

TICKETC: SISTEMA DE GERENCIAMENTO DE EVENTOS

Dionatan Inhoato Markiu¹, Giovane Galvão¹

¹Centro Universitário Campo Real

Rua Comendador Norberto, 1299 - Santa Cruz – Guarapuava – PR – Brasil

{engs-dionatanmarkiu@camporeal.edu.br, prof_giovanegalvao@camporeal.edu.br}

Resumo

Eventos desempenham um papel importante no contexto social e econômico da sociedade, sendo um dos setores que mais cresce no Brasil. Sua devida organização demanda do gerenciamento de diversas regras e informações características e específicas para cada ação planejada. Por meio de uma análise de sistemas similares, detectou-se que a maioria deles não possui funcionalidades capazes de alcançar todos os requisitos para tal, como o gerenciamento de inscrições e de subeventos. Por isso, o presente artigo descreve o processo de pesquisa e construção de um sistema de gerenciamento de eventos que busca romper essa limitação. O sistema é indicado para gerentes de eventos que procuram uma experiência mais ampla, com diversas opções de personalização para os seus projetos. Além disso, os usuários contarão com funcionalidades aprimoradas de inscrição e participação em eventos. A metodologia utilizada para a realização deste projeto foi o framework Scrum. O codificação da plataforma foi realizado com o framework PHP Laravel, voltado para desenvolvimento web. O armazenamento de dados foi efetuado utilizando o SGDB MySQL. A plataforma desenvolvida pelo trabalho apresenta funcionalidades abrangentes e eficientes, além de auxiliar os processos de controle e gerenciamento de eventos, também oferece aos usuários uma experiência mais eficiente e intuitiva.

Palavras-chave: Plataforma. Gestão de Eventos. Subeventos. Laravel. Personalização.

Abstract

Events play an important role in the social and economic context of society, being one of the fastest-growing sectors in Brazil today. Their proper organization requires managing various specific rules and information unique to each planned action. However, research conducted to find systems tailored to this purpose found them lacking in functionalities capable of meeting all necessary requirements, such as registration management and sub-event handling. Therefore, this article describes the research and development process of an event management system aimed at overcoming this limitation. The system is designed for event managers seeking a broader experience with various customization options for their projects. Additionally, users will benefit from enhanced registration and participation features. The methodology used for this project was the Scrum framework. The platform's coding was carried out using the PHP framework Laravel, which is geared toward web

development. Data storage was handled using the MySQL DBMS. The platform developed in this work offers comprehensive and efficient functionalities; in addition to supporting event control and management processes, it also provides users with a more efficient and intuitive experience.

Keywords: Platform. Event Management. Sub-events. Laravel. Customization.

1. Introdução

Os eventos desempenham um papel importante na estrutura social e econômica da sociedade como um todo, sendo sua definição mais simples um acontecimento que visa um fim específico a partir da reunião de um grupo de pessoas. Contudo, um evento também pode ser entendido como uma ferramenta que permite uma resposta precisa e imediata junto ao público, uma vez que deve despertar emoções, gerar sentimentos, incentivar debates sobre determinados temas e estabelecer, assim, uma conexão com o público (MENDONÇA, PEROZIN, 2014, p.26).

O setor de eventos de entretenimento e cultura apresenta a maior taxa de crescimento entre todos os setores da economia no ano de 2023, segundo a ABRAPE (Associação Brasileira dos Promotores de Eventos), abrangendo diversas áreas, como espetáculos, *shows*, atividades de recreação, eventos esportivos e eventos ligados ao patrimônio cultural e ambiental.

Todavia, o gerenciamento de eventos pode ser um desafio para aqueles que assumem o papel de gestor, já que o desenvolvimento de um evento não segue regras sistemáticas universais, cada evento pode conter um formato próprio, com uma série de fatores que variam em todo o processo de sua construção (IVO, MARIN, SOUZA, 2014). Sendo assim, formas de gerenciamento abrangentes se tornam essenciais para otimizar o controle e a organização de informações e tarefas pertinentes a tais eventos.

Atualmente, diversas plataformas de gerenciamento de eventos podem ser encontradas no mercado, como a Sympla¹ e a Even3². No entanto, seus planos gratuitos oferecem funcionalidades limitadas, o que pode ser insuficiente para gestores de eventos que não optam pelos planos pagos mensais ou anuais. Além disso, certas particularidades e necessidades específicas de alguns tipos de

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

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

eventos e instituições podem não ser totalmente atendidas, mesmo por essas plataformas (AIDA, FERREIRA, FONTÃO, 2023, p.10).

Um evento é um fator marcante, no qual erros acabam por passar um sentimento de insatisfação e desconforto para seus envolvidos, tanto organizadores, quanto o público, possivelmente acarretando em problemas maiores, seja com patrocínios ou eventos posteriores (MENDONÇA, PEROZIN, 2014). Além disso, fatores como a eficiência e a facilidade de uso de funcionalidades e recursos de uma plataforma se mostram essenciais para *softwares* que exigem interação e gestão humana. Portanto, uma interface intuitiva e funcionalidades que se adequam às demandas dos usuários se fazem necessárias, desempenhando um papel crucial no sucesso de sistemas (PRESSMAN, MAXIM, 2021, p.233).

Levando em conta tais pontos, e assumindo que o mundo atual demanda produtos e ferramentas tecnológicas que buscam melhorar a eficiência de trabalhos e serviços (PRESSMAN, MAXIM, 2021), a criação de uma plataforma gerenciamento de eventos com diversas opções de funcionalidades de gestão e personalização de eventos se mostra uma excelente alternativa para auxiliar gestores com a coordenação de atividades, inscrições e a comunicação com participantes, etapas essenciais para o desenvolvimento de eventos.

Com a criação desta plataforma, espera-se oferecer uma solução abrangente e eficiente de gestão de eventos, buscando proporcionar uma experiência mais fluida e eficiente para todos os envolvidos no processo de organização e participação, simplificando o processo de inscrição e coordenação de eventos de diversos tipos e tamanhos.

Este artigo está dividido em seções. A Seção 2, Fundamentação Teórica, busca explorar a teoria envolvida nas ferramentas de planejamento, elaboração e criação da plataforma apresentada neste trabalho. A Seção 3, Estado da Arte, busca apresentar o estado atual das pesquisas relacionadas ao desenvolvimento de sistemas para gerenciamento de eventos. A Seção 4, Metodologia, descreve os métodos e procedimentos utilizados na realização da pesquisa atual. A Seção 5, Resultados, expõe a utilização do estudo e ferramentas previamente estudados para a construção da plataforma, além disso, são apresentadas as conclusões alcançadas a partir da realização do trabalho. Por fim, na Seção 6, as considerações finais do trabalho são evidenciadas.

2. Fundamentação Teórica

Nesta seção serão apresentados a base conceitual e embasamento teórico necessários para a compreensão do tema e das ferramentas estudadas e utilizadas, contextualizando os tópicos relevantes e orientando o desenvolvimento do trabalho.

2.1. Eventos

Os eventos são acontecimentos amplamente empregados devido a sua capacidade de alcançar propósitos dos mais variados, são instrumentos potentes no âmbito social com a finalidade de abranger um público, específico ou geral, levá-lo a consumir ou entreter-se no decorrer de sua execução, e alcançar um objetivo por meio disto (NOGUEIRA, et al, 2020, p. 27).

Os eventos também têm a função de atuar como um mediador entre organizações e seu público. Através dos eventos é possível receber um retorno rápido sobre os pensamentos e discussões presentes no público alvo (MENDONÇA, PEROZIN, 2014, p. 26).

Para Matias (2013, p. 115), o conceito de evento é algo dinâmico, por ser uma atividade que sofre variações e modificações com o tempo. As concepções possíveis para evento diversificam-se desde: A ação de planejamento e organização de um projeto que busca atingir um público específico; Uma comemoração, seja por fim econômico ou não, visando atingir seu público; Soma das ações planejadas com objetivo específico a frente um público-alvo.

Os objetivos e elementos específicos diferenciam os eventos entre si, estes são o que lhes atribuem particularidades e definem em qual categoria um evento melhor se encaixa (NOGUEIRA, et al, 2020, p. 27).

2.1.1. Características

Para que a realização de algum evento possa ser possível, uma série de critérios de planejamento precisam ser considerados, desde a fase de planejamento, para Mendonça e Perozin (2014, p. 28), o organizador de qualquer evento necessita saber características como: o tamanho do evento, ou seja, a quantidade de público que ele vai abranger; além disso, a data de realização, sendo ela fixa ou não; e por fim, o perfil do público, se o evento espera um público específico ou um público geral.

Já para Matias (2013, p. 117), a classificação de qualquer evento pode ser realizada a partir de três características iniciais: o tipo de público atendido, podendo um evento ser público ou privado; ao número de participantes esperados ou capacidade máxima; além disso, a área de interesse que o evento busca seguir. Apesar de alguns autores divergirem na escolha dos critérios iniciais do planejamento de eventos, muitos deles acabam por concordarem em diversas características adiante. Levando em consideração o número de participantes, a classificação dos eventos pode ser considerada da seguinte forma:

- Pequeno: até 150 pessoas;
- Médio: entre 150 e aproximadamente 500 pessoas;
- Grande: acima de 500 pessoas;
- Megaevento: Mais de 5000 pessoas.

Dentre as demais características, a área de interesse acaba por se evidenciar. Para Mendonça e Perozin (2014, p. 32) e Matias (2013, p. 118), os eventos se diferenciam nas seguintes áreas de interesse:

- Artística, como *shows*, teatro e literatura; Científica, como apresentações de pesquisas;
- Cultural, como eventos regionais ou folclóricos;
- Educativa, que trazem novidades didáticas;
- Cívica, como desfiles que enfatizam o patriotismo;
- Governamental, como eventos municipais, estaduais ou federais;
- Empresariais, com foco em expor organizações;
- Lazer, como eventos de entretenimento, gincanas, jogos;
- Social, eventos de confraternização de um grupo específico;
- Desportiva, com foco nos esportes;
- Religiosa, como procissões, jornadas e acampamentos;
- Beneficente, com foco em ações sociais;
- Turísticos, eventos que divulgam um ponto turístico, principalmente;

2.1.2. Tipologia

As áreas de interesse citadas anteriormente não representam um limite, eventos mais específicos podem conter outra área, além disso, um evento pode conter mais de uma delas ao mesmo tempo, dependendo do objetivo que se espera alcançar (NOGUEIRA, et al, 2020, p. 29).

A definição das características se faz necessária para a designação do tipo do evento. A falta dessa etapa no planejamento pode acarretar em várias dificuldades durante o processo de organização. O organizador deve, então, utilizar seu conhecimento prévio ou pedir auxílio de terceiros para a definição da tipologia do evento (MENDONÇA, PEROZIN, 2014, p. 30). Dentre os tipos de eventos que existem, alguns deles podem ser visualizados na Figura 1.

Figura 1: Tipos de eventos

Almoço	Congresso	Fórum	Oficina
Assembleia	Confraternização	<i>Garden party</i>	<i>Open day</i>
Aula inaugural	Convenção	Gincana	Painel
Aula magna	Coquetel	<i>Happy hour</i>	Palestra
<i>Brainstorming</i>	Debate	Inauguração	Rodada de negócios
<i>Bazar</i>	Desfile	Jantar	Salão
<i>Brunch</i>	Painel	Jornada	Semana
Campeonato	Encontro	Leilão	Seminário
Concílio	Exposição	Mesa redonda	<i>Showcasing</i>
Concurso	Feira	Mostra	Simpósio
Conferência	Festa	Olimpíadas	Videoconferência ou teleconferência

Fonte: Nogueira, et al (2020, p. 27).

A análise das características e definição do tipo de evento mostram-se fundamentais para o seu gerenciamento subsequente. A partir do resultado dessas tarefas pode-se iniciar o planejamento, buscando alcançar o devido objetivo principal e público alvo, além disso, a organização dos acontecimentos e posterior execução também dependem totalmente dessas ações iniciais.

2.2. E-business e Plataformas Digitais

Segundo Turban e Volonino (2013), no mundo atual, o desenvolvimento de novas tecnologias se faz importante em diversas disciplinas, seja em negócios, vendas e gerenciamento, processos e operações bem organizados e facilitados por tais tecnologias são essenciais para um aprimoramento nas relações de consumidores e gerentes.

Essas tecnologias evoluíram com o passar dos anos, se mostrando necessárias para a atualização e apoio aos processos de negócio (TURBAN, VOLONINO, 2013, p. 17). Segundo Baltzan e Phillips (2012, p. 95), o gerenciamento de operações é a administração de sistemas ou processos responsável pela coordenação dos processos fundamentais utilizados para a produção de bens e serviços.

Para Gonçalves (2017 p. 125) o *e-business* se destaca neste contexto, tratando dessas operações de forma eletrônica, em sistema e ou plataformas sistematizadas, auxiliando na melhora da organização e comunicação com demais públicos, segundo o autor: “A TI impulsiona as organizações por intermédio de suas ferramentas e o *e-business* auxilia na interface dessas transações com os demais componentes do sistema” (GONÇALVES, 2017 p. 127).

Segundo Baltzan e Phillips (2012, p. 94), conforme a evolução e aperfeiçoamento desses processos de negócio, o *e-business* pode demonstrar ser benéfico e ao mesmo tempo, um estímulo para o gerenciamento em geral. A utilização de plataformas ligadas à internet já está sendo o foco de empresas, todas possuindo o intuito de contribuir com a melhora nos seus processos de negócio, os autores Baltzan e Phillips (2012, p. 95) afirmam: “Um número crescente de empresas já está utilizando a internet para dinamizar seus processos de negócios, adquirir materiais, vender produtos, automatizar o atendimento ao cliente e criar fluxos de receita”.

Nesse sentido, a implementação de uma plataforma de gerenciamento *web* acaba sendo uma opção extremamente válida, no contexto de apoiar os processos de negócio e melhorar o relacionamento entre cliente e empresas, mantendo o gerenciamento atualizado e ativo no contexto atual.

2.3. Planejamento e Etapas de Desenvolvimento

Atualmente, o *software* se mostra ter mais de uma função, além de ser um produto, ele pode ser a ferramenta de distribuição de outros produtos ou serviços. Conforme aumenta sua importância e especificação, os programas seguem diferentes caminhos, entre eles, os aplicativos e plataformas *online* (PRESSMAN, MAXIM, 2021, p. 3). Para os autores, tais *softwares* de aplicação solucionam demandas e objetivos específicos para o gerenciamento de negócios, facilitando operações e processos.

Nesse contexto, a criação desses sistemas necessita de um planejamento adequado, a fim de alcançar todas as demandas e progredir em ritmo equilibrado. Segundo Maschietto, Rodrigues e Bianco (2020, p. 52), os princípios para o desenvolvimento de software devem seguir critérios de escolha adequados para a atualidade, ou seja, os métodos escolhidos devem ser mais dinâmicos.

Uma metodologia de processo qualquer, voltada para o desenvolvimento de sistemas deve seguir atividades metodológicas pré-estabelecidas, além de necessitar da aplicação de atividades de apoio durante todo o processo. Tais atividades são alocadas dentro uma configuração de atividades, ações e tarefas, organizadas de forma a seguir uma sequência e um tempo, conhecido como fluxo de processo, é a partir dele que todas as atividades e o processo de *software* deve ser centrado (PRESSMAN, MAXIM, 2021, p. 21).

Levando em consideração tais ideias, um modelo de processo focado em desenvolvimento e entrega de valor, além de ser dinâmico e adaptável aos requisitos e nuances do projeto, revela-se um bom caminho a seguir. Assim sendo, um modelo de processo ágil atende os critérios necessários, segundo Maschietto, Rodrigues e Bianco (2020, p. 52), tais modelos priorizam pontos diferentes do que outras metodologias, entre eles: A entrega contínua de valor, ou seja, artefatos do sistema; A adaptatividade a mudanças e o conserto de problemas; Funcionamento do sistema, acima da documentação extensiva, voltado para o atendimento das funcionalidades necessárias.

2.4. Scrum e Kanban

Dentre os modelos de processo existentes, o *Scrum* é um dos mais utilizados atualmente, além de seguir os ideais ágeis, ele organiza o trabalho de uma forma adaptativa e foca em melhorar o processo continuamente (PRESSMAN, MAXIM, 2021, p. 41).

Segundo Pressman e Maxim (2021, p. 9), uma metodologia de processos universal abrange cinco atividades principais:

- Comunicação: Onde os objetivos do projeto e os requisitos do sistema são levantados, aquilo que agrega valor ao sistema;
- Planejamento: Onde as tarefas, recursos e cronograma são realizados;
- Modelagem: Onde cria-se um “esboço” do sistema por meio de modelos e diagramas para facilitar a compreensão de suas características;

- Construção: Onde o sistema é construído por meio de códigos e testes;
- Entrega: Onde o sistema, ou uma parte dele, é entregue e recebe uma avaliação.

No *Scrum*, a primeira atividade a ser realizada é a criação de um *Product Backlog*, uma lista de requisitos e outras funcionalidades necessárias para o desenvolvimento do sistema. Seguindo tal lista, o trabalho pode ser subdividido em períodos curtos e iterativos com duração fixa, normalmente variando entre uma a quatro semanas. Chamadas de *Sprints*, essas iterações focam em entregar um incremento do produto, seja uma parte potencialmente pronta ou um artefato que auxilia no desenvolvimento futuro da aplicação (RUBIN, 2017, p. 62).

Usando como base o *Product Backlog*, a lista de tarefas a ser realizada durante a *Sprint* pode ser definida na reunião conhecida como *Sprint Planning*, ou planejamento da *Sprint*, tal lista recebe o nome de *Sprint Backlog*. Na fase de análise da *sprint*, os requisitos são organizados conforme aspectos pré estabelecidos, determinando a prioridade, a ordem de realização, e qual grupo ou pessoa ficará encarregado da sua realização (RUBIN, 2017, 335). Durante a fase de Projeto e Evolução, onde as funções do sistema são desenvolvidas e incrementadas, é necessário trabalhar inicialmente com os itens de maior prioridade, as principais funcionalidades ou processos do sistema. Para facilitar a visualização de tais prioridades, outra metodologia ágil pode ser implementada em conjunto com o *Scrum*, o *Kanban* (PRESSMAN, MAXIM, 2021, p. 49).

O *Kanban* é uma metodologia desenvolvida para descrever melhorias de métodos para o aperfeiçoamento de processos ou fluxos de trabalho. Pressman e Maxim (2021, p. 48) descrevem o foco do *Kanban* no gerenciamento de alterações e mudanças na entrega de serviços. Tal gestão é realizada conforme o quadro *Kanban*, nele, o trabalho é dividido em colunas, onde cada uma representa um estágio de desenvolvimento, “À fazer”, “Fazendo” e “Concluído”, por exemplo. Os elementos e funcionalidades do sistema são então organizados no quadro em “cartões” e podem mudar de coluna conforme o avanço do projeto. Além disso, cada um desses cartões pode receber informações importantes, como prioridade e responsável por realizar tal atividade.

O quadro *Kanban* pode ser representado fisicamente ou virtualmente, com a utilização de alguma ferramenta focada em produtividade. Entre as ferramentas utilizadas para a organização de atividades, a ferramenta Trello se destaca,

segundo Maschietto, Rodrigues e Bianco (2020, p. 52), o Trello é uma ferramenta que auxilia na organização de tarefas apoiando a gestão de métodos ágeis, é uma das principais e mais utilizadas ferramentas nas organizações e empresas no mercado atual.

Com o auxílio dos artefatos e ferramentas disponíveis durante a execução da *sprint* é possível alcançar a última fase do processo, a entrega. Nela, uma parte do sistema potencialmente entregável é realizada, isso significa que os objetivos foram alcançados e geraram uma evolução no sistema, mas a entrega do incremento pode ou não ser realizada, dependendo da necessidade do projeto.

2.5. Banco de Dados

A fim de armazenar e organizar os dados necessários para uma aplicação ou plataforma, é imprescindível a criação de um banco de dados bem estruturado e robusto. Os banco de dados relacionais são a principal escolha para sistemas que necessitam armazenar grandes quantidades de dados que se relacionam entre si (HEUSER, 2011, p. 121).

Segundo Duckett (2024, p. 124), os banco de dados relacionais armazenam os dados necessários em tabelas, podendo armazenar diversas delas, cada uma contendo grande quantidade de dados e informações alocadas. Heuser (2011, p. 24) diz que, para a criação de um banco de dados bem estruturado, a modelagem deve ser realizada inicialmente, tal modelagem se refere a descrição textual ou visual dos tipos de informações armazenadas, usando uma linguagem de modelagem de dados. Na modelagem padrão de projetos de *software*, dois níveis de abstração são os mais utilizados, o conceitual e o lógico.

Segundo Heuser (2011, p. 27), o modelo conceitual é o mais abstrato, nele são descritos a estrutura do banco de dados independentemente de um Sistema Gerenciador de Banco de Dados (SGBD), ou seja, registra quais dados podem ser inseridos no banco, porém não registra a forma que eles serão armazenados. Já o modelo lógico representa a estrutura que os dados terão no banco, conforme a sua visualização com um SGBD. O autor complementa, um SGBD é um *software* que possui as principais funções para organizar um banco de dados e facilitar sua manipulação, é através dele que se torna possível inserir, recuperar e alterar dados e elementos do banco.

Para Duckett (2024, p. 127), o MySQL é um SGBD relacional capaz de suportar múltiplos bancos de dados, assim como grande quantidade de dados. Apesar de não possuir uma interface de visualização própria, outras aplicações de fonte aberta auxiliam na realização de tarefas de administração dessa ferramenta, dentre elas está o phpMyAdmin, um *software* desenvolvido na linguagem PHP, com esse objetivo específico, que funciona a partir do servidor *web* Apache.

2.6. PHP e Laravel

O PHP é uma linguagem de programação extremamente confiável que possibilita a criação de diversos tipos de aplicações e sistemas, é uma linguagem *back-end* muito utilizada em páginas *web*, pois através dele qualquer tipo de tarefa pode ser realizada durante o desenvolvimento. Muito utilizado por meio do servidor XAMPP³, desenvolvido pela Apache, possui facilidade de instalação e um pacote de aplicativos que auxiliam no desenvolvimento *web*, entre eles o servidor Apache e o banco de dados MySQL (SARAIVA, BARRETO, 2018, p. 57).

Apesar do desenvolvimento em PHP sem a utilização de nenhuma ferramenta externa ser possível, a utilização de *frameworks* pode auxiliar de forma exponencial a construção e organização de um sistema, dentre os vários *frameworks* de PHP que existem, Laravel se destaca.

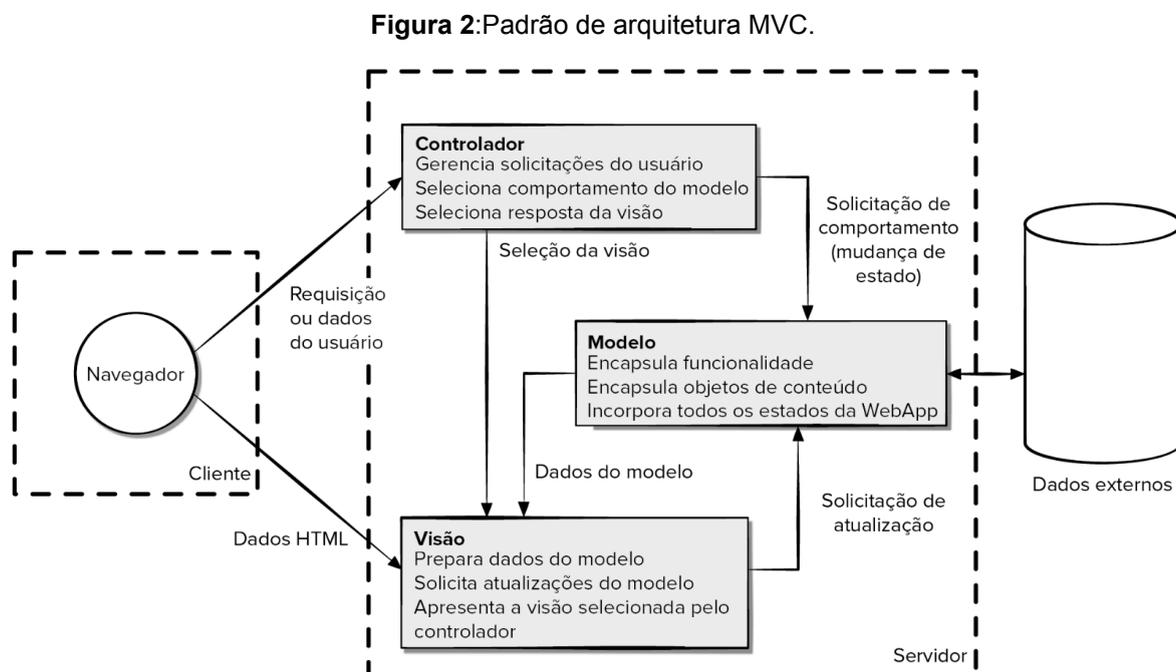
Segundo Kausar e Swati (2019), Laravel é um *framework web* projetado para criar aplicações e plataformas *web* com alta qualidade, utilizando uma sintaxe limpa e expressiva. Ele oferece uma grande variedade de pacotes de ferramentas e módulos prontos que podem facilitar o trabalho de desenvolvimento, economizando tempo e simplificando o processo, reduzindo a necessidade de planejar e desenvolver sites do zero. Além disso, cuida da segurança do sistema e possui uma estrutura robusta para aplicações.

O Laravel utiliza o padrão de arquitetura *Model-View-Controller*, ou MVC, responsável por dividir as responsabilidades no sistema e organizar o código de forma a melhorar o fluxo de controle. Para Zenker, Santos e Couto (2019, p. 96), tal fluxo inicia num evento de interação do usuário e a partir dele:

³ Disponível em: https://www.apachefriends.org/pt_br/index.html

- O *Controller*, ou controlador, manipula essa interação e converte em uma ação para que o *Model* seja capaz de compreender, é o elemento responsável por estruturar e controlar as ações;
- Depois, o *Model*, ou modelo, administra os procedimentos e comportamentos do sistema, além disso, é o responsável por gerenciar e manipular os dados da aplicação;
- Por fim, a *View*, ou visão, é a responsável por adicionar as informações e elementos na exibição para o usuário;

A Figura 2 representa visualmente as conexões existentes entre os elementos e suas características:



Fonte: Pressman e Maxim (2021, p. 189).

Assim, com a utilização desse padrão, a entrada de dados é manipulada pelos *controllers*, as funcionalidades principais e os dados são gerenciados pelo *model* e as informações são mostradas aos usuários através das *views*, separando as regras de negócio de tais interfaces ligadas diretamente ao cliente.

Para a manipulação de dados através do *framework* Laravel, Subecz (2021) destaca as *migrations*, ou migrações, como um dos recursos avançados que podem ser utilizados. As *migrations* são códigos usados para manipular a estrutura e o conteúdo do banco de dados diretamente nos arquivos do projeto, permitindo que o

banco de dados se mantenha em um estado consistente. Além disso, o autor também cita os *seeders*, ou semeadores, que são códigos responsáveis por popular a base de dados com informações e dados randômicos, gerados com base no tipo de dado de cada coluna das tabelas criadas. Tais recursos diminuem a necessidade de trabalho manual e facilitam a realização de testes durante o desenvolvimento do projeto.

3. Estado da Arte

Para selecionar os trabalhos relacionados a esta pesquisa, efetuou-se uma busca de artigos disponíveis no *site* Google Acadêmico⁴, no qual foram fornecidas palavras-chave relacionadas ao tema, como por exemplo, a palavra eventos

Após as pesquisas realizadas em busca de sistemas similares ao proposto por este trabalho, foram encontrados dois trabalhos semelhantes em objetivo, porém com certas diferenças.

A ferramenta Tô Nessa foi a primeira a ser analisada, ela foca na criação e gerenciamento de eventos e informações de usuários, a criação de eventos foca em eventos únicos e com uma quantidade de informações insuficientes para uma capacidade de gerenciar eventos de forma abrangente, além disso, não há a funcionalidade de gerenciar ingressos e quantidade de participantes por evento (ESTEVES, et al, 2017). A figura 3 mostra a página de criação de eventos da plataforma Tô Nessa:

⁴ Disponível em: <https://scholar.google.com.br/?hl=pt>

Figura 3: Página de criação de evento da plataforma Tô Nessa.

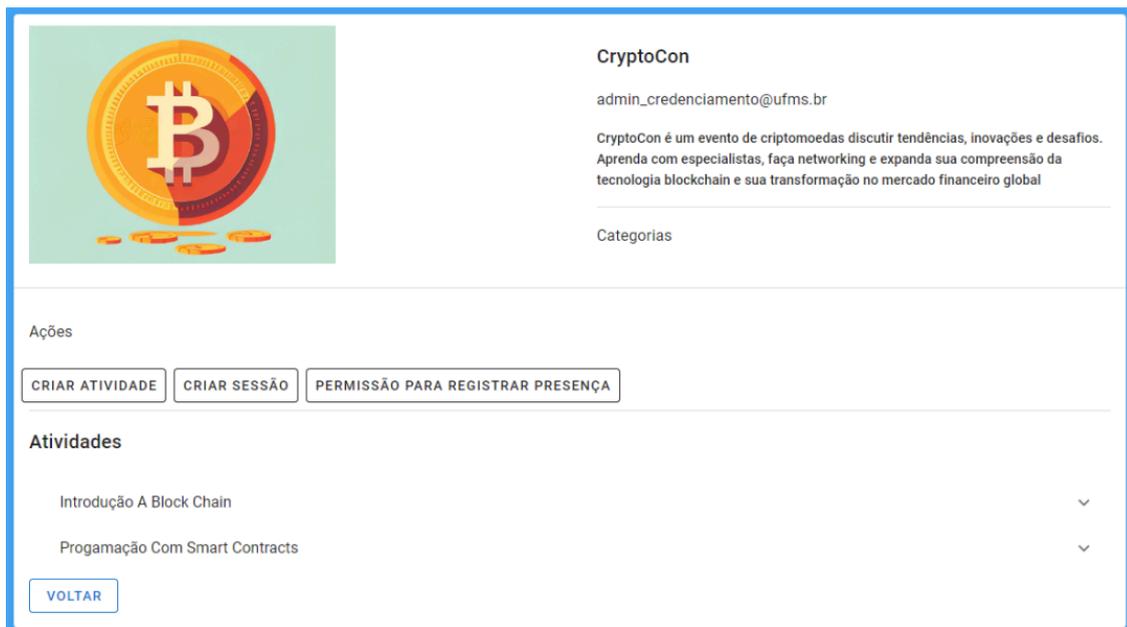
A imagem mostra a interface de usuário para a criação de um evento na plataforma Tô Nessa. O formulário é organizado em seções: 'Nome do Evento' com um campo de texto e um menu suspenso para 'Tipo'; 'Banner do Evento' com um botão 'Escolher arquivos' e o texto 'Nenhum arquivo selecionado'; 'Início' e 'Término' com campos de data e hora; 'Local' com campos para 'Estado', 'Bairro', 'Rua/Av', 'Nº' e 'CEP'; e 'Descrição' com uma área de texto grande. Um botão 'Criar' está localizado na parte inferior do formulário.

Fonte: Esteves, et al (2017, p. 8).

Porém, o sistema conta com funcionalidades semelhantes às de uma rede social, no qual é possível se comunicar entre usuários, marcar presença em eventos e convidar amigos para participar de eventos que você também vai, além disso, usuários podem ganhar cupons por participar de certos eventos (ESTEVES, et al, 2017).

Já a plataforma SISEVENT, a segunda analisada, conta com funcionalidades mais abrangentes, além de também ser realizado em Laravel, nela é possível criar e gerenciar eventos acadêmicos com uma gama de funcionalidades maior, dentre elas a administração do time de ajudantes na organização, um controle mais rígido dos participantes do eventos, a geração de relatórios e certificados de presença, além da separação de atividades de organização de eventos em sessões mais flexíveis para a execução dos projetos (AIDA, FERREIRA, FONTÃO, 2023). Porém, o foco do sistema é na organização de eventos acadêmicos de instituições de ensino superior, eventos fora dessa área de interesse já não poderiam ser atendidos por essa plataforma, um show musical, por exemplo, possui características distintas de uma feira acadêmica, portanto o seu gerenciamento seria prejudicado pela falta de funcionalidades adequadas às necessidades específicas de sua organização. A figura 4 mostra a página de exibição de evento da plataforma SISEVENT.

Figura 4: Página de exibição de evento da plataforma SISEVENT.



Fonte: Aida, Ferreira, Fontão (2023, p. 36).

Levando em consideração as funcionalidades dos sistemas citados anteriormente, embora consigam abranger um certo nível de gerenciamento de eventos, nenhuma delas evidenciou funcionalidades equivalentes com as quais a plataforma TickETC propõe: personalização de eventos com base em diversas informações, inscrições em eventos com ingressos, organização de subeventos dentro de eventos e um painel de gestão para administradores de eventos.

4. Metodologia

Para entregas mais ágeis e iterativas foi escolhida a metodologia *Scrum* como meio de desenvolvimento do projeto, a partir dela é possível utilizar os recursos de forma mais eficiente, além de tornar os objetivos dos processos mais simples e claros. Como qualquer outra metodologia, o *Scrum* possui atividades estabelecidas que dão rumo ao desenvolvimento, a fase que deu início ao projeto foi a Comunicação. Nesta fase do processo, são levantados os requisitos necessários para o funcionamento do sistema, aqueles que agregam valor ao produto ou ao processo.

Buscando compreender melhor as demandas que os usuários podem apresentar durante o uso do sistema, o levantamento de requisitos deu-se utilizando o método de criação de Histórias de Usuário (HU), tais histórias se referem a

situações nas quais os atores do sistema relatam alguma necessidade que possuem e que desejam conquistar com a ajuda do sistema, são importantes pois descrevem as características, resultados e funcionalidades solicitadas sobre um sistema (PRESSMAN, MAXIM, 2021, p. 46)

O trabalho foi dividido em quatro *sprints* principais, a primeira *sprint* focou em atividades de planejamento, tanto de banco de dados quanto do tratamento das informações sobre eventos e usuários, foi também durante esta fase que as conexões e migrações do banco de dados foram realizadas. A segunda focou na organização do projeto e definição das características das principais funcionalidades e interfaces, além de associar as informações necessárias para as requisições de usuário e servidor.

A terceira *sprint* concentrou-se quase que inteiramente em programação de códigos, durante sua execução foram desenvolvidos os *controllers* da aplicação, foram realizadas parte das associações entre tabelas no banco de dados através dos *Models*, além de uma ligeira evolução no design das interfaces. A quarta *sprint* foi marcada pela finalização e testes do projeto, nela foram realizadas as associações faltantes no banco de dados, aprimoramento da lógica dos *Controllers* e *Models*, incrementos nas interfaces de usuário, além de testes de comportamento das funcionalidades.

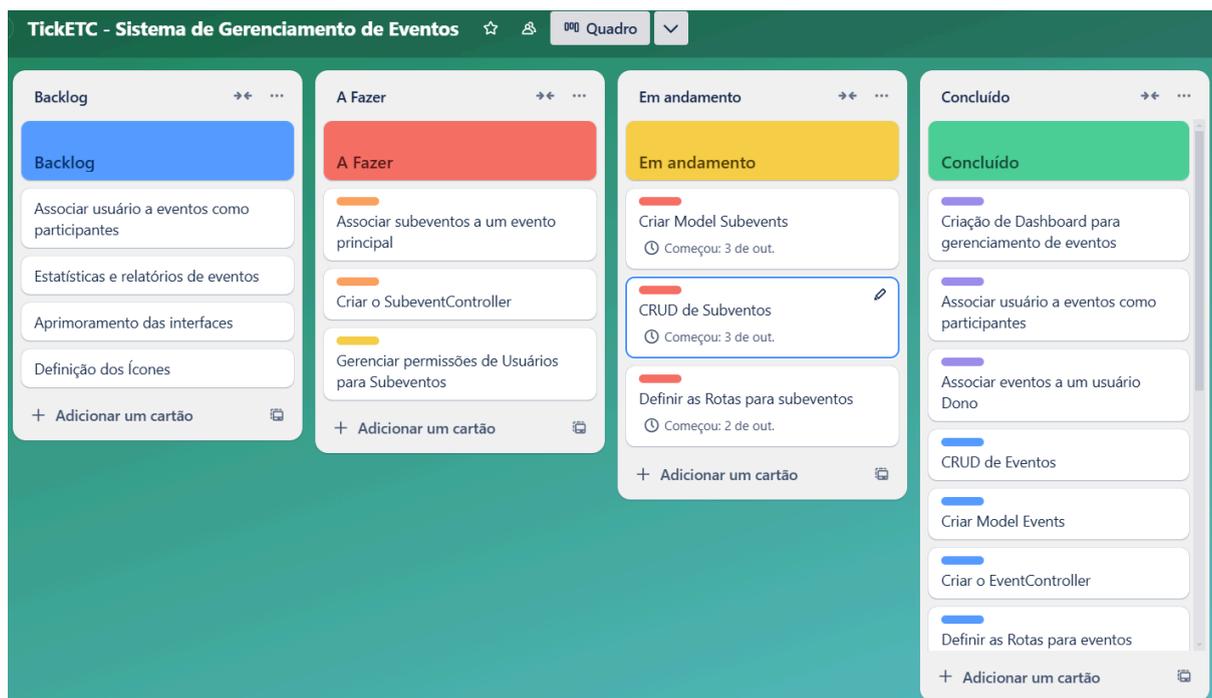
5. Resultados

O desenvolvimento do sistema de gerenciamento de eventos TickETC iniciou com a elaboração das HU, como é possível visualizar no Apêndice I deste trabalho. A partir delas foi possível compreender tanto as necessidades dos usuários, como: registro e login na plataforma; pesquisa e visualização de eventos; inscrição e obtenção de ingressos, e configurações de conta; quanto as demandas dos gerentes, como: cadastro e personalização de eventos conforme regras de negócio; e, relatórios e painéis para gestão mais efetiva.

Observando as necessidades a partir de tais histórias, ressaltou-se a escolha de uma plataforma *web* baseada nos princípios de *e-business*, sistemas mais atuais e relevantes. Esse tipo de sistema se propõe a auxiliar na administração de processos de forma facilitada e conectar melhor os gerentes com usuário final (GONÇALVES, 2017 p. 127).

Seguindo a metodologia *Scrum*, o próximo passo foi a realização do *Product Backlog*, a lista de requisitos do sistema pode ser observada no Apêndice II deste trabalho. Após o levantamento dos requisitos, a separação do trabalho em *sprints* pode ser realizada mais facilmente, agora na fase de planejamento. Para uma melhor visualização do fluxo de trabalho, foi criado um quadro *Kanban* na ferramenta Trello, como mostra a Figura 5.

Figura 5: Quadro *Kanban* durante a *sprint* 4.



Fonte: O Autor

Ainda que contenha grande parte dos requisitos iniciais do projeto, a metodologia *Scrum* libera alterações no *Product Backlog* em fases mais avançadas dos processos, permitindo o projeto a avançar conforme as necessidades encontradas no caminho. O planejamento continuou, agora definido dentro da *Sprint* 1, com a separação das informações necessárias para a criação do banco de dados, tabelas como *users*, *events*, *tickets* e *subevents*, respectivamente, usuários, eventos, ingressos e subeventos, foram esboçadas, dentre as conclusões iniciais foram evidenciadas as seguintes características:

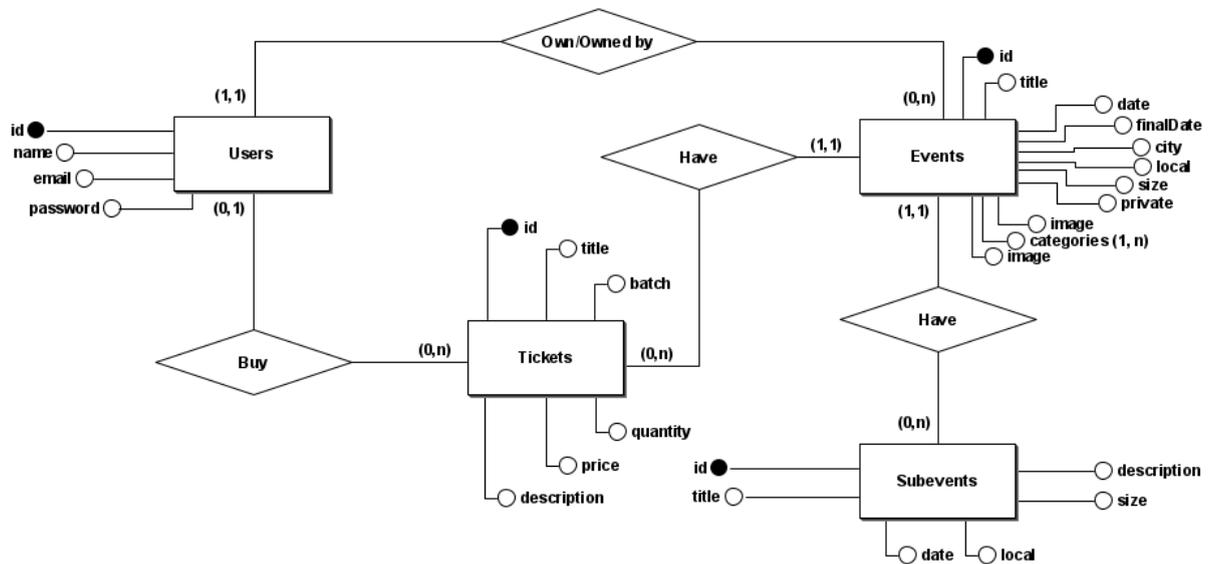
Para a tabela *users* foram designados os dados de: *id*, *name*, *email* e *password*, representando os dados de identificação, nome, email e senha. Na tabela *events*, para uma maior capacidade de personalização, diversos dados foram

selecionados, dentre eles: *id*, *title*, *date*, *finalDate*, *city*, *local*, *size*, *private*, *description*, *categories*, *image* e *user_id*, respectivamente: identificação; título; data de início; data final, no caso de eventos recorrentes ou feiras; cidade; local; tamanho; privado, que pode ser falso ou verdadeiro; descrição; categorias; imagem; e, por fim, identificação do dono do evento.

Já na tabela *subevents* muitas das informações necessárias devem ser ligadas a tabela do evento relacionado, foram optadas uma quantidade menor de características, sendo elas: *id*, *title*, *description*, *date*, *local* e *size*, significando: identificação; título, descrição, data, local e tamanho. Para *tickets*, apenas informações básicas: *id*, *title*, *description*, *batch*, *price* e *quantity*, denotando: identificação, título, descrição, lote, valor e quantidade.

Após isso, foi necessário fazer o planejamento do banco de dados, iniciando com o modelo conceitual, visualizando as entidades e como elas se relacionam entre si, o resultado pode ser visto no diagrama entidade-relacionamento presente na Figura 6.

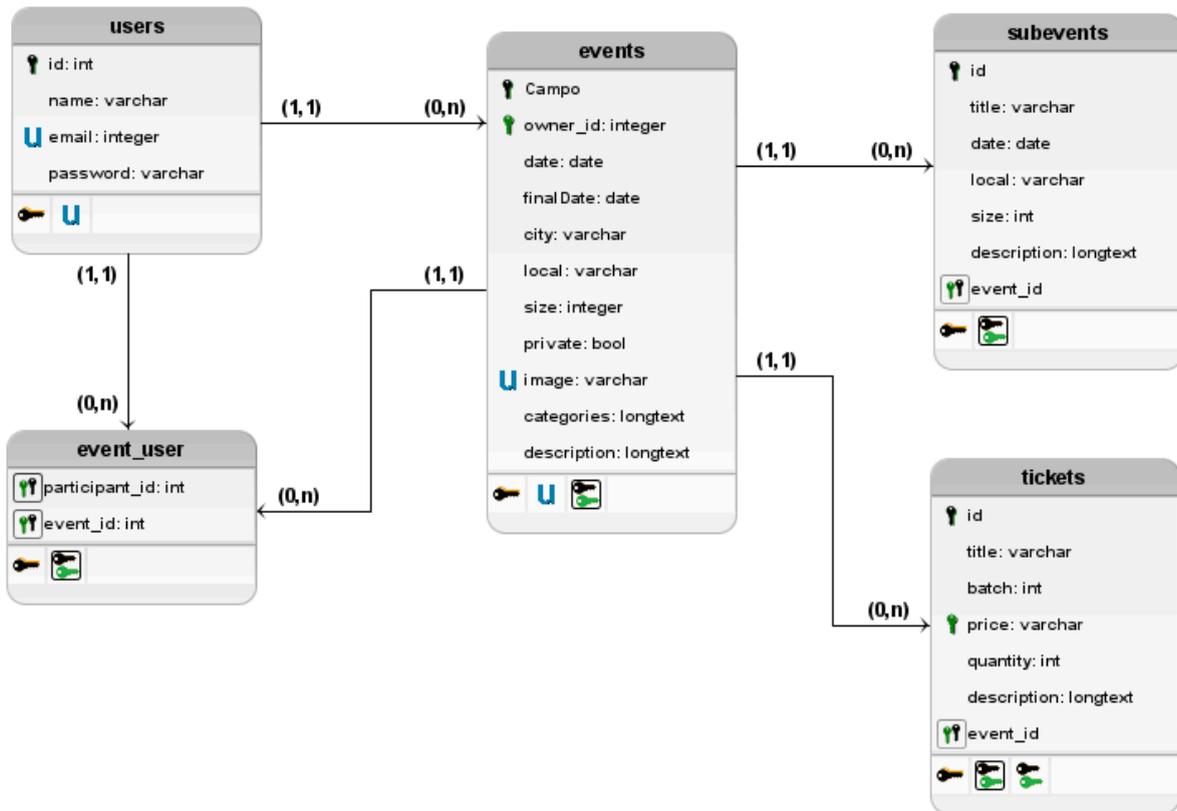
Figura 6: Diagrama entidade-relacionamento do banco de dados.



Fonte: O autor

Logo em seguida, a criação da modelagem lógica do banco de dados foi realizada, mais rigorosa na questão das tabelas necessárias para a organização de relacionamentos no banco de dados, apresentada na Figura 7.

Figura 7: Diagrama lógico do banco de dados.



Fonte: O autor

Com o planejamento e modelagem realizados, deu-se início a fase de construção do sistema, o desenvolvimento do código. Projetos em Laravel podem ser criados via terminal por meio do gerenciador de dependências do PHP, o Composer. Após sua criação, diversas pastas e arquivos são inseridos automaticamente no projeto.

Finalizando as atividades da *Sprint 1*, a conexão com o banco de dados foi criada no arquivo `.env`, responsável pelas conexões, porém sua manipulação se deu por meio das *migrations*, ou migrações, que nada mais são um esquema de definição da base de dados realizadas em código que podem ser migradas ao banco de dados por meio terminal. A Figura 8 exibe a função de criação da tabela de subeventos.

Figura 8: Migration de criação da tabela *sub_events*.

```
public function up(): void
{
    Schema::create(table: 'sub_events', callback: function (Blueprint $table): void {
        $table->id();
        $table->timestamps();
        $table->dateTime(column: 'date');
        $table->time(column: "time");
        $table->text(column: "local");
        $table->integer(column: 'capacity');
        $table->foreignId(column: 'event_id')->constrained();
    });
}
```

Fonte: O autor

Durante a *Sprint 2*, as funcionalidades de login e autenticação de usuários foram realizadas com apoio do pacote de módulos Laravel Jetstream, algumas das *migrations*, dados e tabelas criadas dizem respeito às suas necessidades de armazenamento de informações. O pacote Jetstream é um *kit* de módulos iniciais voltado para implementação de diversas funções ligadas a gerenciamento de sessão de usuário, como login, cadastro, verificação de email, entre outros.

As interfaces visuais, aquelas que os usuários podem acessar e visualizar são criadas na aba de *Views*, onde podem ser separadas por pastas conforme a necessidade do planejamento. Todas as *Views* possuem como extensão padrão *.blade.php*, apontando sua criação com a ferramenta de modelos Blade Templates. Com a sua utilização é possível criar telas de *layouts* que podem ser reutilizadas em outras *views*, diminuindo a quantidade de retrabalho e consequentemente otimizando a utilização do tempo disponível para programação, além disso, o Blade também permite a inserção de funções e métodos escritos em PHP dentro das páginas, facilitando sua criação. A Figura 9 expõe parte de uma página da aplicação que utiliza *layouts* previamente definidos com Blade.

Figura 9: Implementação do Blade na página *dashboard*.

```
@extends(view: 'layouts.main')

@section(section: 'title', content: 'Dashboard')

@section(section: 'content')
    <div class="col-md-10 offset-md-1 dashboard-title-container">
        <h1>Meus Eventos</h1>
    </div>

    <div class="col-md-10 offset-md-1 dashboard-events-container">
        @if (count(value: $events) > 0)
            <table class="table">
                <thead>
                    <tr>
                        <th scope="col">#</th>
                        <th scope="col">Nome</th>
                        <th scope="col">Participantes</th>
                        <th scope="col">Ações</th>
                    </tr>
                </thead>
            </table>
        </div>
    </div>
```

Fonte: O autor

Além da utilização de funções e métodos em PHP na criação das páginas, algumas funções mais simples foram implementadas usando a linguagem Javascript, responsável por tornar as páginas mais dinâmicas, auxiliando na interação do usuário e alteração do layout dos conteúdos. Também para auxiliar na organização da estrutura e estilização, o *framework* Bootstrap foi aplicado na construção das páginas, com ele é possível dividir os conteúdos em uma disposição de grade em diferentes tamanhos e formatos, podendo variar de acordo com as necessidades do projeto, além disso, pode-se utilizar componentes pré-planejados que a ferramenta possui. Para a estilização de botões e facilidade de interpretação de algumas informações, foram utilizados os ícones presentes no pacote de ícones de código aberto Ionicons, da Ionic.

Concluindo a *Sprint 2*, o roteamento das interfaces da aplicação foi realizado, sendo feito de forma separada em um arquivo próprio chamado de Routes, nele, as rotas podem ser criadas e gerenciadas de forma a organizar o encaminhamento de solicitações do usuário para o *controller* apropriado, a Figura 10 mostra o diretório de rotas relacionadas com os Eventos e define suas ações.

Figura 10: Implementação das rotas de eventos.

```
// Página Inicial
Route::get(uri: '/', action: [EventController::class, "index"]);
Route::get(uri: '/events/create', action: [EventController::class, "create"]->middleware(middleware: 'auth'));
// Exibir Evento
Route::get(uri: '/events/{id}', action: [EventController::class, "show"]);
Route::post(uri: '/events', action: [EventController::class, 'store']);
// Dashboard
Route::get(uri: '/dashboard', action: [EventController::class, 'dashboard']->middleware(middleware: 'auth'));
// CRUD de Eventos
Route::delete(uri: '/events/{id}', action: [EventController::class, 'destroy']->middleware(middleware: 'auth'));
Route::get(uri: '/events/edit/{id}', action: [EventController::class, 'edit']->middleware(middleware: 'auth'));
Route::put(uri: '/events/update/{id}', action: [EventController::class, 'update']->middleware(middleware: 'auth'));
// Inscrição em Evento
Route::post(uri: '/events/join/{id}', action: [EventController::class, 'joinEvent']->middleware(middleware: "auth"));
```

Fonte: O autor

A *sprint* 3 foi caracterizada pela codificação das funcionalidades do sistema. Durante sua execução, foram desenvolvidos os *controllers*, camada na qual as solicitações são recebidas, neles, o comportamento das requisições são organizados e separados por grupos lógicos relacionados da mesma classe, no *controller* de eventos, chamado *EventController*, são sequenciadas as funções relacionadas aos Eventos: *index*, para pesquisa no banco de dados; *create*, para criação; *store*, responsável por guardar eventos no banco de dados; *show*, para mostrar detalhes de um evento específico; *dashboard*, para mostrar o painel de gerenciamento; *destroy*, para deletar; *edit*, para edição de dados; *update*, para atualização de dados. Além disso, é no *controller* onde pode-se utilizar funções presentes nos *Models*. A Figura 11 mostra parte do grupo de funções presentes no *EventController*.

Figura 11: Funções de gerenciamento de eventos no *controller*.

```
1 reference | 0 overrides
public function show($id): Factory|View {
    $event = Event::findOrFail(id: $id);
    $eventOwner = User::where(column: 'id', operator: $event->user_id->first()->toArray());
    return view(view: "events.show", data: ['event' => $event, 'eventOwner' => $eventOwner]);
}

1 reference | 0 overrides
public function dashboard(): Factory|View {
    $user = Auth::user();
    $events = $user->events;
    return view(view: 'events.dashboard', data: ['events' => $events]);
}

1 reference | 0 overrides
public function destroy($id): Redirector|RedirectResponse{
    $event = Event::findOrFail(id: $id);
    if($event->image){
        unlink(filename: public_path(path: 'img/events/' . $event->image));
    }
    $event->delete();
    return redirect(to: '/dashboard')->with(key: 'msg', value: 'Evento excluído com sucesso!');
}
```

Fonte: O autor

Nos *Models*, os dados podem ser tratados de forma específica, seja na forma de validação ou na sua definição, como a *array* de *categories*, ou categorias, que se comporta como uma lista no sistema e necessita de tratamento quando requisitada. Além disso, é nos *Models* em que as funções incumbidas de conectar tabelas e programar as regras necessárias para o funcionamento do sistema são definidas. A Figura 12 apresenta o tratamento de dados e as funções responsáveis por relacionar as tabelas no banco de dados, desenvolvidas durante o desenvolvimento da *sprint* 3, presentes no *model* Event, dentre elas, a função *user* define um usuário como dono do evento, a função *users* define vários usuários como participantes em um evento, e a função *sub_events* define um evento que contém vários subeventos.

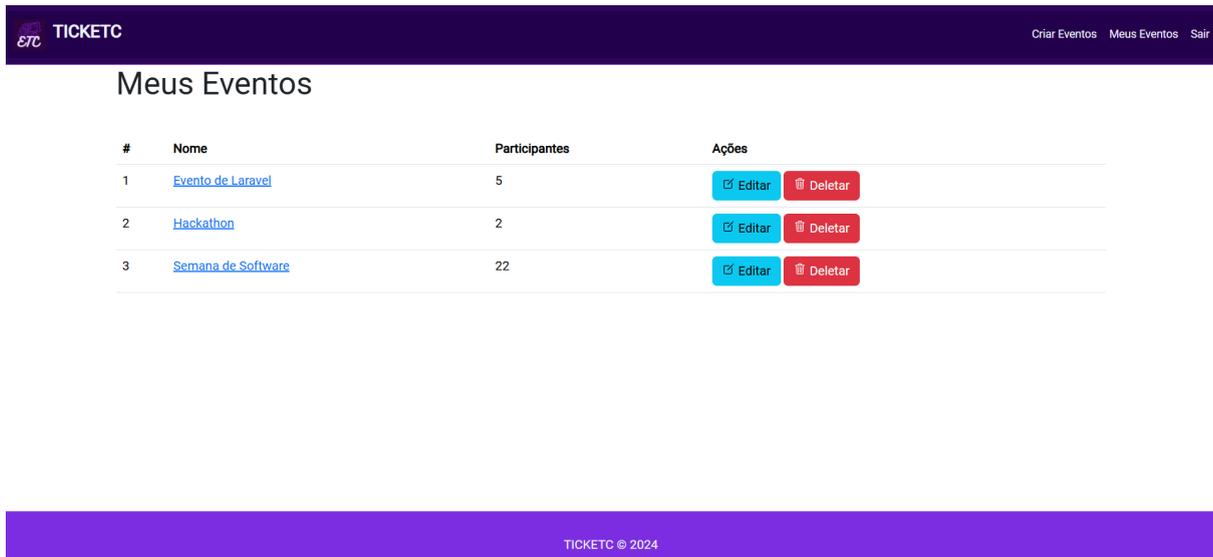
Figura 12: Características do *Model Event*.

```
13 references | 0 implementations
class Event extends Model
{
|   use HasFactory;
|   0 references
|   protected $casts = [
|       'categories' => 'array'
|   ];
|   0 references
|   protected $dates = ['date'];
|   0 references
|   protected $guarded = [];
|   0 references | 0 overrides
|   public function user(): BelongsTo{
|       return $this->belongsTo(related: User::class);
|   }
|   0 references | 0 overrides
|   public function users(): BelongsToMany{
|       return $this->belongsToMany(related: User::class);
|   }
|   0 references | 0 overrides
|   public function sub_events(): HasMany{
|       return $this->hasMany(related: SubEvent::class);
|   }
}
```

Fonte: O autor

O aprimoramento da lógica de programação das ações e funcionalidades presentes nos *controllers* e as associações e tratamento marcaram a execução da última *sprint*, além disso, houve avanço na organização e *layout* das interfaces de usuário. A partir da utilização de todos os elementos integrados, o sistema opera e integra as funcionalidades e interfaces necessárias para o acompanhamento de requisições e acomodação de dados e informações, permitindo um gerenciamento eficiente e organizado. Na Figura 13, a página de *dashboard* lista os eventos criados por um usuário, a quantidade de participantes inscritos até o momento, além de funções para edição de dados e deleção de eventos.

Figura 13: Página de Dashboard do sistema.



The screenshot shows the 'Meus Eventos' (My Events) page of the TICKETC system. The page has a dark purple header with the TICKETC logo on the left and navigation links 'Criar Eventos', 'Meus Eventos', and 'Sair' on the right. Below the header, the title 'Meus Eventos' is displayed. A table lists three events with columns for '#', 'Nome', 'Participantes', and 'Ações'. Each event row includes 'Editar' (Edit) and 'Deletar' (Delete) buttons. At the bottom of the page, a purple footer contains 'TICKETC © 2024'.

#	Nome	Participantes	Ações
1	Evento de Laravel	5	Editar Deletar
2	Hackathon	2	Editar Deletar
3	Semana de Software	22	Editar Deletar

Fonte: O autor

A partir da conclusão das etapas de desenvolvimento, o sistema mostrou obter sucesso em seus objetivos ao facilitar o repasse de dados de eventos aos usuários, assegurando maior clareza nas informações, além de contribuir para uma maior abrangência no controle de eventos, auxiliando os gestores com a definição de suas ações. Dessa forma, o sistema é capaz de aprimorar a experiência tanto dos organizadores quanto dos usuários finais.

6. Considerações Finais

A dificuldade em encontrar uma ferramenta adequada que atenda às características específicas de um projeto de evento pode levar a aplicação de processos confusos de organização por parte dos gerentes, complicando o gerenciamento apropriado e acarretando em prováveis problemas de comunicação entre gerentes e seus públicos.

O sistema gerado pela conclusão deste trabalho mostra-se eficaz na resolução desses problemas, pois possui funcionalidades convenientes de gestão promovendo uma ampla personalização e maior controle sobre informações de eventos, subeventos e suas inscrições. A capacidade de qualquer usuário poder ser capaz de criar eventos ou realizar inscrições se mostra importante por dar mais

liberdade de participação e opções de coordenação mais efetivas para eventos de qualquer tipo e tamanho.

O próximo passo deste trabalho é progredir na implementação de funcionalidades que auxiliem ainda mais na organização de eventos e participantes, a execução em ambiente de projetos reais, realização de ajustes necessários conforme a obtenção de *feedback* dos usuários, além disso, evoluir na implementação de requisitos que melhorem a usabilidade e acessibilidade do sistema.

Referências

AIDA, Kaio M. L.; FERREIRA, Vinícius E.; FONTÃO, Awdren de L. **SISEVENT**: uma proposta de sistema para apoio à gestão de eventos em instituições públicas de ensino. Fundação Universidade Federal de Mato Grosso do Sul, 2023. Disponível em: <<https://repositorio.ufms.br/handle/123456789/7794>>. Acesso em 18 jun. 2024.

BALTZAN, Paige; PHILLIPS, Amy. Sistemas de Informação. Porto Alegre: AMGH, 2012. E-book. ISBN 9788580550764. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788580550764/>. Acesso em: 28 set. 2024.

Bootstrap v5.3.3. Bootstrap Team. Disponível em: <<https://getbootstrap.com/docs/5.3/getting-started/introduction/>>. Acesso em 27 ago. 2024.

DUCKETT, Jon. PHP&MYSQL: desenvolvimento web no lado do servidor. Rio de Janeiro: Editora Alta Books, 2024. E-book. ISBN 9786555205930. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786555205930/>. Acesso em: 12 out. 2024.

ESTEVES, Leandro V.; PACHECO, Lucas V.; DIAS, Thiago M. R.; SILVA, Michael F. da. Tô Nessa! Aplicação web voltada para o mercado de eventos. Centro Federal de Educação Tecnológica, 2017. Disponível em: <<https://www.digddv.cefetmg.br/wp-content/uploads/sites/52/2017/11/To-Nessa1.pdf>>. Acesso em: 29 out. 2024.

GONÇALVES, Glauber R B. Sistemas de informação. Porto Alegre: SAGAH, 2017. E-book. ISBN 9788595022270. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595022270/>. Acesso em: 28 set. 2024.

HEUSER, Carlos A. Projeto de banco de dados - UFRGS. V.4. 6th ed. Porto Alegre: Bookman, 2011. E-book. ISBN 9788577804528. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788577804528/>. Acesso em: 28 set. 2024.

Ionicons 7.1.0. Ionic. Disponível em: <<https://ionic.io/ionicons>> Acesso em 27 ago. 2024.

IVO, A. A.; MARIN, E. C.; SOUZA, L. M. D. **GESTÃO DE EVENTOS: ORIENTAÇÕES BÁSICAS PARA O CONTEXTO DAS UNIVERSIDADES**. Kinesis, [S. l.], v. 32, n. 2, 2014. DOI: 10.5902/2316546416508. Disponível em: <https://periodicos.ufsm.br/kinesis/article/view/16508>. Acesso em: 19 jun. 2024.

KAUSAR, I. Bagwan; SWATI, D. Ghule. A Modern Review on Laravel- PHP Framework. IRE Journals, v. 2, 2019. Disponível em: <<https://www.irejournals.com/formatedpaper/1701266.pdf>>. Acesso em 20 set. 2024.

Laravel 11.x. "Laravel Documentation". Laravel Holdings Inc.. Disponível em: <<https://laravel.com/docs/11.x>>. Acesso em: 27 jun. 2024.

Laravel Jetstream v5.x. Laravel Jetstream. Disponível em: <<https://jetstream.laravel.com/introduction.html>>. Acesso em: 7 ago. 2024.

MASCHIETTO, Luís G.; RODRIGUES, Thiago N.; BIANCO, Clicéres M D.; et al. Processos de Desenvolvimento de Software. Porto Alegre: SAGAH, 2020. E-book. ISBN 9786556900520. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900520/>. Acesso em: 28 set. 2024.

MATIAS, Marlene. Organização de Eventos: Procedimentos e Técnicas. 6th ed. Barueri: Manole, 2013. E-book. ISBN 9788520449028. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788520449028/>. Acesso em: 29 out. 2024.

MENDONÇA, Maria José A.; PEROZIN, Juliana Gutierrez Penna A. **Planejamento e Organização de Eventos**. São Paulo: SRV Editora LTDA, 2014. E-book. ISBN 9788536511030. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788536511030/>>. Acesso em: 18 jun. 2024.

NOGUEIRA, Camila G.; JÚNIOR, Sérgio C.; LIMA, Rosineide Maria de; et al. Planejamento de Eventos. Porto Alegre: SAGAH, 2020. E-book. ISBN 9786556900681. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786556900681/>. Acesso em: 29 out. 2024.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software**. Porto Alegre: Grupo A, 2021. E-book. ISBN 9786558040118. Disponível em:

<<https://integrada.minhabiblioteca.com.br/#/books/9786558040118/>> Acesso em: 18 jun. 2024.

RUBIN, Kenneth S. Scrum essencial: um guia prático para o mais popular processo ágil. Rio de Janeiro: Editora Alta Books, 2017. E-book. ISBN 9788550804118. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788550804118/>. Acesso em: 28 set. 2024.

SARAIVA, Maurício O.; BARRETO, Jeanine S. Desenvolvimento de sistemas com PHP. Porto Alegre: SAGAH, 2018. E-book. ISBN 9788595023222. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595023222/>. Acesso em: 12 out. 2024.

Setor de eventos registra crescimento de 46,6% e se consolida como o maior gerador de empregos no país em 2023. ABRAPE, 2023. Disponível em: <<https://abrape.com.br/setor-de-eventos-registra-crescimento-de-466-e-se-consolida-como-o-maior-gerador-de-empregos-no-pais-em-2023/>>. Acesso em 18 jun. 2024.

SUBECZ, Zoltán. Web-development with Laravel framework. Gradus, v. 8, n. 1, p. 211-218, 2021. Disponível em: <https://gradus.kefo.hu/archive/2021-1/2021_1_CSC_006_Subecz.pdf>. Acesso em 20 set. 2024.

TURBAN, Efraim; VOLONINO, Linda. Tecnologia da informação para gestão. 8th ed. Porto Alegre: Bookman, 2013. E-book. ISBN 9788582600160. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788582600160/>. Acesso em: 28 set. 2024.

Xampp 8.2.12. Apache Friends. Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em: 19 jun. 2024.

ZENKER, Aline M.; SANTOS, Jailson Costa dos; COUTO, Júlia M C.; et al. Arquitetura de sistemas. Porto Alegre: SAGAH, 2019. E-book. ISBN 9788595029767. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595029767/>. Acesso em: 28 set. 2024.

APÊNDICE I - DOCUMENTO DE LISTAGEM DE HISTÓRIAS DE USUÁRIO

A definição de uma HU foi feita a partir dos seguintes elementos:

- Numeração da história precedida por HU;
- Título da história;
- Sendo: Qual ator do sistema participa da história;
- Quero: A demanda do ator;
- Para: O objetivo do ator a partir da demanda;

HU01 - VISUALIZAÇÃO DE EVENTOS

**SENDO USUÁRIO
QUERO ENTRAR NA PLATAFORMA
PARA VER OS EVENTOS QUE IRÃO OCORRER**

HU02 - VISUALIZAÇÃO DE EVENTO ESCOLHIDO

**SENDO USUÁRIO
QUERO SELECIONAR UM EVENTO
PARA PODER VER SUAS INFORMAÇÕES QUE ME INTERESSEM**

HU03 - CADASTRO DE USUÁRIO

**SENDO USUÁRIO
QUERO ME CADASTRAR NA PLATAFORMA
PARA CONSEGUIR MARCAR PRESENÇA EM EVENTOS**

HU04 - LOGIN DE USUÁRIO

**SENDO USUÁRIO
QUERO FAZER LOGIN NA PLATAFORMA
PARA TER ACESSO A TODAS AS FUNCIONALIDADES QUE O SITE
OFERECE**

HU05 - PESQUISA DE EVENTOS

**SENDO USUÁRIO
QUERO PESQUISAR EVENTOS COM BASE EM FILTROS
PARA VISUALIZAR APENAS EVENTOS QUE ME INTERESSAM**

HU06 - INSCRIÇÃO EM EVENTOS

**SENDO USUÁRIO
QUERO ME INSCREVER EM EVENTOS
PARA QUE POSSA PARTICIPAR QUANDO CHEGAR O MOMENTO**

HU07 - CADASTRO DE EVENTOS

SENDO GERENTE DE EVENTOS
QUERO CADASTRAR MEUS EVENTOS
PARA QUE AS PESSOAS POSSAM VISUALIZÁ-LOS NA PLATAFORMA

HU08 - PERSONALIZAÇÃO DE EVENTOS

SENDO GERENTE DE EVENTOS
QUERO TER VÁRIAS OPÇÕES DE PERSONALIZAÇÃO DE
CARACTERÍSTICAS DE EVENTOS
PARA QUE MEUS EVENTOS SEJAM CADASTRADOS DA FORMA QUE
PLANEJEI

HU09 - RELATÓRIO DE INFORMAÇÕES DE EVENTOS

SENDO GERENTE DE EVENTOS
QUERO ACESSAR RELATÓRIOS DE MEUS EVENTOS
PARA OBTER INFORMAÇÕES IMPORTANTES PARA EVENTOS
FUTUROS

HU10 - RELATÓRIO DE PARTICIPANTES DE EVENTOS

SENDO GERENTE DE EVENTOS
QUERO TER ACESSO A LISTA DE PARTICIPANTES DOS MEUS
EVENTOS
PARA PODER GERENCIAR MEUS EVENTOS DA MELHOR FORMA

HU11 - ANÚNCIO DE EVENTOS

SENDO GERENTE DE EVENTOS
QUERO ANUNCIAR MEUS EVENTOS NA PLATAFORMA
PARA QUE MEUS EVENTOS TENHAM MAIOR ALCANCE DE PÚBLICO

HU12 - PLANOS PARA GERENCIAMENTO DE EVENTOS

SENDO GERENTE DE EVENTOS
QUERO TER OPÇÕES DE PLANOS PARA CADASTRAR MEUS EVENTOS
PARA TER ACESSO A DIFERENTES TIPOS DE CADASTROS

HU13 - PAGAMENTO DE EVENTOS

SENDO USUÁRIO

**QUERO TER OPÇÕES DE FORMAS DE PAGAMENTO
PARA PAGAR MINHAS INSCRIÇÕES EM EVENTOS DA FORMA QUE
FOR MELHOR PARA MIM**

HU14 - CONFIGURAÇÃO DE CONTA

**SENDO USUÁRIO
QUERO TER OPÇÕES DE CONFIGURAÇÕES DE CONTA
PARA PERSONALIZAR MINHA CONTA NA PLATAFORMA**

HU15 - PRIVAR EVENTOS

**SENDO GERENTE DE EVENTOS
QUERO PRIVAR MEU EVENTO APENAS PARA PESSOAS COM EMAIL
DA INSTITUIÇÃO QUE TRABALHO/SOU RESPONSÁVEL
PARA QUE APENAS PESSOAS AUTORIZADAS PARTICIPEM**

HU16 - CANAIS DE CONTATO

**SENDO USUÁRIO
QUERO TER OPÇÕES DE CONTATO COM OS RESPONSÁVEIS PELA
PLATAFORMA
PARA SANAR MINHAS DÚVIDAS E MANDAR FEEDBACK**

HU17 - ACESSAR LOCALIZAÇÃO

**SENDO USUÁRIO
QUERO DAR ACESSO A MINHA LOCALIZAÇÃO PARA A PLATAFORMA
PARA VISUALIZAR OS EVENTOS QUE ESTÃO OCORRENDO PERTO DE
MIM**

APÊNDICE II - Listagem de requisitos do sistema (Product Backlog)

1. Requisitos de Banco de dados:

- a. Configuração do Banco de Dados MySQL;
- b. Conectar o banco de dados com Laravel;
- c. Migrar tabelas principais;

2. Requisitos de Sistema:

- a. Importar Laravel Jetstream;
- b. Gerenciar autenticação de usuário;
- c. Criar sessão de usuário;
- d. Gerenciar permissões e níveis de acesso (middleware);

3. Requisitos de Frontend:

- a. Criar telas principais;
- b. Criar layouts;
- c. Organizar a ligação entre as telas;
- d. Inserir formulários;
- e. Ligar os scripts JS e PHP;
- f. Conexão com Bootstrap;
- g. Definição dos Ícones;
- h. Aprimoramento das interfaces;

4. Requisitos de Eventos:

- a. Definir as Rotas para eventos;
- b. Criar o EventController;
- c. Criar Model Events;
- d. Criação de Eventos;
- e. Selecionar eventos do banco de dados;
- f. Atualização de informações de eventos;
- g. Deleção de evento;
- h. Associar eventos a um usuário Dono;
- i. Associar usuário a eventos como participantes;
- j. Criação de Dashboard para gerenciamento de eventos;
- k. Estatísticas e relatórios de eventos;

5. Requisitos de Subeventos:

- a. Definir as Rotas para subeventos;
- b. Criar o Controller;
- c. Criar Model Subevents;
- d. CRUD de Subeventos;
- e. Associar subeventos a um evento principal;
- f. Associar usuário a eventos como participantes;