

Manoel Gomes Teixeira

**Desenvolvimento do Frontend do aplicativo
Agenda Cultural de Vilhena**

Vilhena - RO

2021

Manoel Gomes Teixeira

Desenvolvimento do Frontend do aplicativo Agenda Cultural de Vilhena

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – campus Vilhena, realizado em cumprimento de requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – IFRO
Campus Vilhena
Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas

Orientador: Juliano Fischer Naves

Vilhena - RO
2021

FICHA CATALOGRÁFICA

Biblioteca IFRO – Campus Vilhena

T266d

TEIXEIRA, Manoel Gomes

Desenvolvimento do Front End do aplicativo agenda cultural de Vilhena /
Manoel Gomes Teixeira – Vilhena, Rondônia, 2021.

57f. ; il.

Orientador Prof. Dr. Juliano Fischer Naves

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento
de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia
- IFRO

1. Agenda cultural 2.Eventos 3. Informação cultural I. Instituto Federal
de Educação, Ciência e Tecnologia de Rondônia – IFRO II. Título

005.133

Bibliotecária responsável: Rosilene Maria do Couto Marques

Manoel Gomes Teixeira

Desenvolvimento do Frontend do aplicativo Agenda Cultural de Vilhena

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – campus Vilhena, realizado em cumprimento de requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Trabalho aprovado. Vilhena - RO, 13 de dezembro de 2021

Dr. Juliano Fischer Naves

Orientador

Esp. Gilberto Pereira da Silva

IFRO - Campus Vilhena

M.Sc. Roberto Simplício Guimarães

IFRO - Campus Vilhena

Vilhena - RO

2021

Este trabalho é dedicado a todos os que não deixam de buscar seus objetivos mesmo depois de parecerem perdidos.

Agradecimentos

Em primeiro lugar, a Deus por permitir alcançar esse objetivo. A minha família que me ajudou a não desistir, ao meu orientador, professor Juliano, por me ajudar e orientar na construção do projeto e demais professores por todo o conhecimento que compartilharam durante o curso.

RESUMO

Eventos Culturais são importantes para o desenvolvimento local, comercial e também para a manutenção de uma identidade local através de costumes e tradições adquiridas. Este trabalho teve como objetivo criar uma plataforma que facilita a divulgação de eventos culturais por meio de um aplicativo disponível para o público. O projeto teve o objetivo de desenvolver o *Front End* da aplicação, que é a parte visual para o usuário interagir. O desenvolvimento foi realizado buscando fornecer ao usuário uma aparência simples e objetiva para suas informações. Através de entrevistas foram determinadas as necessidades do usuário e também do organizador do evento, onde foi desenvolvido um painel para que fosse possível criar um evento. As telas foram desenvolvidas e atenderam o objetivo proposto.

Palavras-chave: Agenda Cultural; Eventos; Informação Cultural.

ABSTRACT

Cultural Events are important for local and commercial development and also for the maintenance of a local identity through acquired customs and traditions. This work aimed to create a platform that facilitates the dissemination of cultural events through an application available to the public. The project aimed to develop the application's FrontEnd, which is the visual part for the user to interact with. The development was carried out seeking to provide the user with a simple and objective appearance for their information. Through interviews, the needs of the user and the event organizer were determined, where a panel was developed so that it was possible to create an event. The screens were developed and met the proposed objective.

Keywords: Cultural Agenda; Events; Cultural Information.

Lista de ilustrações

Figura 1 – Ciclo de vida	27
Figura 2 – Fase de iniciação	28
Figura 3 – Fase de execução	29
Figura 4 – Fase de finalização	30
Figura 5 – Arquitetura Macro	37
Figura 6 – Interação usuário e Aplicativo	37
Figura 7 – Interação organizador e Aplicativo	38
Figura 8 – Diagrama de Sequência	39
Figura 9 – Diagrama de Classe	40
Figura 10 – Diagrama de Sequência Organizador	41
Figura 11 – Gráfico de Burndown	45
Figura 12 – Teste de Componente - exemplo 1	46
Figura 13 – Documentação Usuário Final - Página Categoria	48
Figura 14 – Documentação Usuário Final - Registro de Usuário	49
Figura 15 – Demonstração de Software - Cadastro de Eventos	51
Figura 16 – Demonstração de Software - Cadastro de Categoria	52
Figura 17 – Demonstração de Software - Alterar dados organizador	52
Figura 18 – Demonstração de Software - Aplicativo Início	53
Figura 19 – Demonstração de Software - Aplicativo Filtrar por Data	54
Figura 20 – Demonstração de Software - Aplicativo Categoria de Eventos	55
Figura 21 – Demonstração de Software - Aplicativo Eventos Anteriores	56

Lista de quadros

Quadro 1 – Requisitos Funcionais - Organizador	34
Quadro 2 – Requisitos Funcionais - Usuário	35
Quadro 3 – Requisitos Não Funcionais - Usuário	35
Quadro 4 – Sprints Desenvolvidas	44
Quadro 5 – Organização dos Testes	47
Quadro 6 – Resultado dos testes	47

Lista de abreviaturas e siglas

API - Application Programming Interface

MVC - Model View Controller

Sumário

1 INTRODUÇÃO	21
1.1 CONTEXTO E PROBLEMA.....	21
1.2 OBJETIVOS.....	21
1.2.1 Objetivo geral	21
1.2.2 Objetivos específicos	21
1.3 JUSTIFICATIVA.....	21
2 FUNDAMENTAÇÃO TEÓRICA	22
2.1 EVENTOS CULTURAIS E O DESENVOLVIMENTO DA SOCIEDADE.....	22
2.2 A TECNOLOGIA COMO MEIO PARA APROXIMAR A POPULAÇÃO DOS EVENTOS LOCAIS.....	23
2.3 TRABALHOS SIMILARES.....	24
3 MATERIAIS E MÉTODOS	25
3.1 CICLO DE VIDA E PROCESSOS.....	25
3.2 FERRAMENTAS E TECNOLOGIAS UTILIZADAS.....	28
3.2.1 Ferramentas	29
3.2.2 Tecnologias	30
3.3 REQUISITOS.....	32
3.4 ARQUITETURA DO SOFTWARE.....	34
3.5 MODELAGEM.....	36
3.6 PLANO DE TESTES.....	40
3.7 LICENÇA DE USO.....	40
4 RESULTADOS E DISCUSSÕES	41
4.1 GERENCIAMENTO DE CONFIGURAÇÃO E MUDANÇAS.....	41
4.2 PROCESSO DE DESENVOLVIMENTO.....	41
4.3 RELATÓRIO DOS TESTES.....	43
4.3.1 Testes realizados	43
4.4 DOCUMENTAÇÃO.....	46
4.4.1 Documentação para usuários finais	46
4.4.2 Documentação para desenvolvedores	48
4.5 IMPLANTAÇÃO.....	48
4.5.1 Ambiente para solução	48
4.6 DEMONSTRAÇÃO DO SOFTWARE.....	48
4.6.1 Sistema WEB	48
4.6.2 Aplicativo	51
5 CONSIDERAÇÕES FINAIS	55
ANEXO A – Licença MIT	57

1 INTRODUÇÃO

1.1 Contexto e problema

Na cidade de Vilhena - RO, os eventos culturais são divulgados em diferentes fontes de informação, podendo ser em sites de notícias, redes sociais, rádio, etc. A maioria dos eventos divulgados são patrocinados por empresas. Acredita-se que eventos menores, ou em específico, eventos da Fundação Cultural de Vilhena, fiquem prejudicados pelo fato de que o usuário precise pesquisar em diferentes fontes para saber sobre eles ou sua divulgação abrange um pequeno grupo de pessoas.

A divulgação inadequada ou a falta de uma fonte de informação específica para divulgar eventos, pode ser um dos fatores responsáveis pelo baixo número de participantes por parte da sociedade.

Sendo assim, este trabalho pretende responder a seguinte pergunta: Como facilitar a divulgação de eventos culturais na cidade de Vilhena de forma a aumentar a participação popular?

1.2 OBJETIVOS

1.2.1 Objetivo geral

Facilitar a divulgação de eventos culturais que acontecem na cidade através de um aplicativo que concentre essas informações.

1.2.2 Objetivos específicos

- Levantar bibliografia das tecnologias utilizadas para criação do aplicativo;
- Analisar requisitos necessários para o desenvolvimento;
- Desenvolver documentação para o aplicativo;
- Prototipar sistema.

1.3 JUSTIFICATIVA

Um aplicativo específico para a divulgação de eventos culturais em Vilhena, permite aos organizadores uma melhor divulgação de seus eventos para a população em geral, pelo fato de que um aplicativo pode ser instalado no telefone e chegar a todas as pessoas, além de proporcionar a divulgação de eventos menores ou gratuitos para o público, mais facilmente, que por vezes ficam sem saber sobre esses eventos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 EVENTOS CULTURAIS E O DESENVOLVIMENTO DA SOCIEDADE

Definir o que é cultura não é tarefa tão simples, uma vez que ela está ligada a mais de um elemento da sociedade. Para Brant, (2009), a cultura abrange não somente uma visão artística. Ele define que a cultura também recebe influência da economia local e das pessoas, que com seus próprios costumes influenciam constantemente as tradições de um lugar.

A cultura se expressa das mais diversas e diferentes formas, variando de acordo com a visão de pessoas e também dos locais onde se encontra. Segundo Matias (2011), a realização de eventos artísticos, crescendo muito nos últimos anos, tanto no Brasil como em outros países. Esse crescimento contribui para um desenvolvimento financeiro, social e cultural dos locais onde ocorrem esses eventos.

A realização do evento por si só, não contribui para a difusão de ideias ou valorização da cultura. Para que a realização dos eventos, tanto os culturais quanto os eventos esportivos, tecnológicos ou científicos, traga benefícios econômicos, promovam a cultura e o conhecimento para a sociedade, os organizadores devem ter como objetivo inicial alcançá-los e implementá-los desde sua fase de planejamento (BRITO & FONTES, 2002). Cabe também às autoridades políticas, incentivar o desenvolvimento cultural de sua região, a fim de garantir sua permanência ao longo do tempo, sem excluir nenhuma forma de expressão cultural, uma vez que essa diversidade é um patrimônio da humanidade e deve ser protegido os direitos de todos expressarem sua liberdade cultural de forma democrática.

Em 2004 foi assinado, em Barcelona, o documento que ficou conhecido como Agenda 21 da Cultura. A reunião teve a presença de representantes de vários países e cidades. O objetivo da Agenda 21 da Cultura é servir de orientação e base aos governantes de como devem ser feitas as políticas públicas voltadas para a área da cultural e também contribuir para a valorização da cultura humana ao longo do tempo.

A agenda 21 da Cultura tem como um de seus compromissos:

Estabelecer políticas que fomentem a diversidade cultural, a fim de garantir a amplitude da oferta e a presença de todas as culturas, especialmente das minoritárias ou desprotegidas, nos meios de comunicação e de difusão, incentivando as co-produções e os intercâmbios, e evitando posições hegemônicas.(Agenda 21 da Cultura, 2004, p.5)

A agenda 21 da Cultura se preocupa também com as expressões culturais que são consideradas minorias, esse fato se deve do monopólio cultural que vem crescendo no mundo devido principalmente a visão mais comercial da cultura em escala global, fazendo uma propagação de um padrão cultural que tem por objetivo vender produtos e valorizar somente uma região (BRANT, 2009).

Utilizar recursos tecnológicos para aumentar a visibilidade de um evento cultural já é uma prática muito utilizada, principalmente por meio de canais de televisão. Com o aumento da internet, porém, as opções de comunicação aumentaram exponencialmente, inclusive as formas de expressões artísticas, que se beneficiam da tecnologia para criar novos conceitos (GONÇALVES, 2009).

Apesar do uso da tecnologia contribuir para uma mudança na forma como a cultura pode ser consumida, tendo a opção de utilizar apenas um computador para tal, o hábito de participar de eventos culturais externos traz benefícios como uma melhor socialização do indivíduo, uma vez que sem as problemas internos, ele está mais relaxado e suscetível a novos aprendizados (PAGLIOTO & MACHADO, 2012).

2.2 A TECNOLOGIA COMO MEIO PARA APROXIMAR A POPULAÇÃO DOS EVENTOS LOCAIS

Nem sempre produzir e organizar todas as etapas anteriores a apresentação de um evento cultural é realizada por profissionais específicos, em eventos de menor porte essa tarefa fica sob a responsabilidade de pessoas que não são profissionais, ou muitas vezes é feita pelo próprio artista. Isso mostra que além do incentivo por parte do poder público, percebe-se que existe uma falha que dificulta a população de conhecer esses eventos menores (GADELHA & BARBADO, 2013).

Como abordado anteriormente, eventos culturais são melhor aproveitados de modo presencial, mas a tecnologia pode ser utilizada como um complemento de valorizar e divulgar esses eventos. Uma forma de melhorar a divulgação de eventos culturais nas cidades pode ser feita através de aplicativos de agenda cultura, que fornecem informações relevantes e precisas sobre os eventos que acontecem na cidade e aproximam a população dos eventos culturais ou turísticos de modo mais pessoal (PAES & LADEIRA, 2018).

Um aplicativo de agenda cultural deve ter como objetivo fornecer informações precisas, afinal o usuário passa a ter todas as opções e eventos na palma da mão, mas também deve possuir uma aparência prática, que permite sua usabilidade por parte dos usuários, sem dificuldades de navegação (Rivero et al., 2014).

O objetivo deste trabalho é desenvolver o Frontend para um aplicativo de Agenda

Cultural, em específico, a Fundação Cultural de Vilhena, que realiza projetos de cultura na cidade. Além de fornecer à população, uma fonte de informação mais detalhada dos eventos realizados pela Fundação Cultural.

2.3 TRABALHOS SIMILARES

A seguir são descritos alguns sistemas de agenda cultural que existem no mercado. Alguns têm destaque mais para eventos online e outros somente para divulgação do evento em si. Vale destacar que não foi possível analisar a tela administrador do sistemas encontrados devido a alguns serem pagos, sendo assim foram analisadas as características do sistema.

Os seguintes softwares proprietários, para gerenciamento de eventos, tanto culturais ou não foram encontrados:

- Fourpass¹: Plataforma para criação e gerenciamento de eventos, fornece dois tipos de categorias. Grátis ou Pagos. Caso o evento seja pago, a plataforma desconta uma porcentagem dos ingressos. Fourpass cobra uma porcentagem do valor dos ingressos.
- Even3²: Plataforma para criação e gerenciamento de eventos, fornece dois tipos de categorias: Grátis ou Pagos. Caso o evento seja pago, o criador do evento deve pagar uma porcentagem para a empresa que desenvolveu o sistema.
- SGE³: Sistema para gerenciamento de eventos e formaturas. O usuário paga para utilizar o sistema criar seus eventos.

Os seguintes softwares de código aberto, para gerenciamento de eventos de instituições de ensino, foram encontrados durante esta pesquisa:

- Odo⁴: É uma plataforma que fornece um conjunto de aplicativos para gerenciamento de negócios, em código aberto, que busca atender as necessidades da empresa onde inúmeras atividades podem ser utilizadas pela plataforma.
- Sympla⁵: Reúne eventos de várias categorias. Está disponível para aplicativo nas plataformas Android e iOS, além de ter um site. O aplicativo também possui um campo para eventos grátis.
- Bitrix24⁶: Reúne eventos de várias categorias. Está disponível para aplicativo nas plataformas Android e iOS.

1. Disponível em <https://www.fourpass.com.br/>

2. Disponível em <https://www.even3.com.br/>

3. Disponível em <https://www.sge.com.br/>

4. Disponível em <https://www.odoo.com.br>

5. Disponível em <https://www.sympla.com.br/>

6. ⁶ Disponível em <https://www.bitrix24.com.br/>

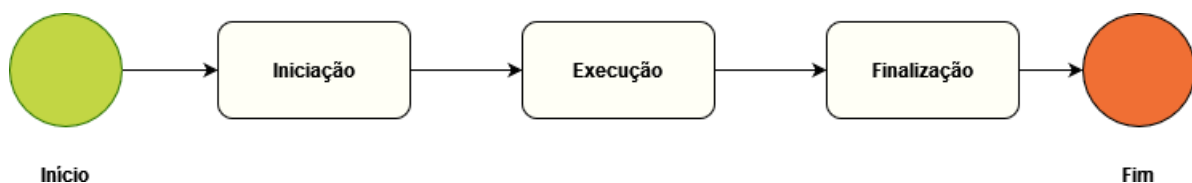
3 MATERIAIS E MÉTODOS

O capítulo descreve os materiais e métodos utilizados que foram utilizados para o desenvolvimento do front-end.

3.1 CICLO DE VIDA E PROCESSOS

Para o desenvolvimento do projeto, foi seguido o ciclo de vida conforme mostra a figura 3. O ciclo de vida está dividido em 3 etapas que são elas: Iniciação, Execução e Finalização. O desenvolvimento do painel administrador e do aplicativo, seguiram o mesmo ciclo de vida.

Figura 1 – Ciclo de vida

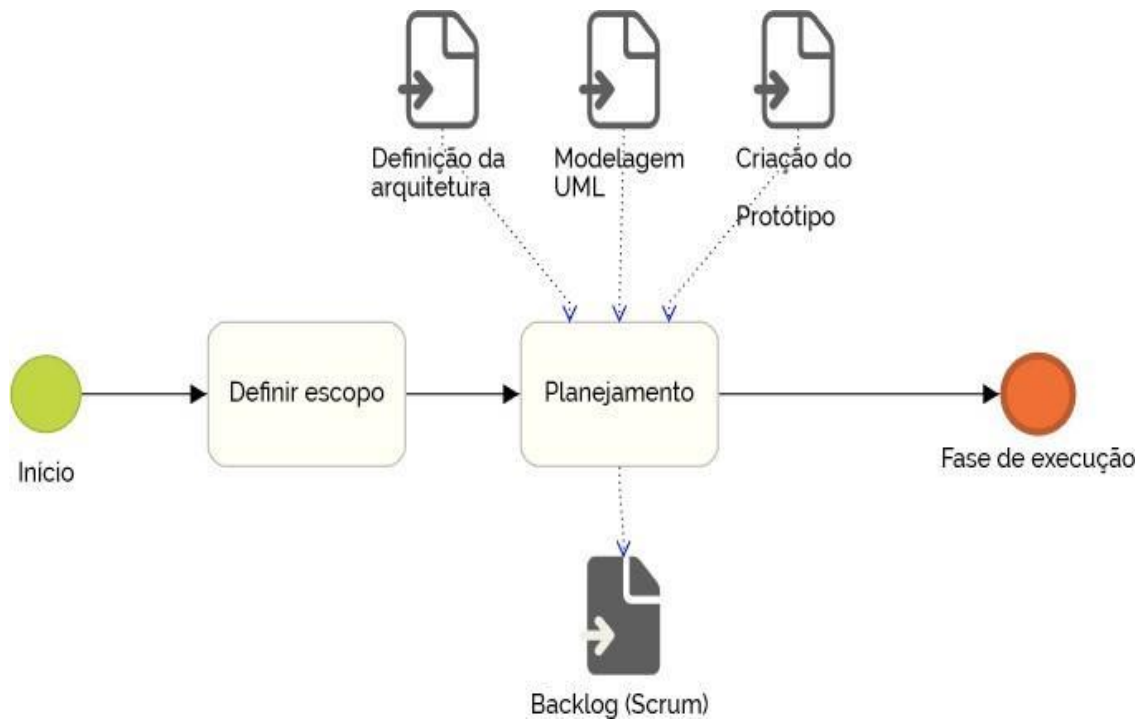


Fonte: elaborado pelo autor

A fase de iniciação, Figura 2, página 27, buscou identificar qual o objetivo seria alcançado com o desenvolvimento do projeto, bem como o público que seria afetado por ele. Nessa fase também foi analisada a questão de viabilidade, onde questões como: Podemos fazer? e Devemos fazer? foram levantadas e analisadas, entrevistas foram realizadas para uma melhor compreensão do projeto e seu impacto. Definido o escopo do trabalho, seguiu a subetapa de planejamento, nela, os prazos foram definidos, recursos foram analisados, e os seguintes itens foram realizados: definição da arquitetura, modelagem UML, criação do protótipo.

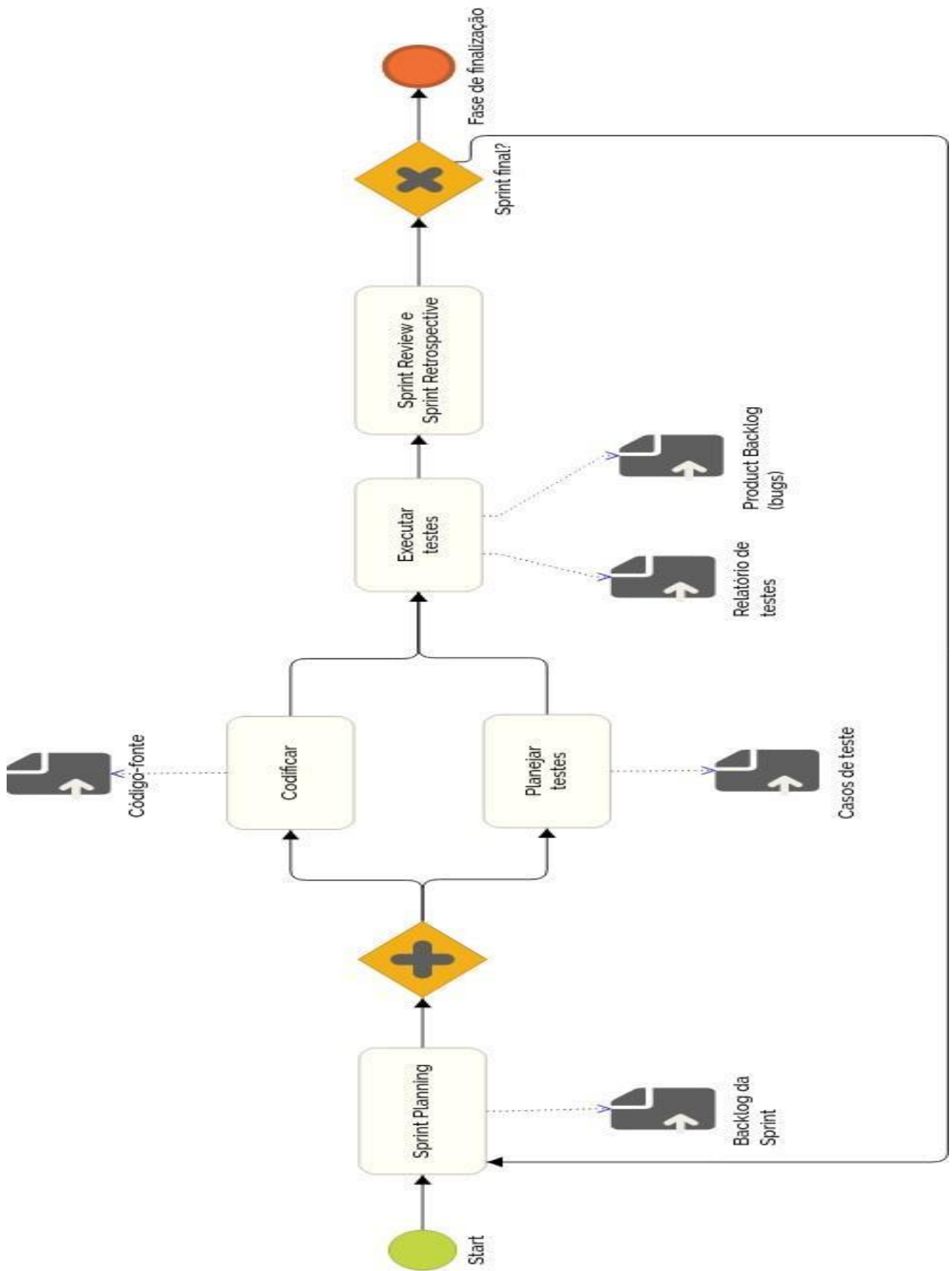
A fase de execução deu início ao desenvolvimento do Front-end do sistema de agenda cultural. Primeiramente o desenvolvimento das telas que estavam relacionadas ao painel administrador do sistema, obedecendo os requisitos que foram levantados anteriormente. Posteriormente foram desenvolvidas as telas do aplicativo agenda cultural. Nessa fase o progresso do desenvolvimento é analisado e ajustes são feitos de acordo com a necessidade conforme mostra a Figura 3, cada etapa do desenvolvimento obedecia a um prazo determinado através de uma sprint.

Figura 2 – Fase de iniciação



Fonte: elaborado pelo autor

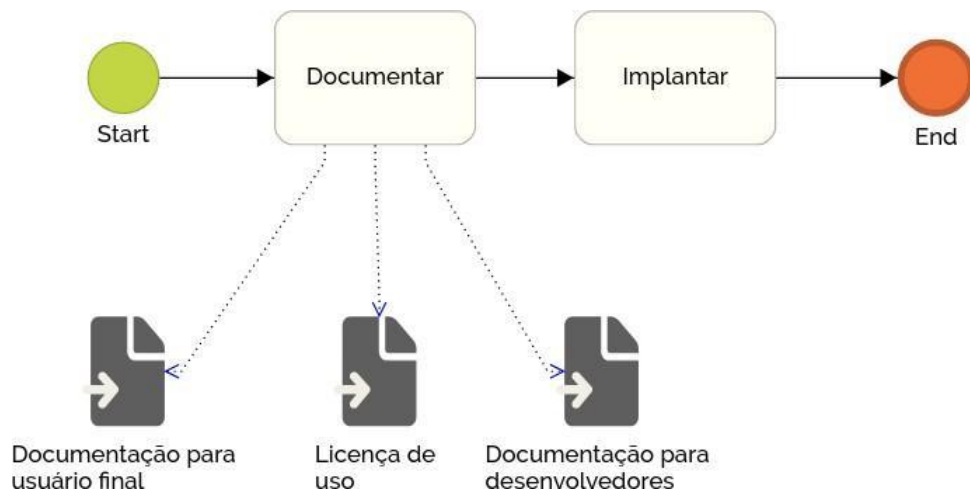
Figura 3 – Fase de execução



Fonte: elaborado pelo autor

A fase de finalização, Figura 4 entregou o projeto desenvolvido que compreende o Front-end do aplicativo agenda cultura. Não houve integração com o back-end, uma vez que essa etapa não estava prevista no projeto descrito. Apesar do projeto ficar pronto, as sprints, durante a fase de execução, não foram seguidas de acordo com o prazo. Sendo o projeto tendo sua maior parte desenvolvido sem a observância dos prazos estipulados anteriormente.

Figura 4 – Fase de finalização



Fonte: elaborado pelo autor

3.2 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Para o desenvolvimento do Front-end, foram utilizadas as seguintes ferramentas e tecnologias:

3.2.1 Ferramentas

Visual Studio Code - VS Code: é um editor de código. Ele é desenvolvido pela Microsoft, tem código aberto e é gratuito para usar. Está disponível para os sistemas Windows, Mac e Linux, o que o torna uma excelente opção para ser utilizada. O sistema permite que uma grande quantidade de linguagens de programação sejam utilizadas por ele, permitindo que ele seja amplamente utilizado para vários tipos de projetos.

O sistema permite a instalação de inúmeras extensões que auxiliam no desenvolvimento do código, possibilitando uma personalização pessoal ou dependendo do projeto desenvolvido. Ainda permite a criação de extensões por parte do programador e possui uma interface simples que ajuda no desenvolvimento. O Visual Studio Code, também possui uma integração com o Git, facilitando o envio do código para o repositório.

Uma desvantagem do Visual Studio Code, está relacionada a dificuldade de personalização inicial para o desenvolvimento do projeto, muitas funcionalidades não vem na instalação padrão. O sistema possui muitas extensões e isso faz com que o usuário perca muito tempo pesquisando as extensões que ele pode utilizar dependendo do projeto, além do fato de existirem muitas extensões com funções semelhantes.

Expo: é uma ferramenta utilizada para o desenvolvimento mobile com a linguagem React Native. Entre as vantagens de utilizar o Expo para o desenvolvimento de aplicações, está o fato dele poder ser utilizado com o Visual Studio Code para edição de código, possuir uma fácil conexão com APIs nativas dos dispositivos. Permitindo por exemplo o fácil acesso da câmera ou microfone, sem a necessidade de instalar dependências ou fazer alteração de código.

Possui vantagem para a criação de sistemas mais simples além de ser uma ótima ferramenta de aprendizado. Sua instalação é bem intuitiva e rápida. Outra vantagem é que o Expo possui um aplicativo para Android ou iOS que permite visualizar como está ficando o sistema desenvolvido, sem a necessidade de instalar nenhum programa no computador. A conexão com as APIs pode por outro lado ser uma desvantagem para quem está iniciando no desenvolvimento mobile, uma vez que o Expo esconde como é feita a conexão com recursos do aparelho são ocultadas.

3.2.2 Tecnologias

O desenvolvimento do projeto se deu em duas etapas. Primeiro foi desenvolvido o frontend do administrador do sistema, posteriormente o aplicativo para o usuário. Para o painel administrador foram usadas as seguintes tecnologias:

O React JS, é uma biblioteca front-end criada pelo Facebook. Essa biblioteca funciona através de componentes, conectando diferentes partes de uma página para um desenvolvimento mais facilitado. Os componentes reaproveitam código e ajudam a padronizar a interface. É um dos recursos mais úteis da biblioteca, além disso o React JS, é fácil de usar e de se compreender, além de permitir ser usado em conjunto com Typescript, como foi feito no projeto desenvolvido.

React JS também permite a criação de um arquivo JavaScript contendo HTML e CSS, essa característica pode ser muito útil para criar formatações, validações, ou até mesmo estilização. A diferença é que JSX usa camelCase assim *class* se transforma em *className* para ser utilizado.

Entre os pontos negativos encontrados está a dificuldade para quem está começando em utilizar justamente o JSX, pois a compreensão desse recurso leva tempo para ser entendido. Outro ponto negativo encontrado é a sua alta liberdade para definir um padrão de criação. Isso pode gerar um projeto pouco escalável se não for bem estruturado.

TypeScript: Criado pela Microsoft, é uma linguagem de código aberto que teve como base o JavaScript. Sua função é adicionar tipagem estática ao JavaScript, permitindo assim que uma variável ou função possa assumir tipos diferentes durante sua execução. TypeScript é usado somente durante o desenvolvimento, na compilação do código ele é convertido para JavaScript. Isso ocorre pelo fato do browser entender somente JavaScript. Ao utilizar TypeScript é possível, por exemplo, ter erros sinalizados ainda durante o desenvolvimento do código, diferente do JavaScript que os mostra somente durante a execução do mesmo.

Combinado com o React, o TypeScript é uma poderosa ferramenta para o desenvolvimento FrontEnd, ele permite a organização do código, além de que quando é combinado com o IntelliSense, sua precisão é muito grande.

Uma desvantagem do TypeScript é sua adaptação para quem já programa com JavaScript, é recomendado uma mudança gradual. A linguagem também pode sofrer com alguma falta de documentação de uma biblioteca.

React Native: Framework criado pelo Facebook, sua função é a de permitir a criação de aplicativos multiplataforma (Android e iOS). React Native é baseado em React, a diferença entre eles é que enquanto o React JS é direcionado para web, independente do dispositivo utilizado, o React Native é específico para aplicativos móveis nativos.

O fato de ser específico para mobile traz algumas vantagens como por exemplo, um carregamento mais rápido da página, mais segurança, uma performance melhor do aplicativo. Além do fato de que uma aplicação desenvolvida para Android, pode ter praticamente todo o seu código reaproveitado para iOS, por exemplo, isso torna o desenvolvimento mais econômico e uma entrega mais rápida. Usar React Native também tem a vantagem de possuir um vasto conjunto de bibliotecas, documentação que auxiliam muito no desenvolvimento.

Uma das desvantagens do React Native é a sua dependência do Facebook que o atualiza. Caso isso acabe a linguagem irá desaparecer. Atualizações para componentes de hardware mais modernos também costumam chegar com atraso no React Native, apesar das constantes atualizações criadas. Outra desvantagem está relacionada à complexidade do aplicativo, aplicativos simples são melhores desenvolvidos em React Native. Aplicativos com recursos mais complexos tendem a reduzir o desempenho da aplicação.

3.3 REQUISITOS

Para o desenvolvimento do aplicativo Agenda Cultural de Vilhena, foi realizada uma entrevista com a presidente da Fundação Cultural de Vilhena, senhora Kátia Valéria. O objetivo da entrevista foi levantar requisitos que poderiam auxiliar no desenvolvimento do aplicativo proposto. Após a entrevista, foram organizadas as prioridades que o aplicativo deveria ter para atender de forma satisfatória os usuários desse tipo de sistema, tanto do lado do organizador do evento quanto do usuário. Também foi pesquisado nos aplicativos semelhantes, quais funcionalidades estavam presentes e quais delas se repetiam nos aplicativos. A entrevista também serviu de base para a criação das telas do organizador do evento.

Conforme o Quadro 1 (Requisitos do Organizador) e 2 (Requisitos do Usuário) mostram abaixo, os requisitos levantados para o organizador do evento buscam fornecer agilidade para a criação de um evento, além de permitir modificações nesse evento ou categoria. O organizador busca uma tela simples e funcional, que forneça informações de forma objetiva para ele, mas ao mesmo tempo permitindo que ele coloque o máximo de informações possíveis para um evento.

Para o usuário final, por outro lado, as telas do aplicativo devem ser construídas para fornecer a informação do evento de forma mais sintetizada possível, tendo uma organização dos elementos que seja harmoniosa e de fácil compreensão. Uma boa organização diminui o tempo para encontrar o que se procura.

Quadro 1 – Requisitos Funcionais – Organizador

Requisitos Funcionais - Organizador
Como organizador eu quero poder criar, excluir ou alterar um evento para colocar no aplicativo
Como organizador, eu quero poder cadastrar, excluir ou alterar uma categoria, para poder organizar os eventos
Como organizador, eu quero que meu sistema tenha login e logout, para ninguém acessar sem permissão

Fonte: Elaborado pela autor

Quadro 2 – Requisitos Funcionais - Usuário

Requisitos Funcionais - Usuário
Como usuário eu quero ver as categorias de eventos para uma pesquisa mais rápida.
Como usuário eu quero listar todos os eventos ligados a uma categoria.
Como usuário eu quero saber os detalhes de um evento, como data, horário, etc.
Como usuário eu quero pesquisar um evento por data, para saber o dia que posso ir.

Fonte: Elaborado pela autor

Os requisitos não funcionais obtidos para o aplicativo, são listados conforme mostra o Quadro 3 logo abaixo. Esses requisitos serviram de base para garantir uma boa funcionalidade de desempenho, disponibilidade do aplicativo de forma abrangente nos principais sistemas disponíveis no mercado. Observando os requisitos não-funcionais, observa-se que a questão da organização e leveza foi levantada principalmente pelo motivo do aplicativo poder conter muitas informações para a tela do usuário.

Quadro 3 – Requisitos Não Funcionais - Usuário

Requisitos Não Funcionais - Usuário
Como usuário eu quero que o aplicativo tenha um bom desempenho, seja rápido para atualizar novos eventos e live para download.
Como usuário, eu quero que o aplicativo seja bem organizado com seus componentes, tenha uma interface agradável e fácil de usar para a maioria das pessoas.
Como usuário, eu quero que o aplicativo funcione em dispositivos Android e iOS.
Como usuário eu quero pesquisar um evento por data, para saber o dia que posso ir.

Fonte: Elaborado pela autor

3.4 ARQUITETURA DO SOFTWARE

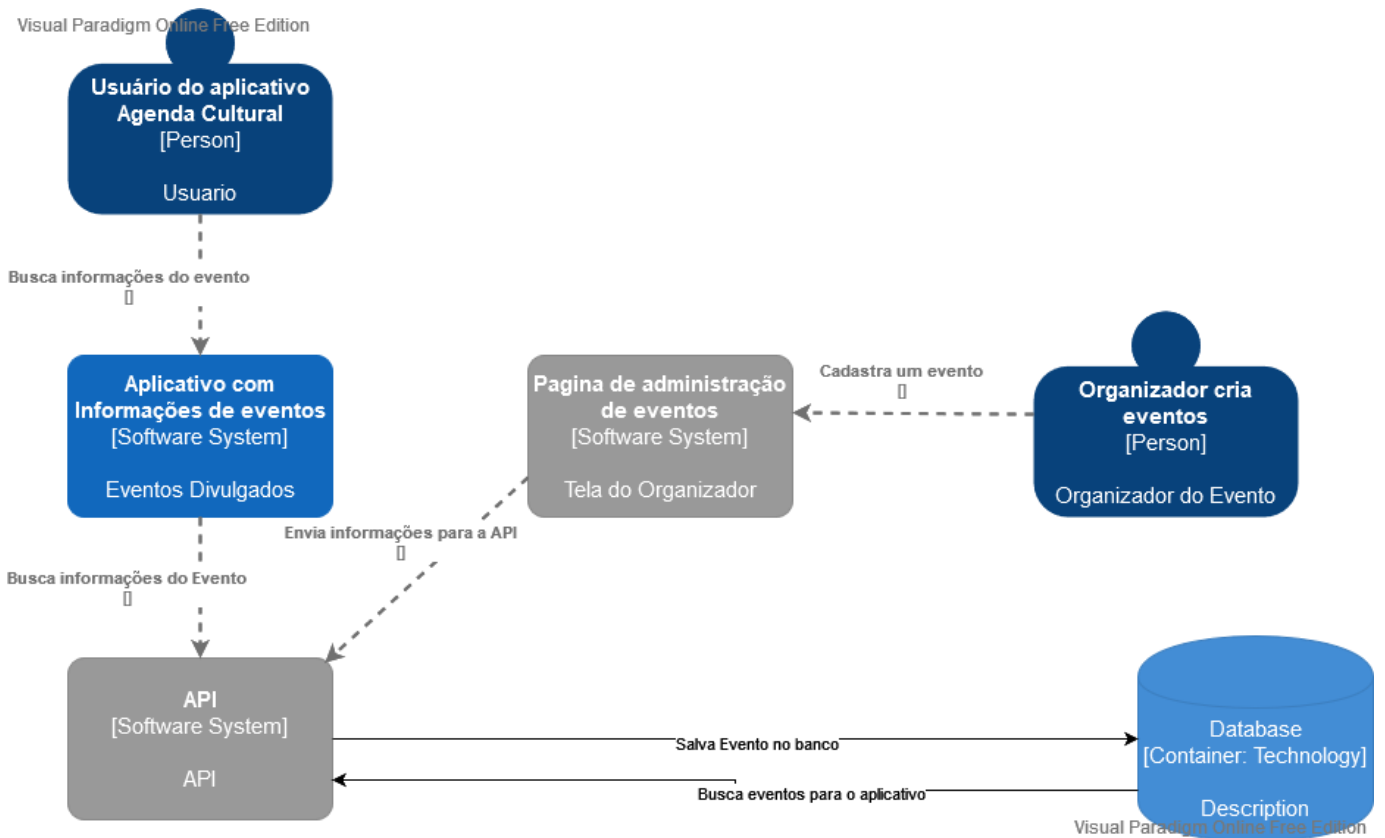
A arquitetura utilizada foi baseada no modelo MVC - (Model, View, Controller), que busca otimizar o envio de informações entre o usuário e o sistema. Essa arquitetura é dividida em três módulos ou componentes.

O View, é o componente que vai mostrar ao usuário as informações disponíveis no sistema. Esse módulo interage através de botões, telas, mensagens, ou seja, recursos visuais. O controller fica responsável pela intermediação da View com o Model. O Controller basicamente processa os dados e os transmite para a camada mais próxima. O Model, tem a função de controlar o comportamento dos dados por meio de regras definidas anteriormente, dependendo da informação recebida do Controller ele faz uma ação ou outra.

O desenvolvimento do projeto se divide em duas etapas, a primeira se refere ao desenvolvimento do FrontEnd do administrador dos sistema, com suas funções, recursos visuais e uma organização dos componentes em tela para melhor uso do sistema.

A segunda é o desenvolvimento do FrontEnd do aplicativo Agenda Cultural de Vilhena, com as funções que foram levantadas anteriormente. A Figura 5 mostra a arquitetura macro do sistema, sua sequência de organização incluindo o BackEnd da aplicação com a API e Banco de Dados para gravação das informações.

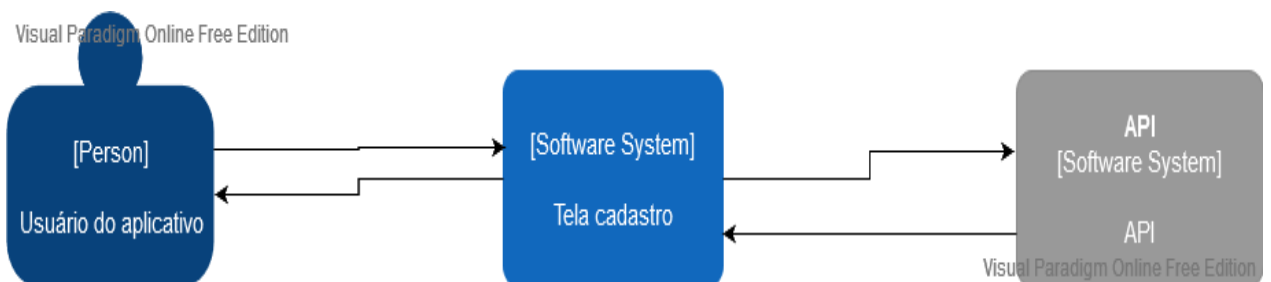
Figura 5 – Arquitetura Macro



Fonte: Elaborado pelo Autor

Com base na arquitetura MVC, segue de forma mais detalhada a interação do usuário com o sistema, através do componente View, conforme mostrado na Figura 6 abaixo.

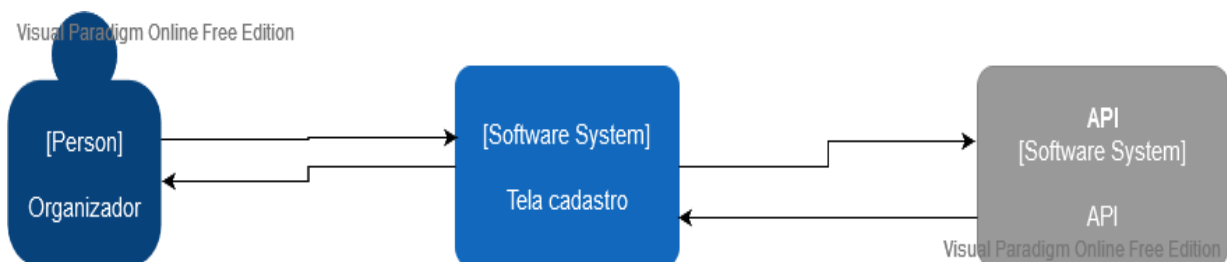
Figura 6 – Interação usuário e Aplicativo



Fonte: Elaborado pelo Autor

A interação do organizador será de forma semelhante, ao usuário, através do componente View ele envia suas solicitações e cabe a API desenvolvida receber essas solicitações e determinar qual o destino da informação. A Figura 7 detalha o uso pelo organizador.

Figura 7 – Interação Organizador e Aplicativo



Fonte: Elaborado pelo Autor

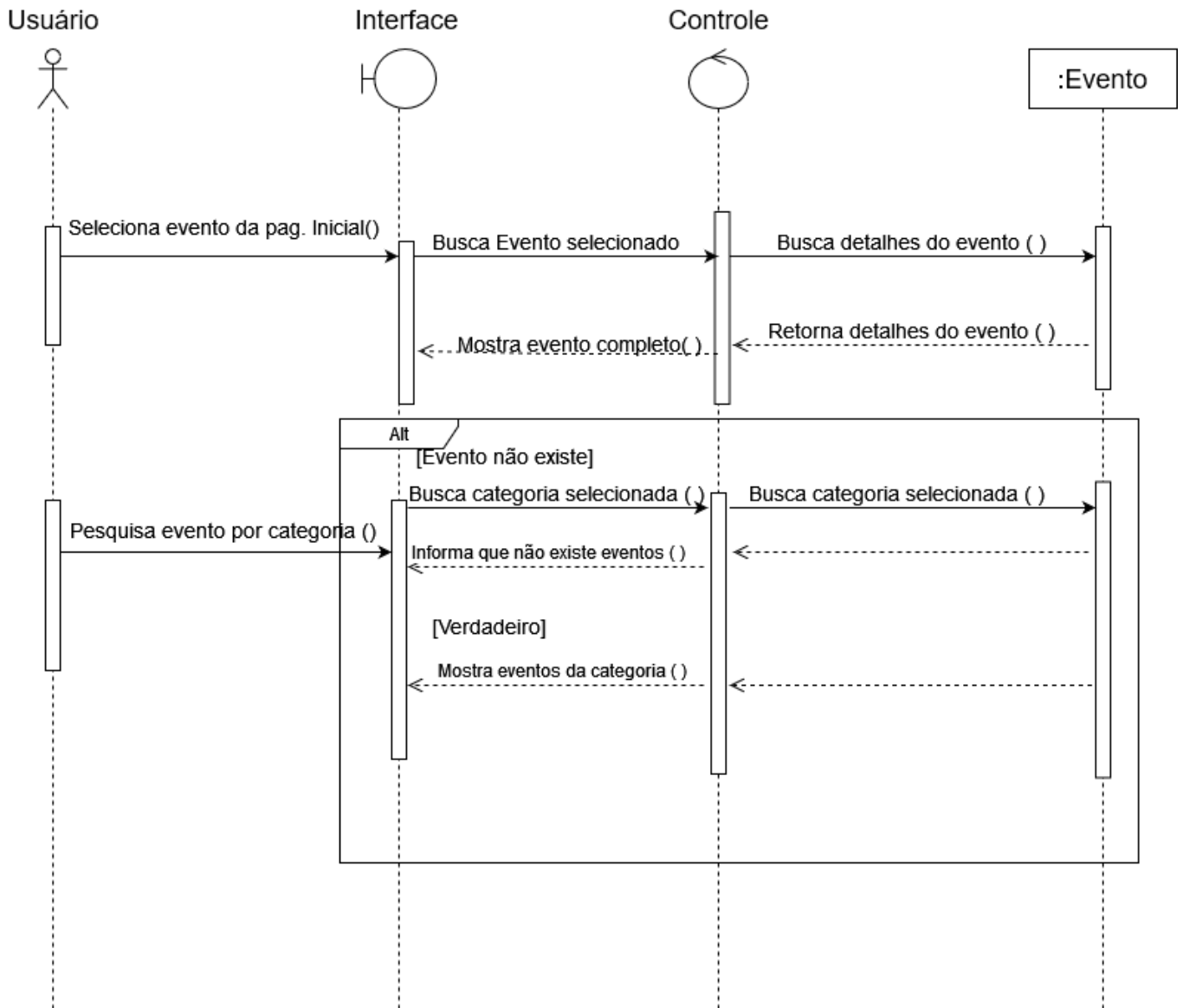
3.5 MODELAGEM

Foi utilizado a modelagem UML para descrever de forma visual o uso do aplicativo. A linguagem UML busca facilitar o entendimento de todos os responsáveis pelo desenvolvimento sobre a organização do software. Para tal, ela utiliza da construção de diagramas que representam uma etapa do sistema, uma função específica, etc.

A interação do usuário com o aplicativo, pode ser observada através do diagrama de sequência elaborado, conforme se observa na Figura 8. O usuário pesquisa um evento ou uma categoria de eventos, sua solicitação é encaminhada através da API que retorna do banco de dados as informações que existem sobre o evento selecionado ou retorna os eventos relativos a uma determinada categoria.

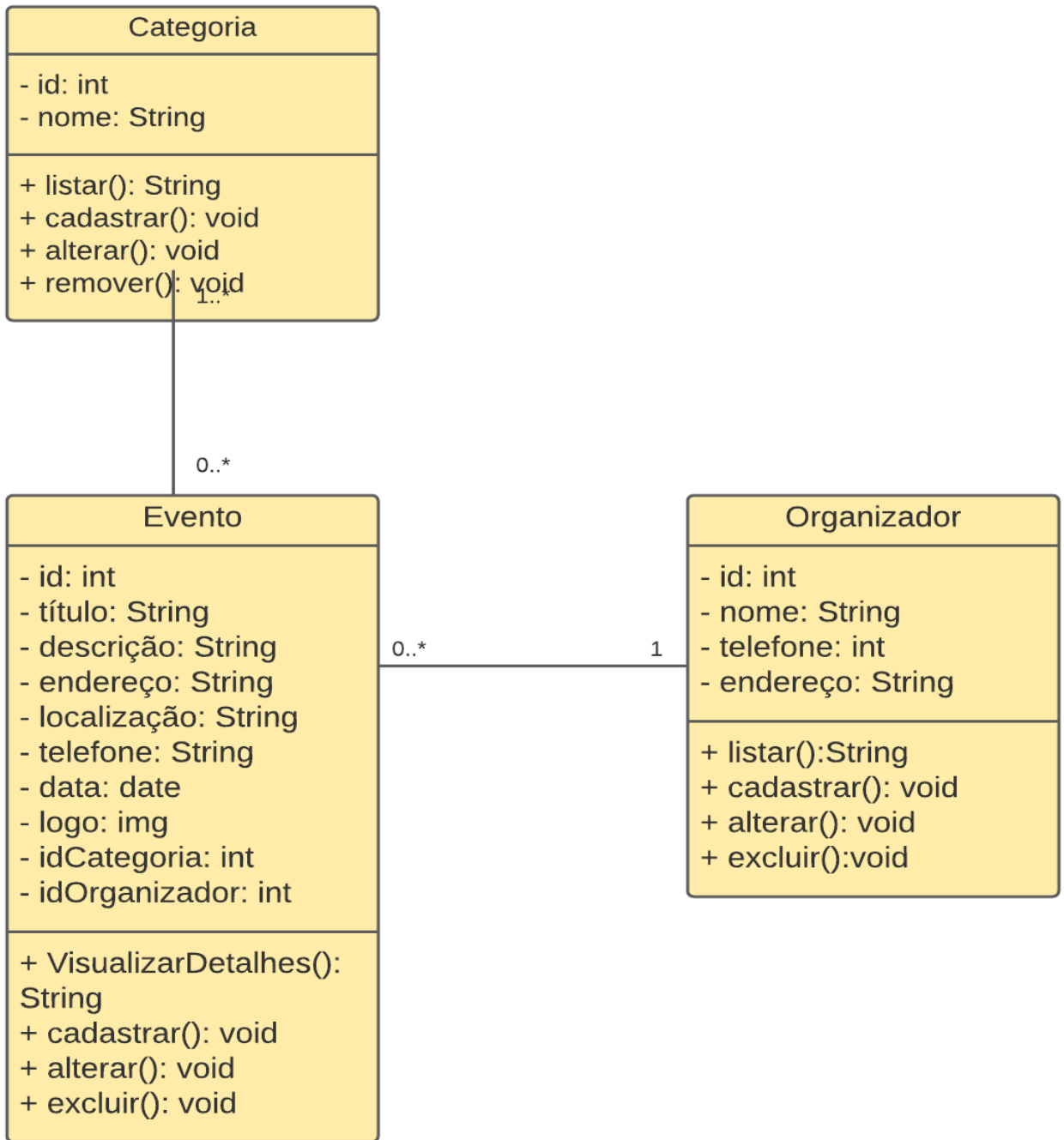
Para o evento, foi criado o diagrama de classe como é visto na Figura 9, o evento criado precisa estar vinculado a uma categoria de eventos, já a categoria, pode ter 1 (um) ou muitos eventos cadastrados. Cabe ao organizador criar o evento e a categoria, além de poder alterar e deletar um evento, Figura 10.

Figura 8 – Diagrama de Sequência



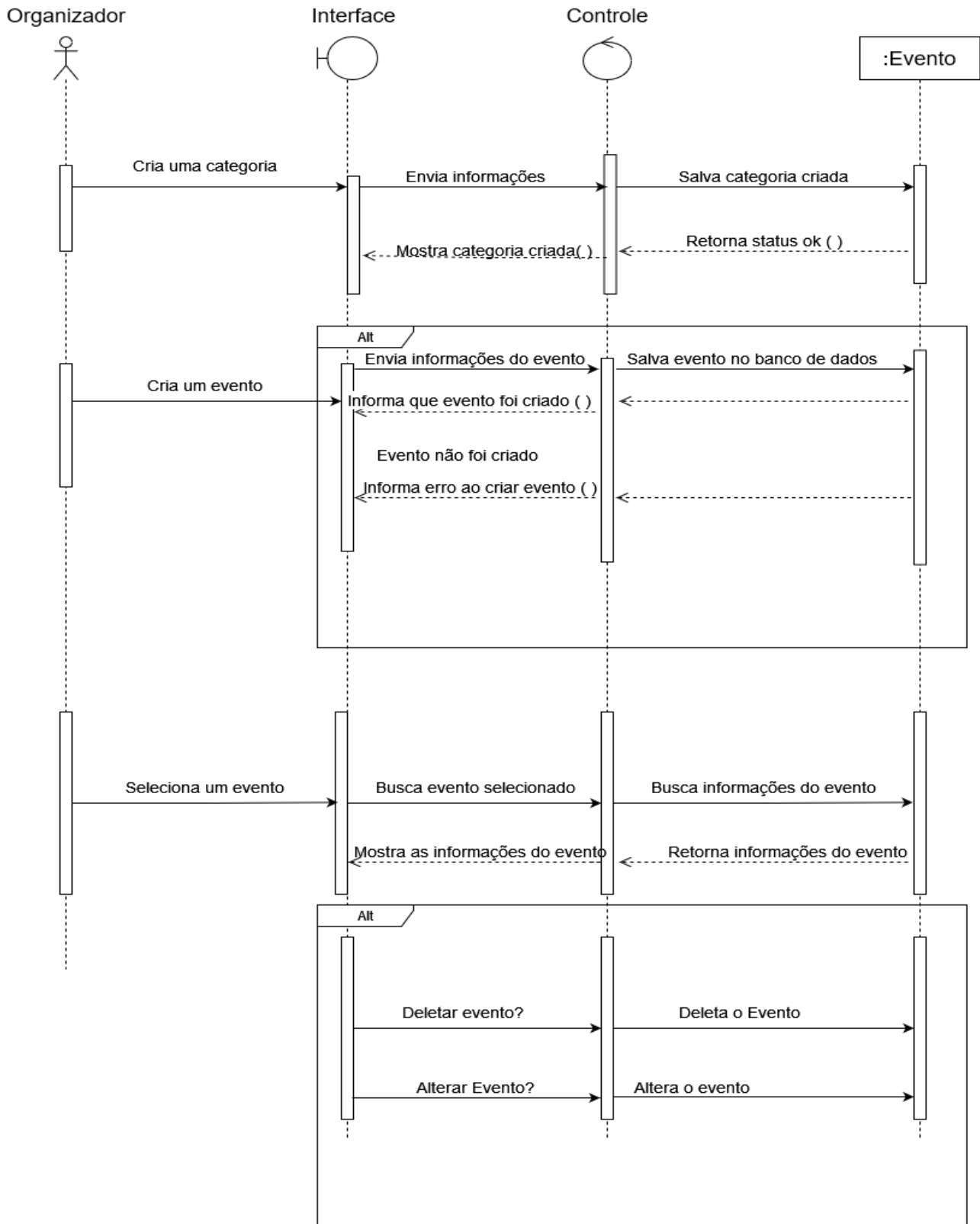
Fonte: Elaborado pelo Autor

Figura 9 – Diagrama de Classe



Fonte: Elaborado pelo Autor

Figura 10 – Diagrama de Sequência Organizador



Fonte: Elaborado pelo Autor

3.6 PLANO DE TESTES

O desenvolvimento do FrontEnd do aplicativo Agenda Cultural de Vilhena, teve seus testes realizados com a biblioteca React Testing Library ³, para a criação dos testes, com o objetivo de diminuir erros durante o desenvolvimento dos componentes. A biblioteca React Testing Library possui inúmeras ferramentas para testar componentes em React. Foram testados todos os componentes do Painel Administrador, para simular o uso do organizador do evento. O objetivo dos testes é identificar se os componentes criados eram renderizados para as páginas que eles estavam sendo exportados, possibilitando uma verificar o comportamento da página para o usuário.

Para determinar se um teste era aprovado ou não, foi criado um identificador dentro do componente a ser testado, após, criado o código de teste com objetivo de localizar tal identificador, se ele fosse localizado, significa que o componente foi lido com sucesso, caso não fosse ele retornaria um erro, que indicaria que o componente não está sendo renderizado.

Outro teste de FrontEnd realizado foi utilizando a extensão Lighthouse, disponível no Google Chrome, que é uma ferramenta criada com o objetivo de simular a experiência do usuário durante a navegação.

3.7 LICENÇA DE USO

O sistema desenvolvido é Open Source "código aberto". Nesse tipo de licença o usuário do sistema pode, além de poder utilizar o produto de forma gratuita, também pode ter acesso ao código fonte e modificá-lo. O termo de licença de uso se encontra anexo (A) a esse trabalho.

3 <https://testing-library.com/docs/react-testing-library/intro/>

4 RESULTADOS E DISCUSSÕES

4.1 GERENCIAMENTO DE CONFIGURAÇÃO E MUDANÇAS

O código do projeto do painel organizador ⁷ e do aplicativo ⁸ foi disponibilizado na plataforma GitHub, que permite uma hospedagem de código gratuita para desenvolvedores e fornece um grande conjunto de ferramentas para desenvolvimento colaborativo.

4.2 PROCESSO DE DESENVOLVIMENTO

A organização do projeto foi feita através da plataforma Taiga, que é uma ferramenta para gestão de projetos de software. A plataforma é gratuita e de código aberto, fornece uma interface simplificada e com bons recursos para gerenciar o desenvolvimento de software, inclusive é uma ferramenta muito recomendada para trabalhar em equipes, permitindo um monitoramento das tarefas por todos os integrantes. Primeiramente foram desenvolvidas as histórias de usuário referente ao organizador do sistema, por possuírem mais funções e páginas.

Após foram desenvolvidas o FrontEnd do aplicativo. O quadro 4 detalha as histórias de usuário criadas por sprint. O processo se desenvolveu inicialmente conforme planejado, Foram realizadas reuniões, encontros para detalhar a organização dos componentes em tela e a sua interface para o usuário. Estava planejado também, a integração do BackEnd desenvolvido a parte, com o FrontEnd deste projeto, porém, devido ao tempo desenvolvimento das páginas que sofreram diversas interrupções, a integração com a API, foi feita de forma parcial, utilizando o serviço Axios ⁹ que é uma biblioteca baseada em promessas para navegador e node.js. As funções de cadastrar organizador, login, logoff, criar categoria de eventos, alterar dados do organizador e deletar organizador, foram integradas com a API.

O scrum é um processo que permite o trabalho em equipe para o desenvolvimento de software. O scrum foi adaptado para uma pessoa, que realizaria todas as tarefas e supervisionava os prazos. Foram observadas durante o processo de desenvolvimento a não atualização das tarefas no tempo determinado, muitas foram desenvolvidas com atualizações na sprint fora do tempo, além de tarefas que foram desenvolvidas antes da criação da sprint em si. o número de sprints previstas também foi alterado, das 5 (cinco) sprints iniciais, foi acrescentada uma, totalizando 6 (seis), além dos pontos de história de usuário também sofrer alterações.

7 <https://github.com/ADS-VILHENA/agenda-cultural-frontend>

8 https://github.com/ManoelGomes/agendacultural_mobile.git

9 <https://axios-http.com/ptbr/>

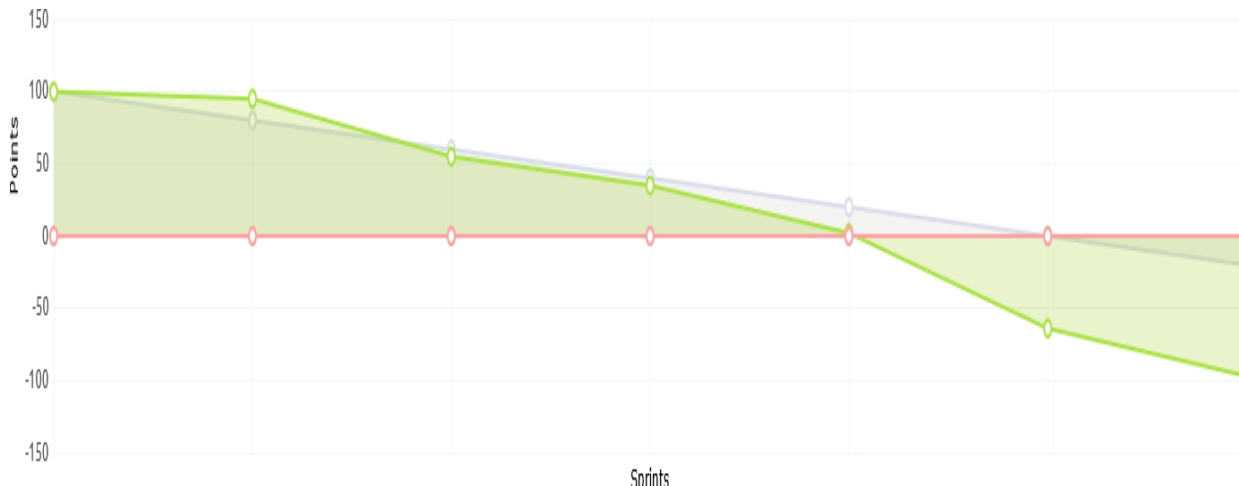
Quadro 4 – Sprints Desenvolvidas

Sprint 1 - Organizador login	
História de Usuário	Tarefa desenvolvida
Como organizador do evento eu quero poder fazer login no sistema de cadastro	Desenvolver página de login
	Desenvolver página de cadastro de organizador
Sprint 2 - Organizador tela de cadastro de evento e tela de cadastro de categoria	
História de Usuário	Tarefa desenvolvida
Como organizador, eu quero poder criar, deletar e alterar uma categoria.	Desenvolver tela de cadastro de categoria.
Como organizador, eu quero poder cadastrar meus eventos no sistema	Desenvolver tela de cadastro de evento.
Sprint 3 - Organizador deletar, alterar, evento e categoria	
História de Usuário	Tarefa desenvolvida
Como organizador, eu quero poder alterar e deletar meus eventos.	Desenvolver botões, página de deletar eventos.
Como organizador, eu quero ver os eventos criados.	Desenvolver página de eventos que foram cadastrados.
Sprint 4 - Usuário - aplicativo - tela de eventos	
História de Usuário	Tarefa desenvolvida
Como usuário, eu quero pesquisar um evento por data, para saber o dia que posso ir.	Criar projeto aplicativo
	Criar cards dos eventos para a página inicial.
	Criar componentes do evento.
	Criar página inicial
Sprint 5 - Componente categorias	
História de Usuário	Tarefa desenvolvida
Como usuário, eu quero listar todos os eventos ligados a uma categoria.	Desenvolver rota para eventos por categoria
Como usuário, eu quero ver as categorias de eventos para uma pesquisa mais rápida.	Desenvolver componente categoria
	Criar rotas para outras páginas.
Sprint 6 - Usuário - página dados eventos	
História de Usuário	Tarefa desenvolvida
Como usuário eu quero saber os detalhes de um evento, como data, horário, etc.	Desenvolver página dados evento.
	Criar rotas para acessar outras páginas

Fonte: Elaborado pela autor

A figura 11, mostra o gráfico de burndown que exibe o total de histórias de usuário vs. o tempo programado inicialmente.

Figura 11 – Gráfico de Burndown



Fonte: Elaborado pela autor

4.3 RELATÓRIO DOS TESTES

Os testes desenvolvidos para o FrontEnd tiveram o objetivo de simular o ponto de vista do usuário sobre o sistema e do organizador dos eventos. O objetivo dos testes tinham a função de encontrar erros de renderização de componentes que haviam sido criados. Após a criação dos componentes, foram aplicados os testes para verificar se os componentes criados, eram renderizados para o usuário. Para isso foi criado o identificador que foi inserido no componente a ser testado.

Ao executar o teste, com a biblioteca React Testing Library, que possui um vasto conjunto de ferramentas para testes em React, o identificador deveria ser encontrado para validar o teste. Ao retirar o identificador do componente, o teste era reprovado, mostrando desse modo que a renderização estava sendo realizada.

4.3.1 Testes realizados

Foram testados os componentes desenvolvidos, utilizando o teste de unitário, conforme o plano de testes. O teste foi aplicado após o desenvolvimento dos componentes para verificar se eram aprovados. Inicialmente os testes falharam, pelo motivo de não haver um identificador nos componentes. Após a inserção deste identificador, era possível realizar os testes com resultado aprovado.

Na Figura 12, é exemplificado, como funcionam os testes desenvolvidos para os componentes. Conforme mostra a imagem, o primeiro passo do teste foi importar o componente PageHeader, para o teste. Utilizando a biblioteca React Testing Library, foi utilizado a função "render", para simular o componente PageHeader dentro do ambiente de testes. O teste criado, busca dentro do componente o identificador, que valida a busca e confirma o teste como aprovado .

O resultado do teste com a extensão Lighthouse, buscou verificar o desempenho das páginas no navegador, analisando itens pré-estabelecidos pela própria ferramenta. O resultado do teste foi considerado aprovado, para isso a extensão foi utilizada em sites para verificar se os resultados dos testes estavam próximos do resultado do teste da aplicação.

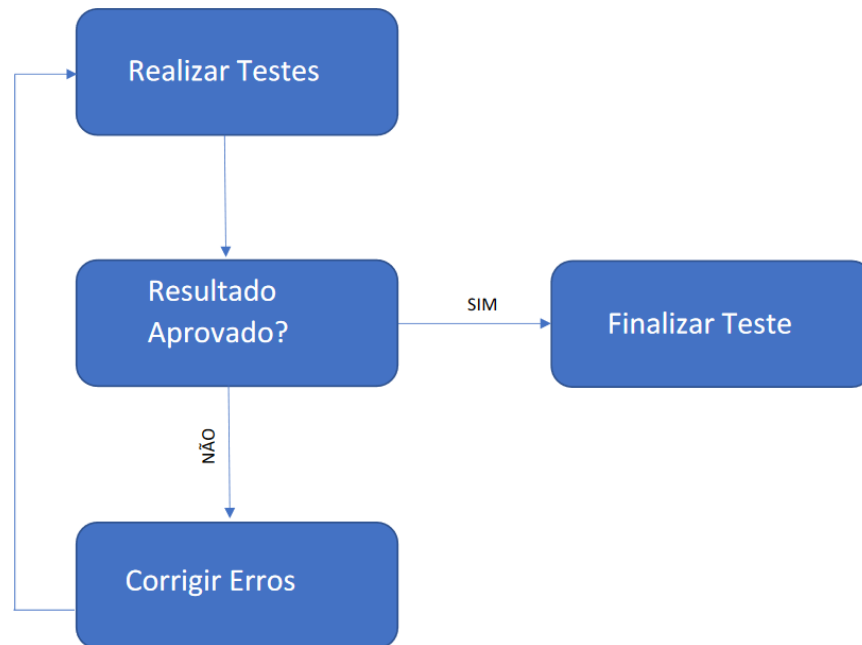
Figura 12 – Teste de Componente - exemplo 1

A screenshot of a code editor with a dark background and light-colored text. The code is a Jest test for a component named PageHeader. It uses the describe and it functions to define the test. The test renders the PageHeader component and then uses waitForElement and getByTestId to find an element with the test ID 'Page_1'. Finally, it uses expect to verify that the element exists.

```
1 describe('Teste de componente PageHeader', () => {
2   it('teste', async () =>{
3     const {getByTestId} = render(<PageHeader/>)
4     const fieldNode = await waitForElement(
5       () => getByTestId('Page_1')
6     )
7     expect(fieldNode)
8   })
9 })
```

Fonte: Elaborado pela autor

Quadro 5 – Organização dos Testes



Fonte: Elaborado pela autor

Quadro 6 – Resultado dos testes

Teste	Resultado Esperado	Resultado Obtido
Renderização de componentes com identificador	Teste aprovado	Teste Aprovado
Renderização de componentes sem identificador	Teste Reprovado	Teste Reprovado
Teste com extensão Lighthouse	Desempenho Satisfatório	Teste Aprovado

Fonte: Elaborado pela autor

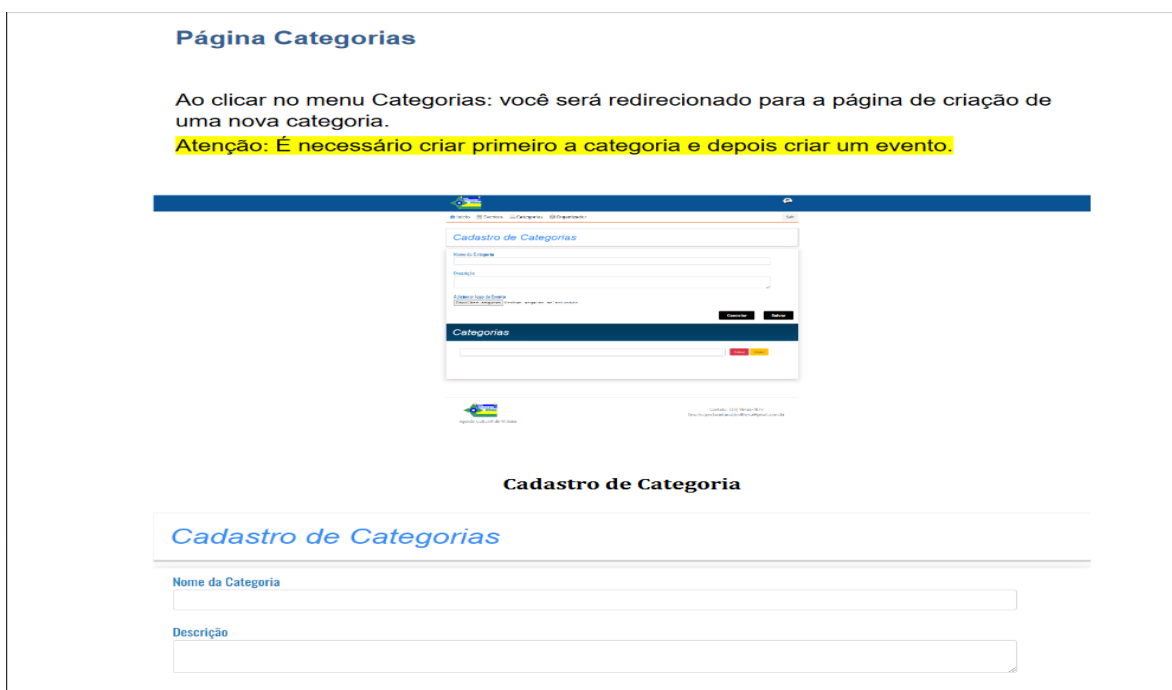
4.4 DOCUMENTAÇÃO

4.4.1 Documentação para usuários finais

A documentação para usuário final, foi feita através de um manual que está disponível no endereço: ⁶. A documentação contém as informações necessárias para que o organizador consiga realizar o cadastro no sistema, criação de categorias, eventos e alteração de dados, além de fornecer um contato para link com o desenvolvedor do sistema.

O organizador pode baixar e acessar sempre que achar necessário. Ela possui screenshots das páginas e uma explicação em texto para melhor compreensão. Além de orientar a ordem de criação de eventos e categorias, e a limitação do usuário com alguns recursos. Abaixo são apresentados exemplos da documentação para usuário final, respectivamente a descrição da página categoria de eventos onde o usuário cadastra uma categoria e uma orientação para a criação de um organizador.

Figura 13 – Documentação Usuário Final - Página Categoria



Fonte: Elaborado pela autor

⁶ <https://mega.nz/file/lBsZgLyL3r6JtBnjAVwASkzyIB5s3YtQI8Xaq90nGcsLEYnS-YA>

Figura 14 – Documentação Usuário Final - Registro de Usuário

Registro de Organizador

Esse passo a passo descreve os procedimentos necessários para realizar o cadastro de organizador no sistema Agenda Cultural de Vilhena de forma correta.

Após acessar o endereço do sistema, fornecido pelo administrador, você é direcionado para a tela de login do usuário. Clique no link "Criar Conta" logo abaixo do botão login para continuar.



Você será redirecionado para a página de registro de organizador. Preencha os campos com as informações exigidas.

- Nome;
- Telefone;
- Endereço;
- Email;

Fonte: Elaborado pela autor

4.4.2 Documentação para desenvolvedores

A documentação foi desenvolvida utilizando a extensão Markdown All in One, que está disponível para download no Visual Studio Code. Ela contém instruções sobre o projeto, como fazer o download e passos necessários para realizar a instalação. O arquivo está localizado no README.md do projeto.

Link para acessar a documentação:<https://github.com/ADS-VILHENA/agenda-cultural-frontend/readme>.

4.5 IMPLANTAÇÃO

Nesta seção será tratado a implantação do projeto para os usuários.

4.5.1 Ambiente para solução

O projeto frontend, foi disponibilizado na plataforma Expo (Mobile)¹⁰ e Netlify (Web)¹¹. Ambas plataformas permitem que projetos sejam colocados online para fins acadêmicos, com algumas limitações de recursos, ou comerciais.

4.6 DEMONSTRAÇÃO DO SOFTWARE

4.6.1 Sistema WEB

O sistema web para cadastro de eventos possui as seguintes características: cadastro de eventos, cadastro de organizador, alteração de dados, cadastro de categoria. Primeiramente, o organizador precisa registrar uma conta no sistema para poder realizar o cadastro de eventos. Abaixo uma sequência de imagens detalha as funções de cada tela do sistema. A Figura 15, detalha os componentes necessários para a criação de um evento.

O organizador preenche os dados solicitados e após clicar no botão salvar para confirmar a criação do evento. É importante que o organizador primeiro crie uma categoria para depois cadastrar o evento. Essa informação também está disponível na documentação para usuário final.

10 <https://expo.dev/@manoelgomes,eixeira/agenda-cultural-de-vilhena>

11 <https://boring-kowalevski-6b0e35.netlify.app>

Figura 15 – Demonstração de Software - Cadastro de Eventos

The screenshot shows a web interface for registering an event. At the top, there is a navigation bar with links for 'Inicio', 'Eventos', 'Categorias', and 'Organizador', and a 'Sair' button. The main heading is 'Cadastrar Evento'. The form contains the following fields:

- Nome do Evento**: A text input field.
- Descrição**: A large text area for the event description.
- Endereço**: A text input field with a placeholder example: 'Ex: Rua A, n° 000, Bairro'.
- Localização**: A text input field.
- Telefone:** A text input field.
- Hora:** A text input field with a placeholder '--:--'.
- Data:** A text input field with a placeholder 'dd/mm/aaaa'.
- Categoria:** A dropdown menu with 'teste' selected.
- Adicionar logo do Evento**: A section with a 'Procurar...' button and the text 'Nenhum arquivo selecionado.'

At the bottom right of the form, there are two buttons: 'Cancelar' and 'Salvar'.

Fonte: Elaborado pela autor

A criação de categorias é importante para a organização dos eventos para o usuário do aplicativo. Uma categoria possibilita uma pesquisa mais rápida ao usuário que tem preferência por um segmento cultural. As categorias criadas ficam localizadas abaixo para que o organizador possa, caso necessário altere ou delete uma categoria.

O organizador do evento também pode alterar seus dados, o sistema permite que ele altere seu nome, endereço, email e telefone. Para alterar a senha, o organizador deverá entrar em contato com o desenvolvedor do sistema para que seja feita a mudança. O contato está localizado no fim da página.

Figura 16 – Demonstração de Software - Cadastro de Categoria

The screenshot shows a web application interface for 'Cadastro de Categorias'. At the top, there is a navigation bar with links for 'Início', 'Eventos', 'Categorias', and 'Organizador', along with a 'Sair' button. The main content area is titled 'Cadastro de Categorias' and contains a form with the following fields: 'Nome da Categoria' (text input), 'Descrição' (text area), and 'Adicionar logo do Evento' (file upload button labeled 'Procurar...'). Below the form are 'Cancelar' and 'Salvar' buttons. A dark blue banner below the form is titled 'Categorias' and contains a search input field with 'Deletar' and 'Alterar' buttons.

Fonte: Elaborado pela autor

Figura 17 – Demonstração de Software - Alterar dados organizador

The screenshot shows a web application interface for 'Meus Dados'. At the top, there is a navigation bar with links for 'Início', 'Eventos', 'Categorias', and 'Organizador', along with a 'Sair' button. The main content area is titled 'Meus Dados' and contains a form with the following fields: 'Nome' (text input), 'Telefone' (text input), 'Endereço' (text input with a placeholder 'Ex: Rua A, n° 000, Bairro'), and 'Email' (text input). Below the form are 'Cancelar' and 'Salvar' buttons.

Fonte: Elaborado pela autor

4.6.2 Aplicativo

O aplicativo desenvolvido para a divulgação dos eventos possui as seguintes funções: visualização de eventos por dia, visualizar eventos por categoria e filtrar eventos por data. A estrutura do aplicativo está organizada para mostrar os eventos do dia em primeiro lugar, e descendo no aplicativo estão as categorias criadas e por último os eventos que já foram realizados. Abaixo algumas imagens do aplicativo desenvolvido.

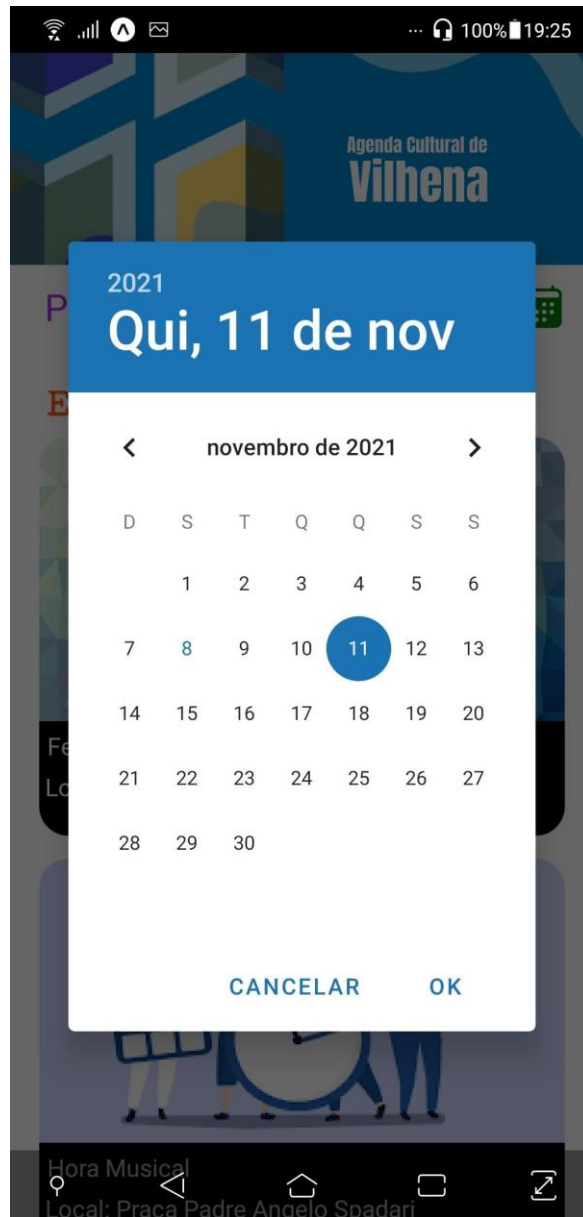
Figura 18 – Demonstração de Software - Aplicativo Início



Fonte: Elaborado pela autor

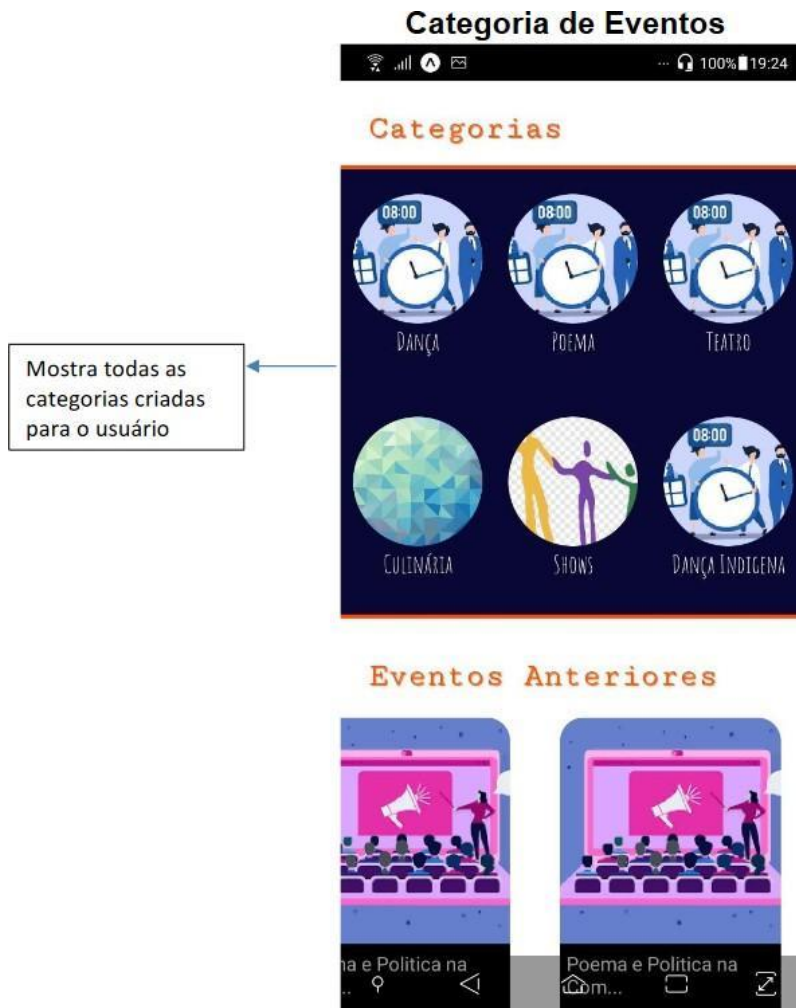
Os eventos anteriores ficam localizados no fim da página do aplicativo, eles ainda podem ser acessados e desaparecem após alguns dias automaticamente.

Figura 19 – Demonstração de Software - Aplicativo Filtrar por Data



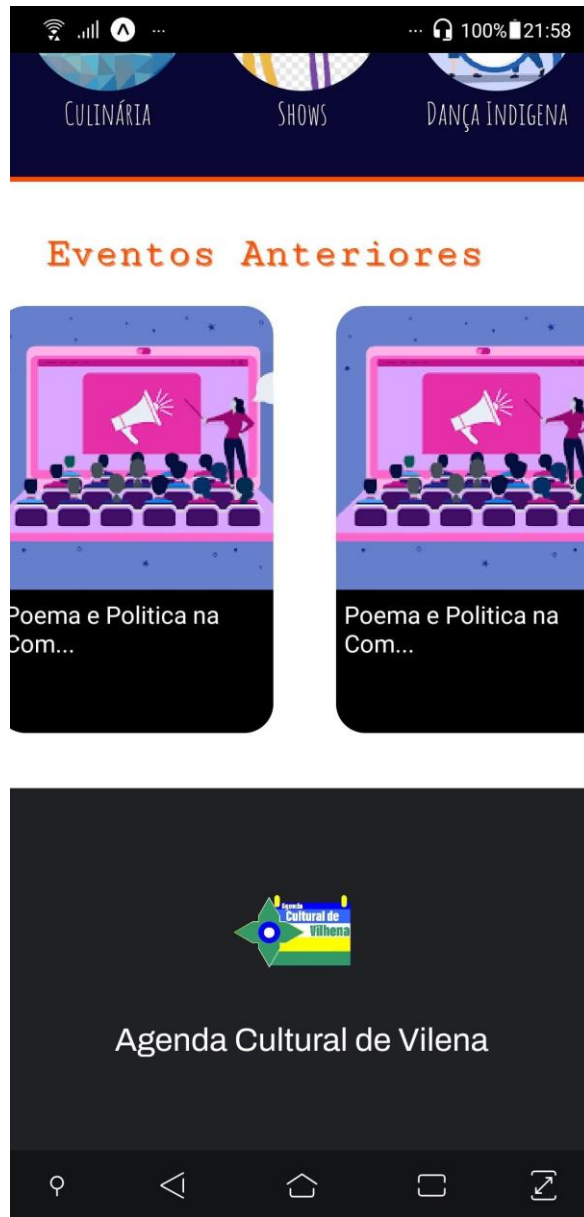
Fonte: Elaborado pela autor

Figura 20 – Demonstração de Software - Aplicativo Categoria de Eventos



Fonte: Elaborado pela autor

Figura 21 – Demonstração de Software - Aplicativo Eventos Anteriores



Fonte: Elaborado pela autor

5 CONSIDERAÇÕES FINAIS

Eventos culturais são importantes para a manutenção de uma identidade própria de um local. Os grandes eventos são amplamente divulgados com o objetivo principal de obter lucro, deixando de lado, muitas vezes pequenos eventos localizados em lugares menos populosos. Proporcionar uma forma de divulgação desses pequenos eventos foi a base para a criação do projeto deste trabalho.

O desenvolvimento de um aplicativo voltado para a divulgação de eventos se mostrou muito receptivo, principalmente após realizada a entrevista com representantes do segmento, que identificou uma ausência de um sistema específico para essa finalidade. Apesar de eventos culturais serem divulgados utilizando muitas plataformas, por exemplo, instagram, facebook, rádio, etc. Sua divulgação ainda pode ser considerada pequena para a população.

O objetivo do trabalho foi desenvolver o FrontEnd do aplicativo Agenda Cultural de Vilhena e do painel administrador para a criação dos eventos. O desenvolvimento do projeto foi finalizado conforme planejado. O projeto se limitou ao desenvolvimento da telas, a integração com o BackEnd foi feita de forma parcial, restando algumas funções ainda por serem integradas com a API, que foi desenvolvida separadamente.

O aplicativo vai contribuir para uma maior engajamento por parte da população, com eventos culturais da cidade. Para sua construção foram traçados objetivos que permitiram uma melhor compreensão do que era almejado com o projeto e como construí-lo. Planejar a parte visual do aplicativo se mostrou desafiadora inicialmente, devido a falta de conhecimento da área cultural. Os objetivos específicos foram extremamente importantes para o desenvolvimento do projeto proposto e para a sua conclusão de forma mais eficiente.

Apesar do FrontEnd estar finalizado, ainda há muito o que ser feito, principalmente na questão da integração com a API, para que de fato ele possa ser utilizado pelas pessoas. Isso permite que os seguintes trabalhos futuros possam ser realizados.

- Integração com API (Organizador e Aplicativo) - Iniciada mas não concluída.
- Implantação de um relatório para que o organizador verifique os eventos mais acessados por categoria.
- Ampliação do aplicativo para suportar eventos de mais cidades.

REFERÊNCIAS

BRANT, L. O Poder da Cultura. São Paulo: Ed. Peirópolis, 2009.

Matias, M. (2011). Planejamento, organização e sustentabilidade em eventos. São Paulo: Manole.

BRITTO, Janaína e FONTES, Nena. Estratégias para eventos: uma ótica do marketing e do turismo. São Paulo, SP: Aleph, 2002.

CITIES, United; GOVERNMENTS, Local. Agenda 21 for culture. **Barcelona: Committee on culture–United Cities and Local Governments**, 2004. GONÇALVES, Fernando. Tecnologia e cultura: usos artísticos da tecnologia como prática de comunicação e laboratório de experimentação social. In: Revista Famecos, Porto Alegre, n. 38, v. 1, p. 100-110, 2009.

PAGLIOTO, Bárbara Freitas; MACHADO, Ana Flávia. Perfil dos frequentadores de atividades culturais: o caso nas metrópoles brasileiras. **Estud. Econ.**, São Paulo, v. 42, n. 4, p. 701-730, Dec. 2012 Available from <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-41612012000400003&lng=en&nrm=iso>.access on 24 June 2020. <https://doi.org/10.1590/S0101-41612012000400003>.

GADELHA, Rachel; BARBALHO, Alexandre. Políticas públicas de cultura e o campo da produção cultural. *Pensamento & Realidade*, v. 28, n. 4, p. 70-84, 2013.

PAES, Tais Alexandre Antunes; LADEIRA, Rodrigo. Aplicativos turísticos e a Sua Relação com a Atividade Turística, Varejo Virtual e o Comportamento de Consumo. In: CLAV 2018. 2018.

RIVERO, Luis; DO VALLE, Raquel; CONTE, Tayana. Aplicando design e avaliação de usabilidade para melhorar a qualidade de um aplicativo web móvel. Anais do Simpósio Brasileiro de Qualidade de Software (SBQS 2014), p. 260-267, 2014.

ANEXO A – Licença MIT

MIT License

Copyright (c) 2020 ADS Vilhena

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.