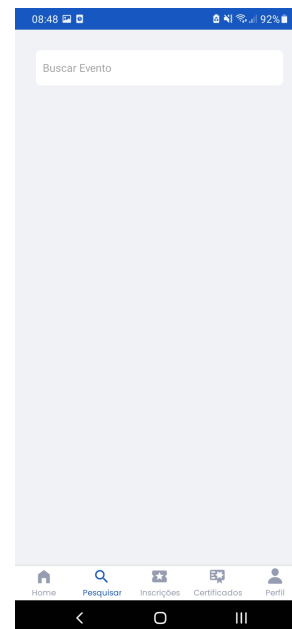
Figura 11. Telas do organizador



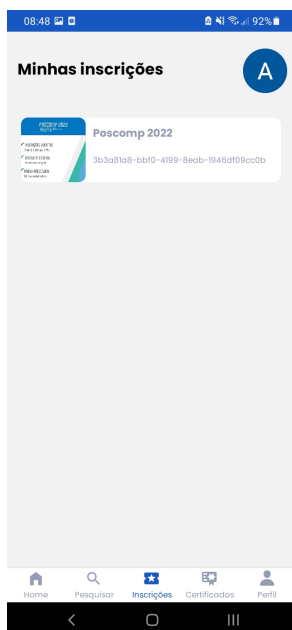
(a) Tela de login.



(b) Tela inicial do participante.



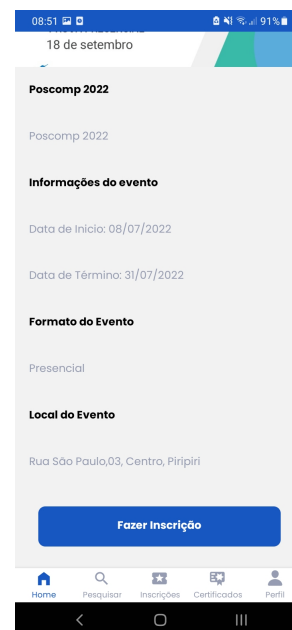
(c) Tela de pesquisa de eventos



(d) Tela de inscrições



(e) Tela do evento



(f) Tela do evento

Figura 12. Telas do participante .

Poscomp 2022

Certificado

Kleidimilson



Piripiri(PI), 08/07/2022

3b3a81a8-bbf0-4199-8eab-1946df09cc0b

A Universidade Estadual do Piauí - UESPI certifica que
KLEIDIMILSON participou com sucesso do Poscomp
2022, com aproveitamento de 120h

Figura 13. Modelo do certificado.

Tabela 7. Heurísticas de Nielsen

#	Heurística	Descrição
H01	Visibilidade do estado do sistema	O sistema mantém o usuário sempre informado sobre o que está acontecendo no mesmo.
H02	Consistência de padrões	Evitar que o usuário tenha que pensar se ações ou situações diferentes significam a mesma coisa.
H03	Flexibilidade e eficiência de uso	Fornece opções que otimizam a experiência de usuário (atalhos, teclas de funções, abreviações)
H04	Estética e Design minimalista	Evita o uso de informações irrelevantes.
H05	Compatibilidade do sistema e o mundo real	O sistema utiliza uma linguagem aos usuários, em vez de termos técnicos e específicos.
H06	Reconhecimento de erros	Utilizar linguagem simples para apresentar os erros e mostrar como contorná-los.
H07	Controle do usuário e liberdade	Oferece saída de emergência, permitindo que os usuários saiam facilmente de situações inesperadas.
H08	Reconhecimento e memorização	Fazer com que as ações e opções, sejam visíveis, o usuário não tem a obrigação de lembrar de tudo
H09	Prevenção de erros	Prevenir a ocorrência de erros.
H10	Ajuda e documentação	Fornece informações que podem ser facilmente encontradas e orienta os usuários.

7. Trabalhos Futuros

Como trabalhos futuros, pode-se citar:

Tabela 8. Severidade

#	Severidade	Descrição
S01	Problema cosmético/superficial	Não precisa ser corrigido, somente se tiver um tempo.
S02	Problema de usabilidade pequeno	Esse problema possui uma baixa severidade para ser solucionado.
S03	Problema de usabilidade grande	Importante solucionar o problema. Alta prioridade
S04	Catástrofe de usabilidade	Imprescindível solucionar o problema antes que o produto seja liberado para comercialização

Tabela 9. Análise Heurística do aplicativo

Problema	Violação	Severidade
Falta de ícones de carregamento	H01	S02
Não alerta a falta de eventos no sistema	H01	S01
Não mostra a data selecionada	H01	S03

- Corrigir os problemas das interfaces identificados na análise heurística.
- Migrar a API para os servidores da UESPI.
- Implementação de uma solução Web para a gestão dos eventos. O aplicativo desenvolvido, utiliza uma API. Desse modo, é possível consumir essa API em uma aplicação Web.
- Permitir a organização do evento por vários organizadores. O aplicativo desenvolvido, possibilita apenas que um organizador cuide de todo o gerenciamento do evento.
- Permitir a confirmação de presença nos eventos através de um Qrcode individual para cada participante. O aplicativo desenvolvido, possibilita apenas que um organizador selecione os participantes para emissão dos certificados.

Referências

- Barbosa, U. and Neves, G. (2022). Doity. Disponível em: <https://www.doity.com.br/>. Acesso em: 15 maio de 2022.
- Blomkvist, S. (2002). Persona—an overview. *Retrieved November, 22:2004*.
- do Ceará, U. F. (2022). Sistema de eventos. Disponível em: <https://eventos.sobral.ufc.br/autenticar>. Acesso em: 15 março de 2022.
- Eisenman, B. (2015). *Learning react native: Building native mobile apps with JavaScript*. "O'Reilly Media, Inc."
- em Tecnologia, L. S. (2022). Even3. Disponível em: <https://www.even3.com.br/>. Acesso em: 03 junho de 2022.
- FERREIRA VIEIRA, J. (2021). Eventoservice: Aplicação para auxílio à organização de eventos acadêmicos. B.S. thesis.
- Masse, M. (2011). *REST API design rulebook: designing consistent RESTful web service interfaces*. "O'Reilly Media, Inc."

- Meta (2015). React native. Disponível em: <https://reactnative.dev/>. Acesso em: 02 abril de 2022.
- Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 152–158.
- Postgres (1989). Postgres. Disponível em: <https://www.postgresql.org/docs/current/intro-what-is.html>. Acesso em: 10 abril de 2022.
- Schwaber, K. (1997). Scrum development process. In *Business object design and implementation*, pages 117–134. Springer.
- Silvers, J. R. (2005). The potential of the embok as a risk management framework for events. In *Conference Proceedings, 2005 Las Vegas International Hospitality and Convention Summit*, pages 1–16. University of Nevada Las Vegas.
- UESPI (2021). Resolução CEPEX 002/2021. Disponível em: <https://drive.google.com/file/d/16jWcXJsjtRAqvNQqs8opt2B-p4X752Gg/view?usp=sharing>. Acesso em: 10 janeiro de 2022.
- Valente, M. T. D. O. et al. (2020). *Engenharia de software moderna: princípios e práticas para desenvolvimento de software com produtividade*. Universidade Federal de Minas Gerais.