



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS I - CAMPINA GRANDE  
CENTRO DE CIÊNCIA E TECNOLOGIA  
CURSO DE COMPUTAÇÃO**

**DAVI GONÇALVES CAMPOS**

**APP PARA AGENDAMENTO DE EVENTOS**

**CAMPINA GRANDE, PARAÍBA  
2022**

**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS I - CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA**

**DAVI GONÇALVES CAMPOS**

**APP PARA AGENDAMENTO DE EVENTOS**

Relatório de Técnico apresentado ao Curso de Computação do Centro de Ciência e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Computação.

**Orientador:** Profa. Msc. Ana Isabella Muniz Leite.

**CAMPINA GRANDE, PARAÍBA  
2022**

DAVI GONÇALVES CAMPOS

## APP PARA AGENDAMENTO DE EVENTOS

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 31 de Março de 2022.

*Ana Isabella M. Leite*

---

Prof. MSc. Ana Isabella Muniz Leite (DC - UEPB)  
Orientador(a)

*Kézia de Vasconcelos Oliveira Dantas*

---

Prof. Dra. Kézia de Vasconcelos Oliveira Dantas (DC - UEPB)  
Examinador(a)

*Anderson Y. Iwazaki da Silva*

---

MSc. Anderson Y. Iwazaki da Silva (ICMC/USP)  
Examinador(a)

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

C198a Campos, Davi Goncalves.  
App para agendamento de eventos [manuscrito] / Davi  
Goncalves Campos. - 2022.  
35 p. : il. colorido.

Digitado.  
Trabalho de Conclusão de Curso (Graduação em  
Computação) - Universidade Estadual da Paraíba, Centro de  
Ciências e Tecnologia , 2022.  
"Orientação : Profa. Ma. Ana Isabella Muniz Leite ,  
Coordenação do Curso de Computação - CCT."

1. Desenvolvimento mobile. 2. Gerenciamento de eventos.  
3. Aprendizado baseado em projetos. I. Título

21. ed. CDD 005.369

## **AGRADECIMENTOS**

Em primeiro lugar, a Deus, que me ajudou desde o início até o presente momento em meus anos de estudos.

Aos meus pais e familiares, por todo apoio e ajuda durante todos esses anos, que contribuíram para a minha formação.

Aos meus professores e colegas que conheci na UEPB, que contribuíram para meu aprendizado e minha vida acadêmica, não somente no âmbito acadêmico, mas profissional e da vida.

## RESUMO

O crescimento na indústria de aplicativos móveis e o uso deles nos últimos anos tornou o desenvolvimento desses aplicativos uma das habilidades mais procuradas por grandes empresas no mundo todo, a exemplo Apple, Google e Amazon, uma vez que estas tem relatado números recordes de downloads de seus aplicativos. Assim, a alta demanda por habilidades de desenvolvimento de aplicativos nos mercados corporativo e de consumo estão alinhadas com a rápida expansão dos aplicativos móveis no mercado consumidor, que cada vez mais exige uma aplicação com mais *features* e aprimoramentos das já existentes. Desta forma, este trabalho visa aplicar e aprofundar o conhecimento construído ao longo de todo o curso no desenvolvimento ágil de um aplicativo móvel em um problema real. Buscando compreender a natureza do problema e prover uma solução que seja compatível com as necessidades dos usuários. Nesse contexto, de acordo com o Aprendizado baseado em Projeto, conduzimos o desenvolvimento de um aplicativo móvel para o gerenciamento de eventos. Devido ao risco de disseminação da COVID-19, o planejamento de eventos e alocação segura de pessoas em ambiente fechado durante a pandemia se faz necessário. O Aplicativo móvel ajudará o usuário final a conseguir gerenciar os eventos que o mesmo criou, no qual por uma simples interface, o mesmo usuário poderá criar ou solicitar participação para determinado evento, assim propondo uma melhor experiência do usuário quando comparado com outros aplicativos que já existem no qual grande parte deles possui uma interface complexa para tal atividade. Como resultado, primeiramente destacamos a experiência do aprendizado em projetar e desenvolver uma solução própria, buscando ser criativa e tecnologicamente viável. Além disso, a percepção da importância de habilidades como a de resolução de problemas, liderança e experimentação.

**Palavras-Chave:** Desenvolvimento mobile, Gerenciamento de Eventos, Aprendizado Baseado em Projetos.

## ABSTRACT

The growth in the mobile application industry and its use in recent years has made the development of these applications one of the most sought after skills by large companies worldwide, such as Apple, Google and Amazon, as they have reported record numbers of downloads of Thus, the high demand for application development skills in the corporate and consumer markets is in line with the rapid expansion of mobile applications in the consumer market, which increasingly demands an application with more features and improvements to the existing ones. In this way, this work aims to apply and deepen the knowledge built throughout the course in the agile development of a mobile application in a real problem. Seeking to understand the nature of the problem and provide a solution that is compatible with the needs of users. In this context, according to Project Based Learning, we lead the development of a mobile application for event management. Due to the risk of the spread of COVID-19, event planning and safe allocation of people indoors during the pandemic is necessary. The Mobile Application will help the end user to manage the events that it has created, in which, through a simple interface, the same user will be able to create or request participation for a certain event, thus proposing a better user experience when compared to other applications that already exist in which most of them have a complex interface for such activity. As a result, we first highlight the learning experience in designing and developing an own solution, seeking to be creative and technologically viable. In addition, the perception of the importance of skills such as problem solving, leadership and experimentation.

**Keywords:** Mobile Development, Event Management, Project-based Learning.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>07</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEORICA .....</b>	<b>09</b>
2.1	DESENVOLVIMENTO ÁGIL DE APP MOVEIS .....	09
<b>2.2</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>11</b>
<b>3</b>	<b>METODOLOGIA .....</b>	<b>12</b>
<b>4</b>	<b>APP PARA AGENDAMENTO DE EVENTOS .....</b>	<b>13</b>
4.1	Metodologia de desenvolvimento .....	14
4.2	User stories .....	15
4.3	Casos de uso .....	16
4.4	Descrição arquitetural .....	16
4.4.1	Visão de contexto .....	17
4.4.2	Visão de container.....	18
4.4.3	Visão de componentes .....	19
4.4.4	Visão de funcional .....	20
4.4.5	Visão de lógica .....	23
4.4.6	Visão de deployment .....	24
4.5	Interface com o Usuário .....	25
4.6	Tecnologias utilizadas .....	28
4.6.1	SwiftUI .....	28
4.6.2	NodeJs .....	28
4.7	Validações .....	28
<b>5</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>29</b>
	<b>REFERÊNCIAS .....</b>	<b>30</b>
	<b>APÊNDICE A - Documentação do App para Gerenciamento de Eventos</b>	<b>33</b>

## 1 INTRODUÇÃO

O crescimento na indústria de aplicativos móveis e no uso de aplicativos (referenciados como *Apps* ao longo do texto) nos últimos anos tornou o desenvolvimento de aplicativos uma das habilidades mais importantes por diversas empresas internacionais (Gallaughier et al., 2017). Uma pesquisa recente respondida por mais de 2.000 CIOs de todo o mundo relata alta demanda por habilidades de desenvolvimento de *Apps* nos mercados corporativo e de consumo (Columbus, 2016). Cerca de 80% dos entrevistados afirmam alta demanda por desenvolvedores de *Apps* à medida que o uso de aplicativos cresce em suas empresas e entre seus clientes. Essas descobertas estão alinhadas com a rápida expansão dos *Apps* móveis pelos usuários consumidores. Empresas como Apple, Google e Amazon reportaram números recordes de downloads de *Apps*, como apresentado na Figura 1. Por exemplo, o número de *Apps* disponíveis na Apple App Store cresceu em menos de 12 anos, de apenas 500 aplicativos em 2008 para mais de 3,7 milhões de aplicativos em 2021 (conforme apresentado na Figura 2). A “economia de aplicativos iOS” criou 4,5 milhões de empregos globalmente e trouxe US\$ 189 bilhões para desenvolvedores de aplicativos iOS (Statista, 2022).

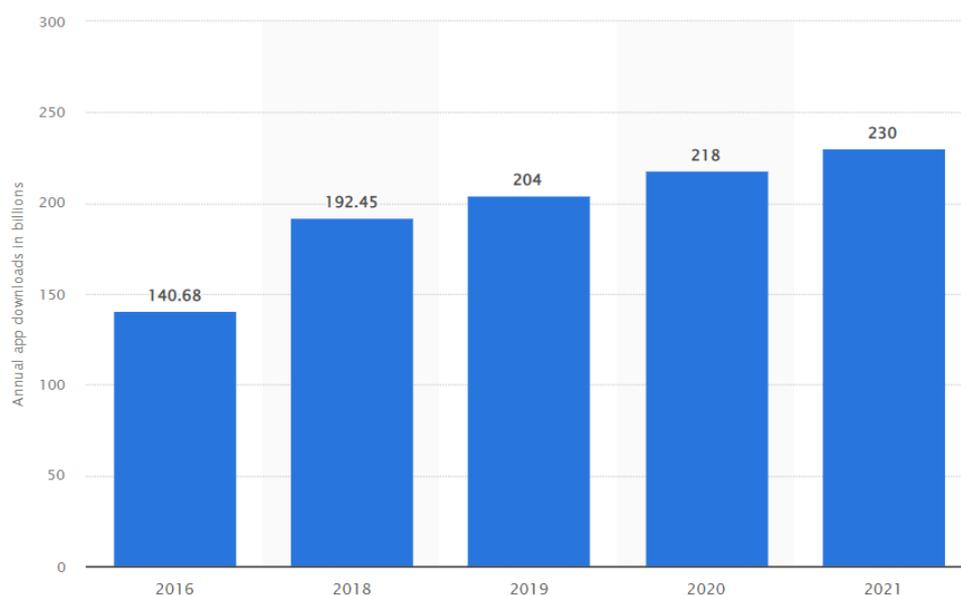


Figura 1 - Número de downloads de *Apps* móveis no mundo todo de 2016 a 2021 (em bilhões) (Statista, 2022).

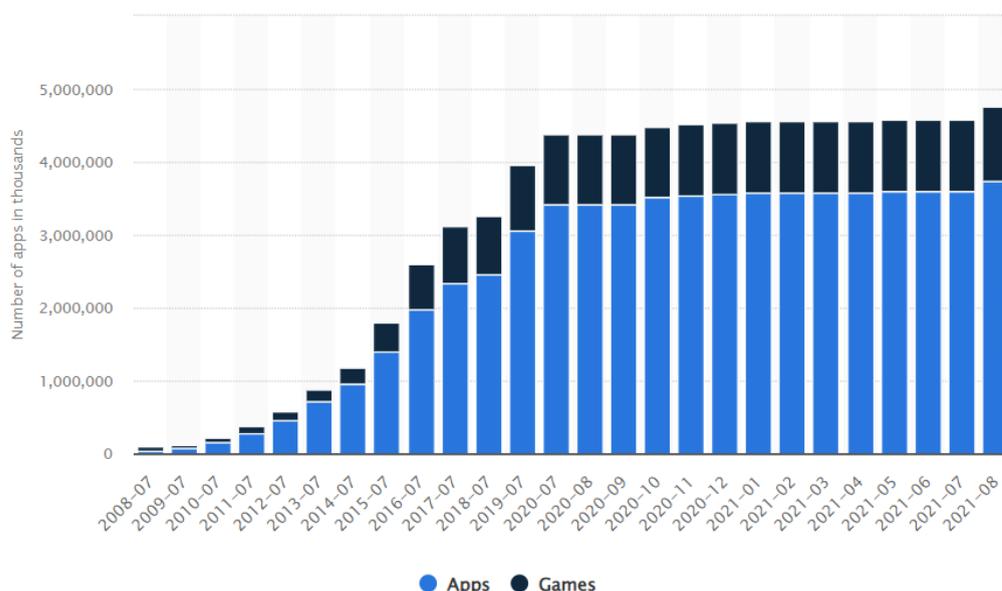


Figura 2 - Número de Apps disponíveis na Apple Store de 2008 a 2021 (Statista, 2022).

Diante desse contexto, buscando o desenvolvimento das habilidades demandadas pelas empresas de Apps móveis, o Aprendizado baseado em Projeto (PBL), processo no qual estudantes aplicam o conhecimento conceitual em contextos da vida real na forma de projetos, permite aprofundar a compreensão dos conceitos estudados ao longo do curso, mas principalmente o desenvolvimento das habilidades como pensamento crítico, liderança, criatividade e experimentação (KOKOTSAKI et al., 2016).

Sendo assim, este trabalho visa aplicar e aprofundar o conhecimento construído ao longo de todo o curso no desenvolvimento ágil de um *App* móvel em um problema real. O projeto alvo desse estudo refere-se ao desenvolvimento de um *App* móvel para o Gerenciamento de Eventos. Este *App* possui características como a criação de eventos, com o propósito de gerenciar participantes que podem estar presentes, e ainda a configuração do espaço definindo os assentos disponíveis ou não, a fim de respeitar as normas sanitárias e de distanciamento.

A relevância do desenvolvimento desse *App* deve-se ao risco de disseminação da COVID-19. De acordo com o Sebrae (2021), desde o início da pandemia, as autoridades recomendaram evitar aglomerações para prevenir o contágio pela doença. Como consequência, mais de 51% eventos foram adiados e até cancelados. Contudo, com o avanço da vacinação no mundo todo, os eventos bem como as aulas nas escolas e universidades tem sido retomadas presencialmente. Entretanto, o planejamento desses eventos e alocação segura

das pessoas em ambientes fechados ainda se faz necessário (VALENTE e GANDRA, 2022).  
O *App* móvel para gerenciamento de eventos

O restante deste trabalho está apresentado da seguinte forma: A Seção 2 apresenta a fundamentação teórica, especialmente os principais conceitos associados ao desenvolvimento ágil de *Apps* móveis juntamente com as ferramentas e tecnologias que foram utilizadas para o desenvolvimento da aplicação. A Seção 3 apresenta um breve comparativo entre a aplicação desenvolvida e soluções já existentes no mercado que propõem uma solução para esse problema acerca do gerenciamento de eventos presenciais. Já a Seção 4 sumariza o método de desenvolvimento utilizado na condução desse trabalho, enquanto a Seção 5 descreve as características do *App* de Gerenciamento de Eventos, desde uma visão geral até os artefatos produzidos durante o desenvolvimento do mesmo. A Seção 6 e 7 abordam o desenvolvimento da aplicação em si, abordando a metodologia e quais validações foram feitas, respectivamente. Por fim, na Seção 8 apresenta as considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção será abordada os conceitos associados ao desenvolvimento ágil de *Apps* Móveis.

### 2.1 Desenvolvimento ágil de *Apps*

O desenvolvimento de *Apps* é realizado em ambientes cada vez mais dinâmicos, cercado de mercados em rápida mudança, requisitos de clientes complexos e mutáveis, pressão de menor tempo de entrada no mercado e tecnologias de informação que avançam rapidamente. Essas características encontradas na maioria dos projetos de desenvolvimento de *Apps*. Fitzgerald et al (2013). descobriram que 78% dos gerentes e executivos do setor entrevistados acham que alcançar a transformação digital é fundamental para suas organizações, enquanto 63% disseram que seu ritmo de mudança tecnológica é muito lento.

De acordo com Khalaf (2015), devido ao aumento na utilização de *Apps* móveis, o desenvolvimento destes envolvem cada vez mais aspetos críticos das empresas. Exemplos como serviços de compartilhamento de carros, sublocações de apartamentos e outros mostram que negócios inteiros agora são baseados em aplicativos móveis. O *feedback* do usuário é essencial no desenvolvimento de *Apps* uma vez que a usabilidade e a experiência do usuário desempenham um papel importante (KRUSCHE e BRUEGGE, 2014).

Os usuários cada vez mais revisam os Apps através de comentários, relatando bugs, sua experiência pessoal e solicitando a inclusão de novos recursos (Pagano e W. Maalaej,2013). Todo esse envolvimento do usuário ajuda os desenvolvedores a entender suas necessidades, quando seu *feedback* é obtido e incorporado ao processo de desenvolvimento (PAGANO e BRUEGGE, 2013).

Schwaber e Beedle (2002) afirmam que para lidar com essas transformações, métodos ágeis como Scrum [6] preconizam flexibilidade, eficiência e velocidade. Muitas empresas de software conseguiram incorporar práticas ágeis em seu fluxo de trabalho de desenvolvimento (VERSION ONE, 2014). Ao mesmo tempo, outras funções dentro dessas empresas (por exemplo, relacionamento com clientes, gerenciamento de produtos e lançamentos de software) ainda seguem ciclos tradicionais medidos em meses ou anos, em vez de introduzir uma abordagem ágil, como, por exemplo, descrito por Pichler (2010).

As características únicas dos métodos ágeis derivam da lista de princípios encontrados no “Manifesto Ágil”: indivíduos e interações são mais importantes que processos e ferramentas, o software funcionando é mais valioso do que a documentação abrangente, a colaboração do cliente é preferível à negociação do contrato e a adaptabilidade é mais valorizada do que criar e seguir um plano (AGILE ALLIANCE, 2001).

Segundo Bosch (2014), o desenvolvimento contínuo de software refere-se à capacidade de desenvolver, liberar e aprender com software em ciclos paralelos. Inclui entregar software aos usuários, coletar e priorizar seu *feedback* e incorporá-lo ao próximo ciclo de desenvolvimento, levando à evolução do software. A capacidade de realizar essas atividades em ciclos curtos requer a incorporação de um fluxo de trabalho de gerenciamento de versões que automatize a entrega de alterações de software aos usuários (HUMBLE and FARLEY, 2010).

Para Fitzgerald et al (2013), a aplicação de metodologias ágeis no desenvolvimento de App já exige uma cultura de abertura entre os membros do projeto e um estilo de gestão solidário. No entanto, se as equipes de projeto não conseguirem adaptar seu fluxo de trabalho de gerenciamento de versões às suas necessidades e restrições específicas, elas podem não usá-lo.

Ao contrário dos aplicativos para desktop convencionais que podem funcionar por alguns anos sem atualização, os Apps móveis são muito mais flexíveis para os usuários. Visto que as demandas destes podem mudar com frequência. Deste modo, os métodos ágeis podem beneficiar o desenvolvimento ágil de Apps com o planejamento em tempo real, ou seja, um planejamento adequado para cada etapa da funcionalidade implementada, sem desperdiçar

tempo e recursos extras. Além disso, pequenos incrementos são criados *Sprint a Sprint*, permitindo, portanto, que mudanças sejam feitas rapidamente, minimizando seu impacto negativo, aumentando um gerenciamento eficiente de riscos e o *feedback* constante do cliente (FLORA et al., 2014).

## 2.2 TRABALHOS RELACIONADOS

O Sympla (SYMPLA,2022) é uma aplicação que funciona tanto em dispositivos móveis como também nos navegadores que permite tanto a participação em eventos, como também a comprar de ingressos e busca de eventos na região. Para o usuário poder participar, o mesmo deve possuir o ticket que pode ser tanto pago quanto gratuito a depender do evento desejado, além de ser possível salvar o ticket no aplicativo. Na questão de organizar evento, o aplicativo dispõe de ferramentas que vão de gratuita até planos pagos, entretanto o mesmo deve se tornar um organizador na plataforma.

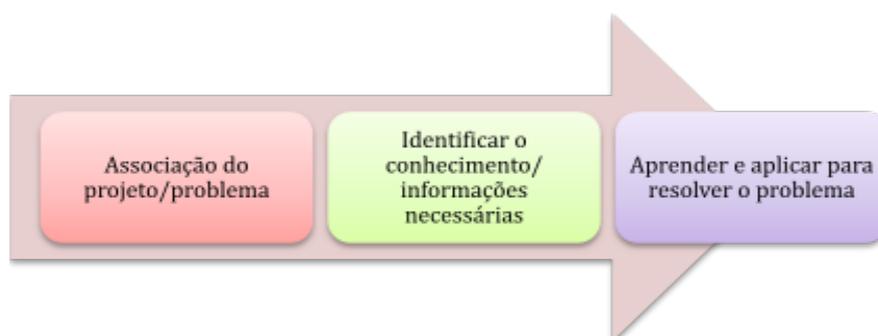
Em comparação como o projeto em questão, vemos que apesar de o Sympla possuir funcionalidades a mais, o mesmo possui uma interface mais complexa, abordando diferentes tipos de eventos que vão desde música até reuniões, ao invés do EventosApp que tem seu foco em eventos de pequeno e médio porte regionais, não sendo possível usuários visualiza-los sem o código do evento, o que deixa a interface e organização mais direta. Acerca do gerenciamento do mesmo, o EventosApp conta com a alocação do espaço no qual podemos definir a organização dos assentos do evento, medida essa para conter a ploriferação do vírus, além de poder organizar as solicitações de maneira simples e rápida, sem a necessidade de enviar algum ticket.

Outro aplicativo similar ao trabalho proposto é o Eventbrite (EVENTBRITE,2022), qual é um aplicativo que funciona em diversas plataformas, desde dispositivos móveis até navegadores web, e possui a mesma proposta que é de criar eventos. No aplicativo, o usuário poderá participar de eventos pagos ou gratuitos, podendo realizar a compra por diversos meios de pagamento. Caso o usuário queira adicionar um evento, o mesmo deverá baixar outro aplicativo para organização, uma vez que o mesmo somente permite a participação.

Quando comparamos o EventosApp com Eventbrite, vemos que não necessita de aplicativo auxiliar para o organizador, e diferente do Eventbrite, há a possibilidade de alocação do espaço e controle direto dos participantes ao invés de tickets. Outra coisa que deve ser mencionada é que o Eventbrite tem seu maior foco em eventos maiores, ainda que seja possível criar eventos locais.

### 3 METODOLOGIA

A abordagem Aprendizado Baseada em Projetos (PBL) (COCCO, 2006) foi utilizada neste trabalho, a fim de que a condução do aprendizado fosse realizada através da investigação e solução de problemas no contexto de um problema real. A Figura 3 apresenta as três fases do PBL seguidas nesse trabalho. A fase 1 (Associação do projeto/problema), obtivemos a compreensão do problema, dos objetivos. Esta fase do PBL está relacionada as atividades: Planejamento e a análise de tarefas (estabelecimento de metas, identificar os recursos necessários, estabelecer um cronograma de tarefas, estabelecer os papéis dos membros da equipe e comunicar os planos). Vale destacar que durante esta fase, as atividades PBL incluem a ativação de conhecimento prévio. Já a fase 2 (Identificar o conhecimento e informações necessárias) inclui ciclos iterativos de coleta de informações, reflexão e testes de possibilidades (por meio de experimentação e/ou prototipação) e revisão conforme necessário. Por fim, a fase 3 (Aprender e aplicar para resolver o problema) reflete sobre os resultados gerais da aprendizagem e os resultados do processo, à medida que se relacionam com os objetivos e expectativas do projeto. Durante esta fase, o projeto/solução é apresentada, assim como é realizado a avaliação de satisfação, de modo a identificar ajustes que precisam ser feitos até que o produto final possa ser liberado e compartilhado. As fases 2 e 3 apesar de apresentadas de forma linear, foram executadas de forma iterativa e incremental, visto que durante estas foi realizado o desenvolvimento ágil do App.



*Figura 3- Aprendizado Baseado em Projetos (COCCO, 2006).*

#### 4 APP PARA AGENDAMENTO DE EVENTOS

Como mencionado anteriormente, com o advento da Covid-19, a forma de reunir pessoas em determinado lugar para algum fim se tornou uma atividade complicada, principalmente no quesito organização e distribuição de pessoas. Tal problema foi agravado mais ainda com surgimento de novas ondas e restrições sanitárias, as quais tornaram muitas vezes as reuniões mais difíceis de serem executadas devido a ocupação e distribuição de participantes nos ambientes. Para amenizar estas dificuldades encontradas na organização e realização de eventos, propomos um App para o sistema IOS.

O aplicativo permite como função primária a possibilidade de criação de um evento no qual o usuário poderá adicionar as informações acerca do mesmo, como nome, descrição e data. Além disso, como diferencial, o usuário no momento da criação poderá configurar a alocação do espaço definindo quais assentos devem ser ou não utilizados no momento do evento a fim de respeitar as normas sanitárias e distanciamento entre participantes, tal alocação é feita demarcando quais assentos estão disponíveis e quais estão indisponíveis de acordo com o local a ser realizado.

Além da criação do evento, o aplicativo ainda permite ao criador do evento a capacidade de gerenciar quais usuários poderão participar ou não do evento. Vale ressaltar que para participar de um evento, o usuário deve inserir um código que corresponde a determinado evento, que o redirecionará para a página principal daquele evento, possibilitando solicitar sua presença no evento. A possibilidade de definir quais usuários irão participar do evento por meio das solicitações que outros usuários podem fazer ao evento em questão e, assim definir se aquele usuário poderá participar ou não do evento. Quando uma solicitação é confirmada, tal usuário poderá visualizar na tela principal que está participando de fato daquele evento.

A lista completa das *features* do App, bem como toda a documentação produzida ao longo do seu desenvolvimento é apresentada mais a seguir nessa Seção. Com tais *features*, o aplicativo tornou o processo de controle dos eventos uma atividade mais simples, contribuindo para uma melhor experiência tanto para o administrador do evento como para os participantes.

#### 4.1 Metodologia de desenvolvimento

A metodologia utilizada para o desenvolvimento do aplicativo se baseia no Scrum, metodologia ágil que permite a realização de atividades de maneira incremental e cíclica, ao invés de metodologias que sejam sequenciais no desenvolvimento. O projeto foi composto por 3 releases de 10 Sprints no total, cujo tamanho de cada Sprint corresponde a uma semana.

A primeira release consiste no planejamento do aplicativo como um todo no qual ao final temos toda parte de arquitetura como também escopo e prototipação para as demais etapas do desenvolvimento. Seguindo, temos o início do desenvolvimento, no qual começa-se pela parte lógica, ou seja, o servidor juntamente com a conexão com o banco de dados, implementando features desde o login até o gerenciamento dos participantes.

Por último, temos o desenvolvimento do aplicativo integrado com a release anterior, o qual permite que o sistema anteriormente planeja seja entregue em funcionamento.

RELEASES	SPRINTS
<b>1 Planejamento</b>	
1.1 Visão do projeto	4 Sprints
1.2 Definição de Objetivos	
1.3 Definição de requisitos	
1.4 Diagrama de caso de uso	
1.5 Diagrama de contexto	
1.6 Prototipação	
1.7 User stories	
1.8 Estudo de usuário (Qualificação Usuário)	
1.9 Definição de arquitetura	
1.9.1 Visão de contexto (diagrama de contexto)	
1.9.2 Visão de Funcional (diagrama de casos de uso)	
1.9.1 Visão de Lógica (Modelagem do BD, diagrama de classes, diagrama de atividades, <b>diagrama de sequencia</b> )	
1.9.1 Visão do Deployment (diagrama de blocos/componentes)	
<b>2 Desenvolvimento Backend</b>	
2.1 Criação do módulo Usuário	3 Sprints
2.2 Criação do módulo de autenticação	
2.3 Criação do módulo de Evento	
2.4 Criação do módulo de gerenciamento de eventos	
<b>3 Desenvolvimento Front-end</b>	
3.1 Criação das telas Login e Usuário	3 Sprints
3.2 Criação das tela adicionar e listar eventos	
3.3 Criação da tela Explorar	
3.4 Criação das telas de evento e Gerenciamento	
<b>Total</b>	<b>10 Sprints</b>

Figura 4 - Planejamento das Releases.

## 4.2 User Stories

O *backlog* do produto é composto pelas *user stories*, listadas na Tabela 1. Essas *user stories* estão divididas em três *Epics*: Usuário, Eventos e Gerenciar eventos. A primeira *Epic* é composta exclusivamente pelas *user stories* referente ao usuário, desde o processo de autenticação à alteração dos dados. A segunda *Epic* é composta pelas *users stories* relacionadas a Eventos, como a adição e visualização de eventos no sistema. Por último, temos a *Epic* composta pela *users stories* envolvendo o gerenciamento dos eventos cujo objetivo é gerenciar as informações dos eventos criados, tais como nome, descrição, a ocupação e aceitar/recusar as solicitações recebidas. A cada *user story* foi associado uma prioridade, qual pode ser *Must* ou *Would*. A prioridade *MUST* indica que tal *user story* é fundamental para o funcionamento adequado do sistema, enquanto a prioridade *WOULD* significa que tal *característica* agregaria somente na experiência do usuário, não sendo considerada uma característica essencial ao sistema.

Tabela 1 - *User stories* da aplicação

Épico	ID	História de usuário	Crerios de aceitaço	Prioridade
E01 - Usuário	US01	Eu, como usuário, gostaria de poder alterar meus dados básicos da minha conta	Possibilidade de alterar nome, sobrenome, email	Must
	US02	Eu, como usuário, desejo poder adicionar ou alterar a imagem do meu perfil para identificaço	Possibilidade de adicionar ou alterar imagem a partir da imagem da galeria ou da câmera	Must
	US03	Eu, como usuário, desejo poder alterar minha senha	Senhas devem ser criptografada Para alterar senha deve ser solicitado a senha atual	Must
	US04	Eu, como usuário, gostaria de poder deslogar do sistema afim de não ficar logado em um dispositivo	Caso o aplicativo encerre, a sessão será expirada automaticamente	Must
E02 - Eventos	US05	Eu, como usuário, gostaria de poder criar um novo evento	Possibilidade de adicionar nome, descriço, data, capacidade, imagem	Must
	US06	Eu, como usuário, desejo poder adicionar um novo evento a partir de um PIN	PIN deve ser válido	Must
	US07	Eu, como usuário, gostaria de favoritar meus eventos afim de facilitar na busca	Os favoritos devem possuir funço de busca geral e especifica por favoritos	Would
	US08	Eu, como usuário, gostaria de pesquisar os eventos que criei ou adicionei	Possibilidade de buscar por nome e tipo	Would
	US09	Eu, como usuário, desejo visualizar informaçoes como data sem a necessidade de ver todas informaçoes na tela	Deve ser exibido um badge com informaçoes resumida do evento	Would
E03 - Gerenciar eventos	US010	Eu, como usuário, gostaria de alterar as informaçoes básicas do eventos que eu criei	Possibilidade de alterar nome, descriço, data, capacidade, imagem	Must
	US011	Eu, como usuário, gostaria de confirmar ou negar a presen de um usuário para o evento	Quando uma presença é negada, não há possibilidade de reverter a escolha	Must
	US012	Eu, como usuário secundário, desejo poder participar de um evento a partir do que é exibido	Para a presença ser confirmada, o usuário que criou o evento deve confirmar	Must
	US013	Eu, como usuário, gostaria de poder cancelar um evento que eu criei	Quando um evento for cancelado, seu status deve ser alterado para cancelado	Must
	US014	Eu, como usuário secundário, gostaria de saber se minha presença está confirmada em um evento	Status da presença deve ser atualizado e exposto na lista de eventos	Must
	US015	Eu, como usuário secundário, gostaria de poder cancelar minha presença em um evento	Caso a presença seja cancelada, uma nova confirmaço do usuário criador do evento dever ser efetuada	Would

### 4.3 Casos de uso

O diagrama de casos de uso documenta o que o sistema faz do ponto de vista do usuário, descrevendo as principais funcionalidades do sistema e sua interação com os usuários. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz.

Os principais casos de uso da aplicação desenvolvida são: logar no sistema, adicionar evento, participar de evento, gerenciar eventos criados e gerenciar perfil, conforme apresentado na Figura 5. O sistema como todo possui dois atores, sendo eles o principal e secundário. O usuário principal podemos considerar como o usuário ativo da aplicação como um todo, já o secundário aquele usuário que envia uma solicitação na qual depende de outro usuário para que essa possa ser realizada. Uma solicitação pode ser considerada atividades como participação em um evento, na qual é necessário que o usuário que criou o evento aceite tal solicitação.

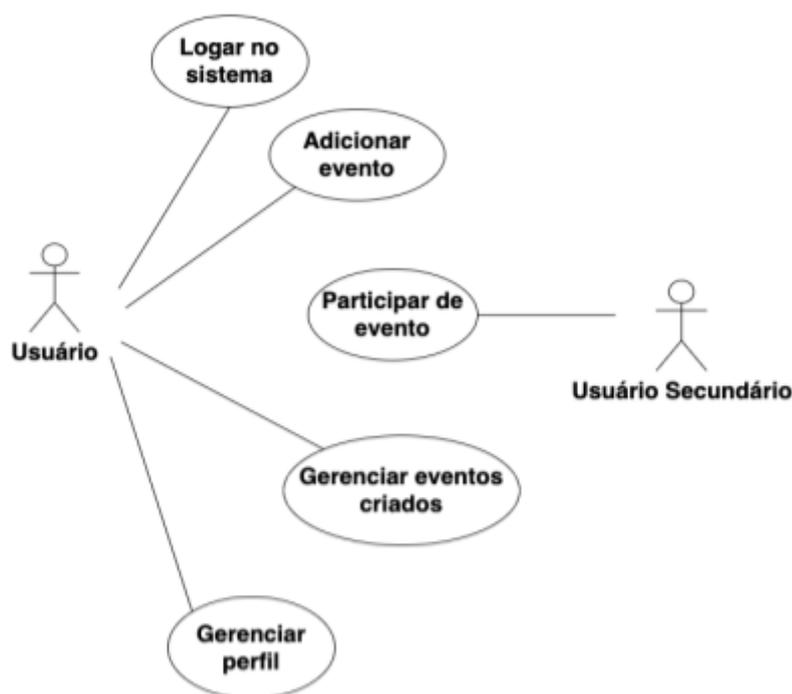


Figura 5 - Diagrama de caso de uso da aplicação.

### 4.4 Descrição Arquitetural

A arquitetura de software tem como principal objetivo descrever como o software se comporta sob diferentes perspectivas e pontos de vista. No desenvolvimento ágil de apps, a arquitetura tem um papel fundamental, que é definir não somente a estrutura, mas também o escopo que a solução se encontra, para que assim possa ser implementado de maneira correta.

Permite aprofundar a comunicação sobre os requisitos e/ou atributos de qualidade que são significantes para o design da solução e aceitação do cliente.

A descrição da arquitetura do App proposto foi concebida seguindo a abordagem C4<sup>1</sup>, na qual são apresentadas as visões arquiteturais de contexto, container e componente. Esta abordagem C4 fornece uma maneira para as equipes de desenvolvimento de software comunicarem de forma eficiente e eficaz sua arquitetura de software, em diferentes níveis de detalhe. Ela é recomendada quando requer compartilhamento rápido e eficiente e atualizações constantes da arquitetura de software durante o desenvolvimento contínuo do software. É adequado para a visão estática da arquitetura de software. Além disso, complementamos as visões propostas pela C4 com a visão funcional, através do diagrama de atividades descrevemos o comportamento das principais funções do App, bem como a visão lógica, descrevendo como os dados são organizados na base de dados e, por fim, a visão de *deployment*, apresentando uma visão geral de como os componentes estão implantados.

#### 4.4.1 Visão de Contexto

A visão de contexto representa o nível mais alto em abstrações que representam o valor dos sistemas existentes ou sistemas em desenvolvimento e a interação entre esses sistemas. A Figura 6 corresponde à visão de contexto do sistema de Gerenciamento, no qual um usuário cadastrado no mesmo, pode tanto visualizar como gerenciar eventos por meio do sistema de agendamento, este é responsável por executar a lógica de negócio. Além desse, temos o mainframe que é nada mais que o servidor onde são executadas as operações acerca dos eventos e usuários, permitindo compartilhamento dos dados.

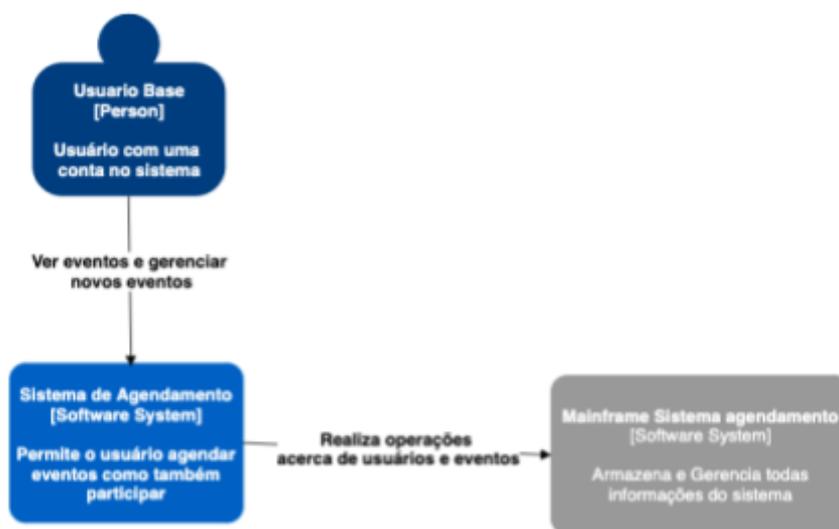


Figura 6 - Diagrama de contexto

<sup>1</sup> <https://c4model.com/>

#### 4.4.2 Visão de Container

A visão de Container representa uma visão interna dos sistemas de software; geralmente aplicativos ou soluções para armazenamento de dados (um conceito diferente para containers no *Docker*). Os exemplos de um contêiner são servidores ou aplicativos cliente (por exemplo, web, móvel ou PC), aplicativos CLI, processos em lote, bancos de dados, armazenamento de *blob*, sistemas de arquivos ou scripts de shell. Refere-se principalmente a um software implantado de forma independente.

A Figura 7 apresenta as partes internas que compõem o sistema de Gerenciamento de Eventos. Este é composto por uma *web application* responsável pelas funcionalidades essenciais acerca dos eventos e usuários e por salvar as informações no banco de dados para persistência. Além da *web application*, encontra-se também um aplicativo IOS o qual realiza a conexão entre usuário e a *web application* por meio de chamadas JSON/HTTP, funcionando como intermediador entre ambos, permitindo ao usuário executar as funcionalidades do sistema. Por sua vez, a persistência é realizada através de um banco de dados relacional MySQL que armazenará as informações acerca dos usuários e os eventos.

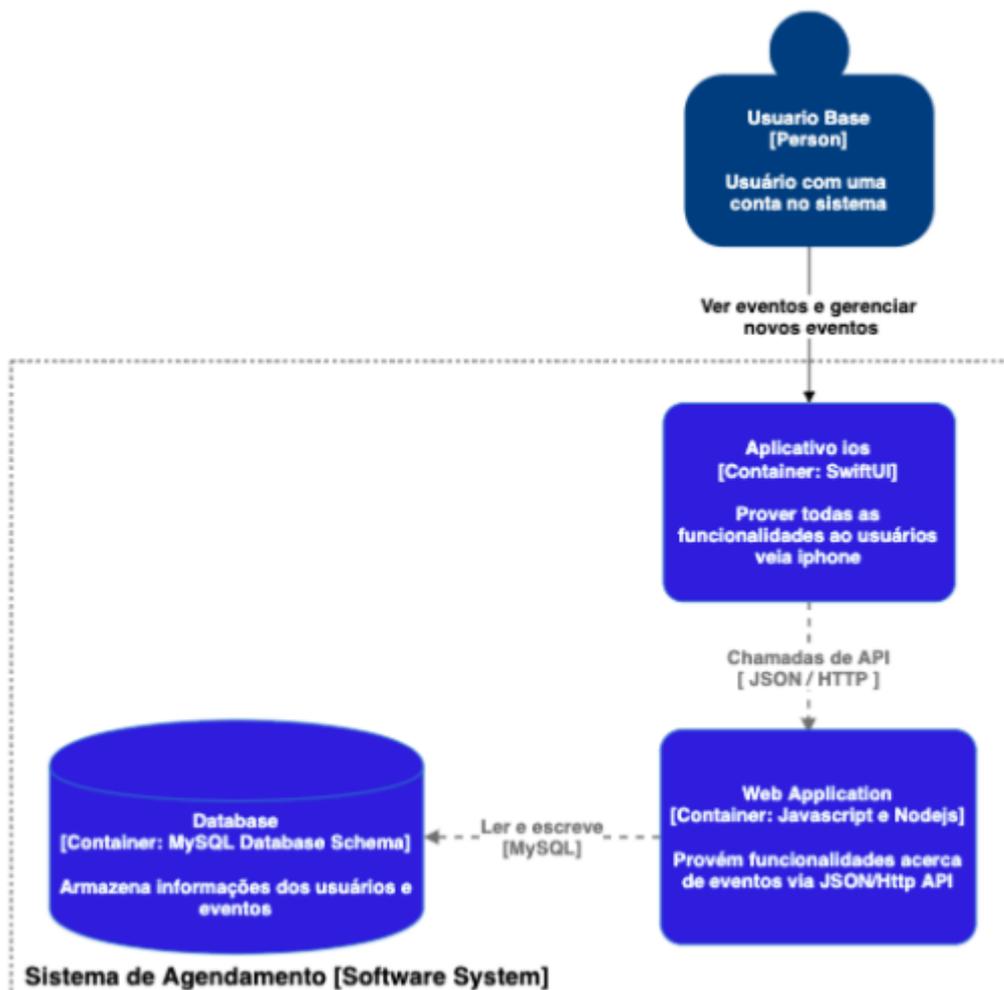


Figura 7 - Diagrama de container do App para o Módulo do Sistema de Agendamento

#### 4.4.3 Visão de Componentes

A visão dos componentes representa o detalhamento dos módulos ou um conjunto de interfaces que podem ser agrupadas como uma unidade funcional. Em termos de Java ou C#, eles podem ser vistos como um conjunto de classes implementadas para interfaces ou pacotes. Eles diferem do Container, pois não podem ser implantados de forma independente.

A Figura 8 apresenta uma descrição por meio dos componentes que formam o sistema de agendamento, no caso a *web application* que realiza a conexão entre usuário e banco de dados. A *web application* por sua vez é desenvolvida em 2 camadas, sendo a primeira de controladores, responsável pelas rotas da API. A segunda consiste na camada de repositórios a qual esses são responsáveis por salvar e manipular os dados diretamente com o banco de dados.

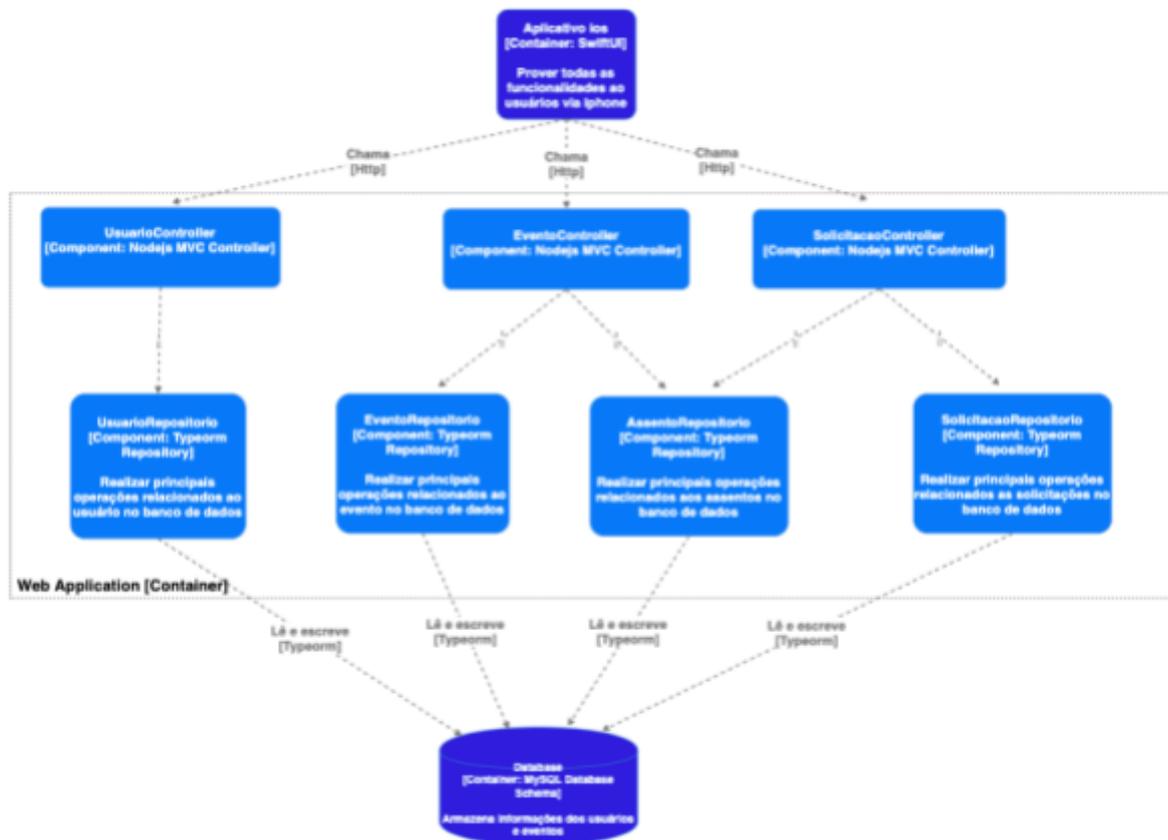


Figura 8 - Diagrama de componentes da *web application*

Na primeira camada temos os controladores de rotas responsáveis pela autenticação/gerenciamento de usuários, eventos e solicitações dos eventos. Já na segunda camada temos os repositórios de usuário, eventos, assentos e solicitações, os quais esses manipulam os objetos diretamente com o banco de dados, o primeiro dos usuários, outro os eventos, assentos e solicitações respectivamente.

#### 4.4.4 Visão de Funcional

O diagrama de atividades visa descrever o fluxo de controle de uma atividade para outra. Desta forma, esse diagrama permite uma melhor compreensão sobre o comportamento das funções do sistema e sua interação com o usuário.

A Figura 10 apresenta o diagrama de atividade de Login no sistema, em caso de sucesso, o usuário poderá realizar as atividades que o sistema propõe, caso contrário o mesmo não acontecerá.

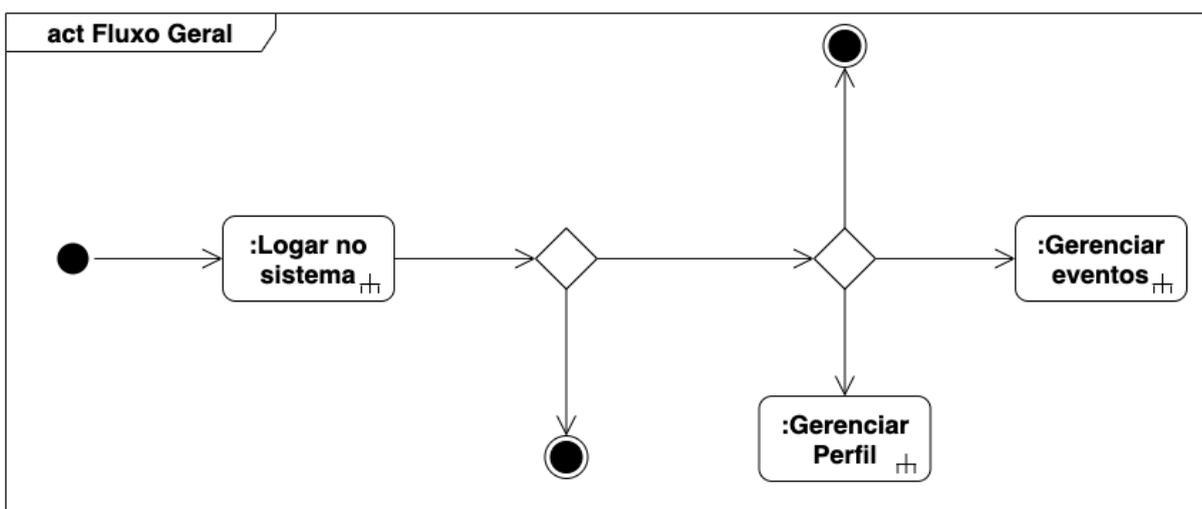


Figura 9 - Diagrama de atividades do sistema

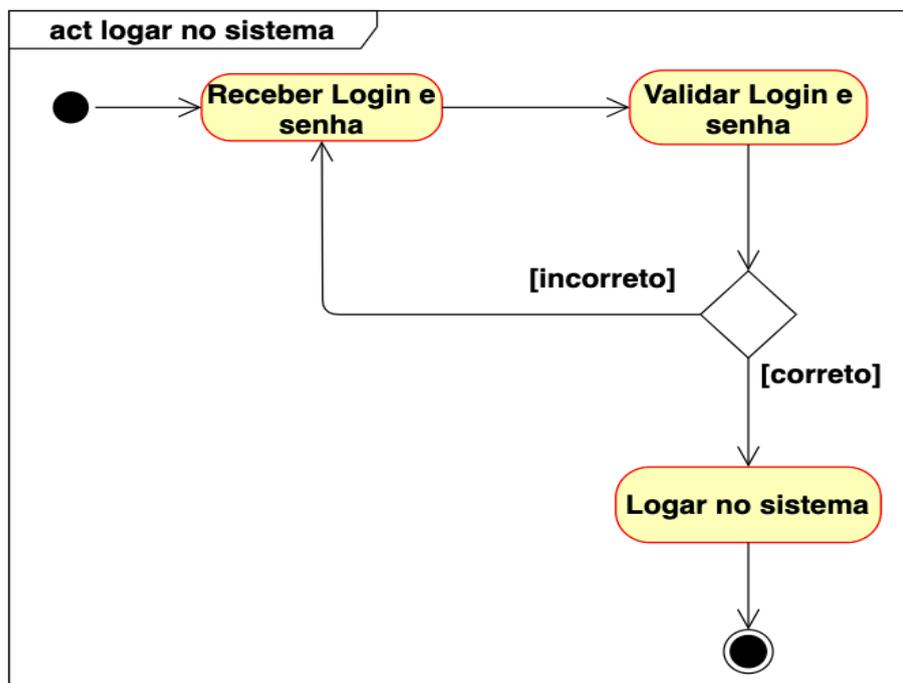


Figura 10 - Diagrama de atividade do processo de logar no sistema

A aplicação vista como um todo é dividida em dois módulos no que se diz a funcionalidade, sendo eles relacionados aos eventos (Figura 11) e o perfil (Figura 12).

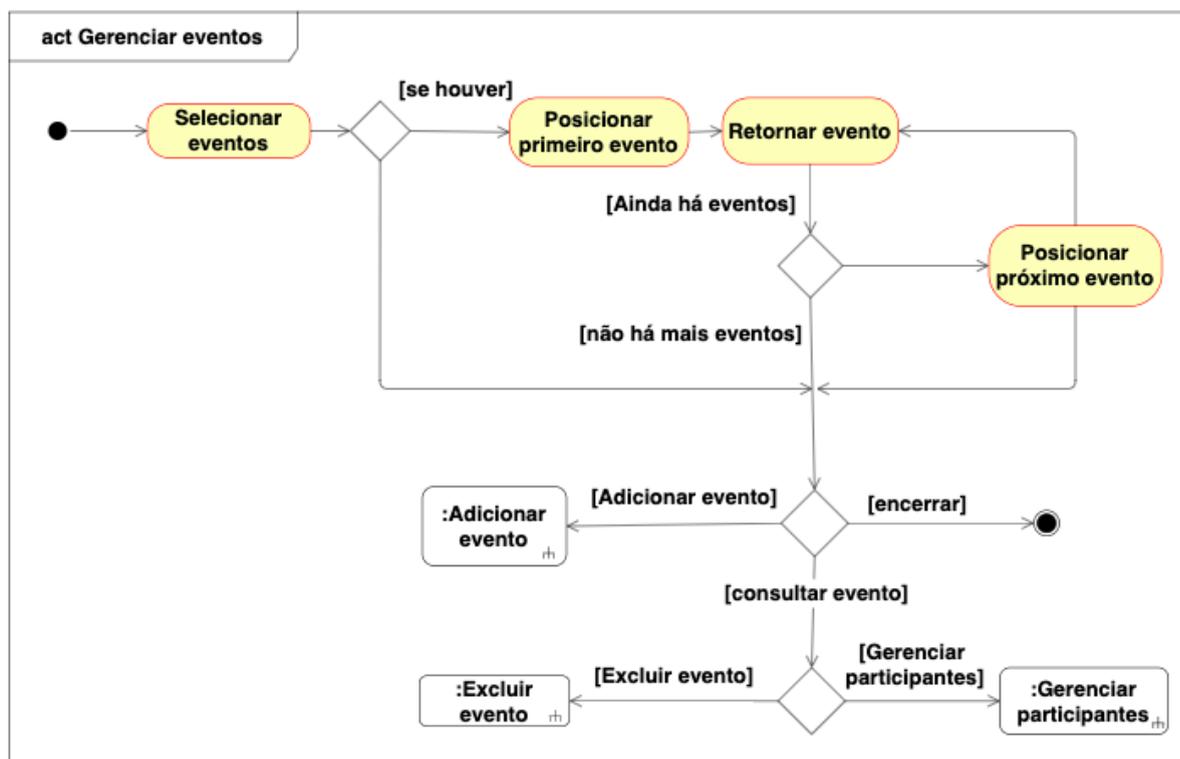


Figura 11 - Diagrama de atividade do gerenciar eventos

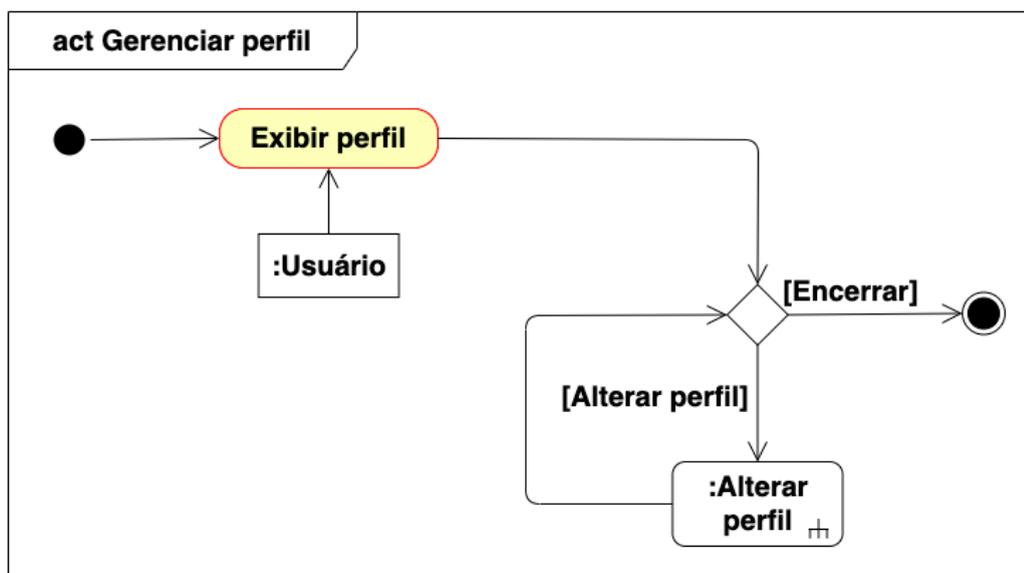


Figura 12 - Diagrama de atividades do gerenciar perfil

Quando selecionamos o módulo de usuário, que no aplicativo será por meio da tela de perfil, o usuário poderá realizar atividades no que se diz a respeito do seu perfil como nome, *email* e senha. Já o outro módulo se refere às operações acerca do evento. Primeiramente, temos o caso básico de gerenciamento de evento, que o mesmo é invocado quando o usuário abre a tela de seus eventos, na qual o usuário por meio dela poderá tanto adicionar o evento escolhido e caso seja o proprietário de um dos eventos, o mesmo poderá gerenciar os participantes, alterar informações ou até mesmo excluí-lo. Quando o usuário deseja realizar o gerenciamento dos participantes do evento que o próprio criou, o fluxo (Figura 13) segue a lógica do anterior, no qual o usuário poderá confirmar ou negar a presença de um usuário que solicitou, como também cancelar a presença de um participante.

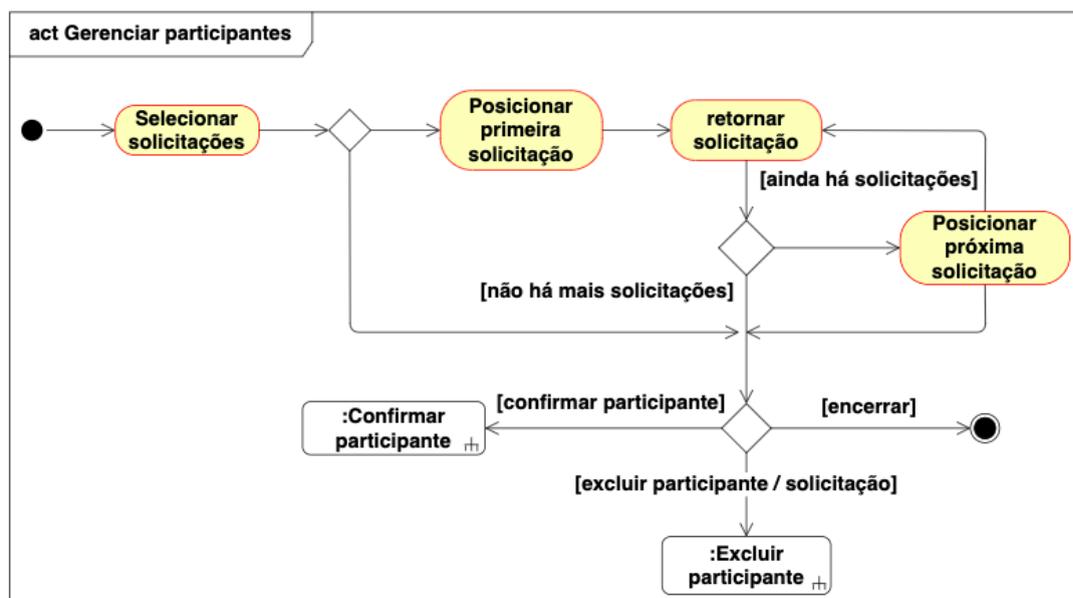


Figura 13 - Diagrama de atividade de gerenciar participantes

#### 4.4.5 Visão de Lógica

A modelagem dos dados auxilia no desenvolvimento e evita que várias alterações sejam necessárias para corrigir erros de concepção provenientes de falhas durante a análise, ou ainda por problemas de comunicação entre os membros da equipe. Nesse sentido, a Figura 14 apresenta o diagrama de Entidade-Relacionamento como uma representação do domínio do Gerenciamento de Eventos, facilitando a compreensão sobre as entidades ou partes envolvidas no domínio. O modelo é composto por 4 entidades básicas: Usuário, Evento, Assento e Solicitação. A entidade Usuário e Evento são responsáveis por representar os usuários e eventos criados respectivamente. Já a entidade Assento corresponde aos lugares de cada evento, armazenando seu status, numeração e a qual evento ele pertence. Por último temos a entidade Solicitação a qual interliga os participantes aos eventos que ele deseja ou está participando.

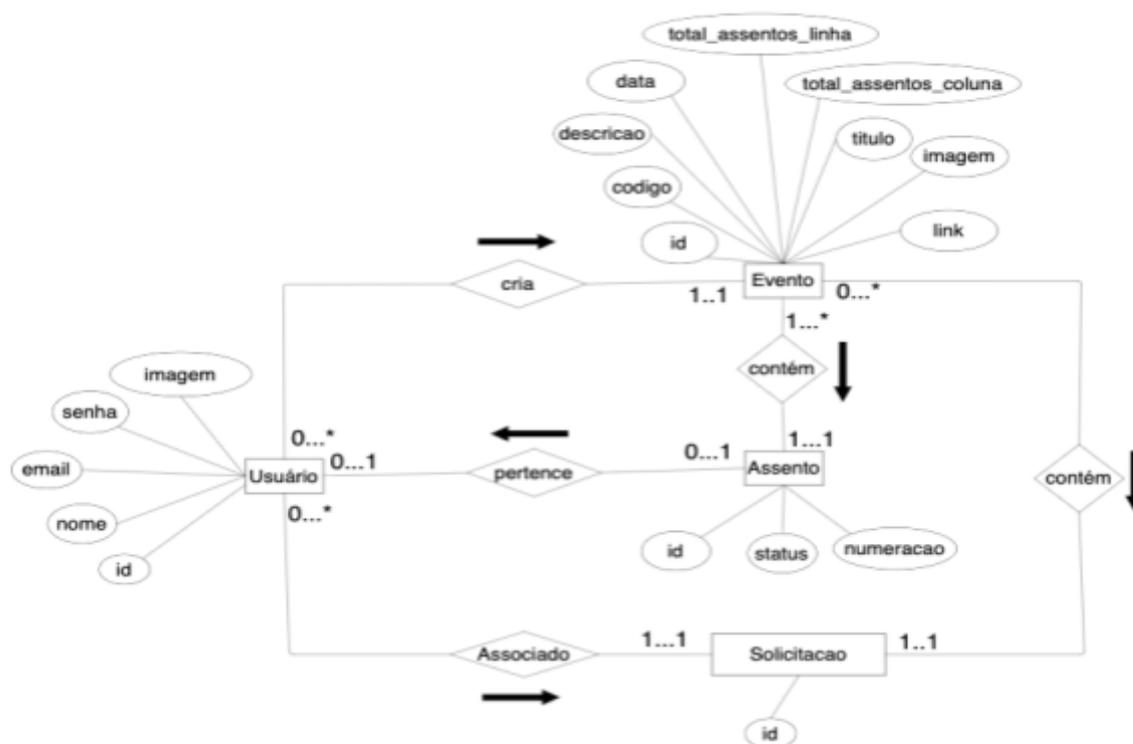


Figura 14 - Diagrama Entidade Relacionamento

#### 4.4.6 Visão de Deployment

A visão de *deployment* da aplicação é composta por três containers (Figura 15) sendo eles o dispositivo do usuário, o servidor principal onde está localizado toda a parte das operações e lógica de negócio e o servidor onde está o banco de dados. O dispositivo do usuário tem instalado o IOS, sistema operacional desenvolvido pela *Apple*, em sua versão a partir 15.0. Neste dispositivo do usuário será instalado a aplicação responsável por realizar a comunicação entre usuário e servidor principal que, por sua vez é composto por um ambiente de execução Nodejs o qual é construído na linguagem de programação *Javascript*. Além disso, encontra-se nele um framework já desenvolvido na própria linguagem chamado *Express.js*, que tem como função facilitar o desenvolvimento com funções e métodos já definidos. Por último temos o servidor onde está localizado o banco de dados relacional, que nesse caso será o MySQL, o qual armazenará os dados da aplicação como usuários e eventos.

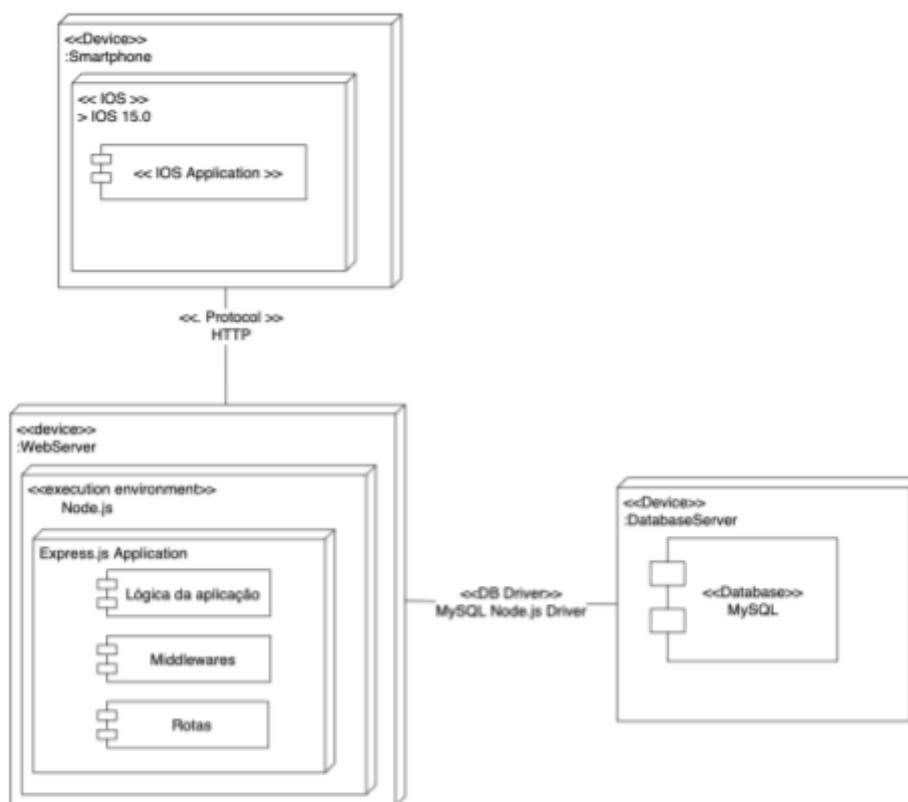


Figura 15 - Diagrama de Deployment da aplicação

#### 4.5 Interface com o Usuário

A interface do usuário foi inicialmente desenvolvida através da definição de um protótipo, o qual consiste em duas telas principais sendo elas: "Meus Eventos" (Figura 16a), "Perfil" (Figura 6b). Em Meus Eventos, o usuário poderá visualizar os eventos em que está participando, como também adicionar um novo evento a sua lista de eventos. Já em Perfil, o usuário poderá visualizar e alterar suas informações básicas.

Além disso, é disponibilizado a opção Adicionar evento (Figura 17), nesta o usuário poderá criar um novo evento a partir dos próprios dados dos eventos como descrição, data, local, ocupação do mesmo, entre outras informações acerca do próprio evento.

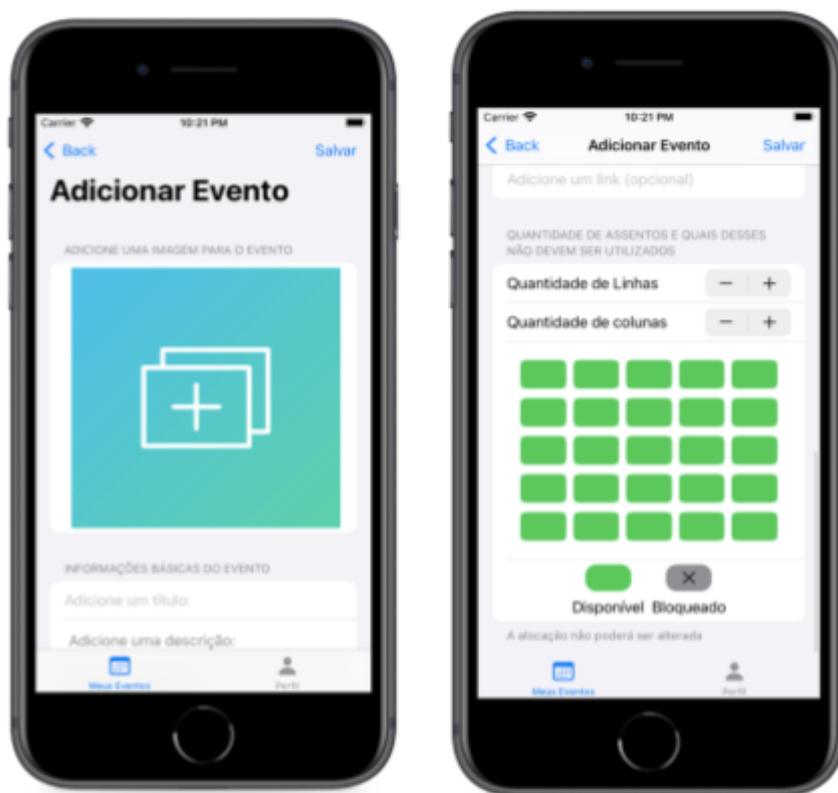


Figura 17 – (a) Tela Adicionar Evento (Parte 1), (b) Tela Adicionar Evento (Parte 2).

Já a Figura 18(a) apresenta as informações acerca do evento, como data, local e a ocupação atual, além disso, por meio dessa tela o usuário poderá solicitar sua presença. Além disso, a Figura 18(b) destaca o gerenciamento dos participantes. O usuário administrador do evento poderá gerenciar as solicitações pendentes como também os usuários participantes confirmados.

Por fim, a Figura 19 apresenta o Login, pela qual o usuário poderá se autenticar no sistema e então realizar as operações apresentadas anteriormente.



Figura 18 – (a) Tela do evento, (b) Tela de gerenciar participantes

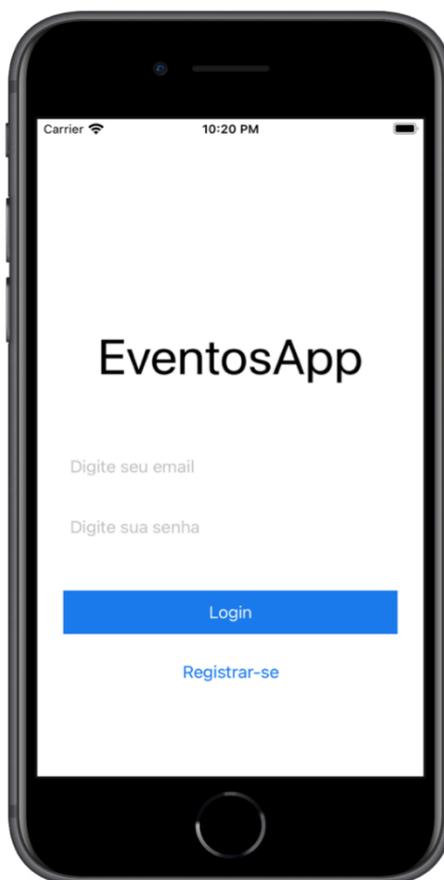


Figura 19 - Tela de Login

## 4.6 Tecnologias Utilizadas

### 4.6.1 SwiftUI

O SwiftUI (APPLE,2022) é uma biblioteca nativa para facilitar o desenvolvimento de interfaces de usuário para as plataformas da Apple. Diferentemente da maneira imperativa tradicional, com SwiftUI nós utilizamos uma abordagem declarativa para nosso código, no qual permite desenvolvermos uma interface baseado nos dados que são recebidos.

Além disso a biblioteca fornece uma rápida prototipação por meio das previews (APPLE,2022) , na qual temos quase instantaneamente uma visão de como a interface gráfica está projetada, sendo possível alterar diretamente da prévia, o qual automaticamente já insere o código na aplicação.

### 4.6.2 NodeJs

O NodeJs (NODEJS,2022) é um ambiente de execução Javascript no qual permite executar Javascript fora do navegador Web, assim podemos criar sistemas fáceis e escaláveis de maneira síncrona ou assíncrona utilizando a sintaxe padrão da linguagem. Em adição a isso, devido a sua grande popularidade, o desenvolvimento é facilitado devido a grande quantidade de livrarias auxiliares no qual podemos utilizar em nossos projetos que vão desde as coisas mais simples até as que envolvem questões de arquitetura

## 4.7 Validações

Para efetuar a validação da aplicação desenvolvida, foi realizado testes comportamentais e de exaustão. O primeiro teste consistia em testar o comportamento do servidor quando submetido a algumas chamadas, no qual permitiu ter uma noção do tempo de resposta, que em média ficou abaixo dos 5 segundos, que é um tempo aceitável para a infraestrutura que foi alocada. Além disso, foi colocado o aplicativo em rotinas de testes exploratórios a fim de verificar bugs que posteriormente foram corrigidos como por exemplo o layout de exibição e alguns bugs no que se diz a respeito de exibição de imagens, possibilitaram um melhor feedback para o desenvolvimento do aplicativo.

Como a plataforma escolhida foi o IOS, a Apple possui algumas restrições devido a testes públicos com usuários, sendo necessário uma assinatura ao programa de desenvolvedores da própria empresa, o que inviabilizou o processo de distribuição por causa do seu alto valor.

## 5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o projeto e desenvolvimento de uma aplicação destinada à usuários que queiram realizar eventos de forma organizada e prática. Esse objetivo é alcançado por meio de um aplicativo IOS no qual permite ao usuário diversas funcionalidades no que se diz a respeito de gerenciamento de eventos com o propósito de tornar tal processo simples e fácil quando comparado aos que já existem no mercado.

Para o futuro, seria interessante adicionar alguma funcionalidade de *QRCode* para evitar o processo de digitar o código do evento correspondente. Além disso, funcionalidades no quesito experiência de usuário como barra de pesquisa para buscar os eventos por nome ou data, além de poder tornar favorito um evento para obter rápido acesso, seriam mudanças que acrescentariam na usabilidade. Em conjunto, seria interessante o desenvolvimento de um aplicativo para a plataforma Android com o fim de expandir o público alvo.

Através da abordagem PBL, foi conduzido o desenvolvimento ágil do aplicativo como da parte do servidor no qual é responsável por armazenar e gerenciar esses dados. A condução desse projeto tratou de desafios multidisciplinares relacionados a qualidade, análise e arquitetura de Apps. De modo que todo o processo de construção na prática contribuiu para a aquisição de novas habilidades e skills, sejam elas técnicas, por exemplo o aprofundamento no desenvolvimento de aplicativos iOS, na questão de arquitetura e comunicação direta com servidor, como também pessoais, por exemplo proficiência na solução de problemas e planejamento de tarefas a serem executadas. Além desses foram reforçados conceitos vistos na graduação como estrutura de dados, programação web e banco de dados, no quais foram trabalhados de maneira em conjunto afim de realizar a melhor solução para cada problema.

## REFERÊNCIAS

- J. BOSCH, “Continuous software engineering: An introduction,” in *Continuous Software Engineering*. Springer, 2014, pp. 3–13.
- COCCO, S. (2006). *Student leadership development: The contribution of project-based learning*. Royal Roads University, Victoria, BC, Canada.
- COLUMBUS, L. (2016). 63% of Companies Operating in the Cloud can Develop an App in 3 Months or Less. *Forbes Tech Report*. em 30 de março de 2022. Disponível em: <https://www.forbes.com/sites/louiscolumbus/2016/02/24/63-of-companies-operating-in-the-cloud-can-develop-an-app-in-3-months-or-less/?sh=2255d1d56c15>.
- M. FITZGERALD, N. KRUSCHWITZ, D. BONNET, and M. WELCH, “Embracing digital technology: A new strategic imperative,” *MIT Sloan Management Review* in collaboration with Capgemini Consulting, 2013.
- FLORA, H. K., CHANDE, S. v, & WANG, X. (2014). Adopting an Agile Approach for the Development of Mobile Applications. In *International Journal of Computer Applications* (Vol. 94, Issue 17).
- GALLAUGHER, J., CHO, S., FITZGIBBON, J., & MAHECHA, J. A. (2017). The Flipped Classroom and Mobile App Development: A New Model for Engaging Students in a First Programming Course. In *Proceedings of the Twentythird Americas Conference on Information Systems*, Boston, Massachusetts.
- J. HUMBLE and D. FARLEY, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison Wesley, 2010.
- S. KHALAF, “Flurry analytics insights,” 2015. Acessado em 30 de março de 2022. Disponível em <http://www.flurry.com/blog/flurry-insights>.
- KOKOTSAKI D, MENZIES V, WIGGINS A. Project-based learning: A review of the literature. *Improving Schools*. 2016;19(3):267-277.

S. KRUSCHE and B. BRUEGGE, “User feedback in mobile development,” in Proceedings of the 2nd International Workshop on Mobile Development Lifecycle. ACM, 2014, pp. 25–26.

D. PAGANO and W. MAALEJ, “User feedback in the appstore: An empirical study,” in Proceedings of the 21st International Requirements Engineering Conference. IEEE, 2013, pp. 125–134.

D. PAGANO and B. BRUEGGE, “User involvement in software evolution practice: a case study,” in Proceedings of the 2013 international conference on Software engineering. IEEE, 2013, pp. 953–962.

R. PICHLER, Agile product management with scrum: Creating products that customers love. Addison-Wesley Professional, 2010.

STATISTA (2022). Statista - The Statistics Portal. Acessado em 30 de março de 2022. Disponível em: <https://www.statista.com/statistics/276623/number-of-appsavailable-in-leading-app-stores/>.

K. SCHWABER and M. BEEDLE, Agile software development with Scrum. Prentice Hall, 2002.

VALENTE, J., GANDRA, A. - Escolas retomam atividades e exigem respeito a protocolos sanitários. Acessado em 30 de março de 2022. Disponível em: <https://agenciabrasil.ebc.com.br/saude/noticia/2022-02/apesar-do-avanco-da-covid-19-estados-mantem-aulas-presenciais>

VERSION ONE, “8th annual state of agile survey,” 2014, <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>.

APPLE. **The SwiftUI Framework**. 2022. Disponível em: <<https://developer.apple.com/documentation/swiftui>>.

APPLE. **SwiftUI - Preview Provider**. 2022. Disponível em: <<https://developer.apple.com/documentation/swiftui/previewprovider/previews-swift.type.property>>.

NODE.JS. Disponível em: <<https://nodejs.org/>> . Acesso em: 07 abr. 2022.

SYMPLA. Disponível em: <<https://www.sympla.com.br/>>. Acesso em: 07 abr. 2022.

EVENTBRITE. Disponível em: <<https://www.eventbrite.com.br/>>. Acesso em: 07 abr. 2022.

## APÊNDICE A – Documentação do App para Gerenciamento de Eventos

### A.1 Descrição Técnica do App

Para a instalação e execução do App, a máquina deve possuir as versões mínimas dos softwares como descritas na Tabela A.2.

Tabela A.1 - Versões mínimas do software para execução

Software	Versão
iOs	15.0
MacOs	12.0
Xcode	13.0
NodeJs	14.0
Mysql	8.0

Com ambos os softwares instalados na máquina, criamos um arquivo chamado *ormconfig.json* dentro da pasta principal do servidor para permitir a conexão com banco de dados e nele colocamos as credenciais do nosso banco de dados como na Figura A.15, inserindo dados como nome do banco de dados, usuário e senha. Em seguida, executa-se o comando "npm install" para instalar as dependências necessárias para a execução do projeto, e depois o comando "npm start" para início do servidor.

```
// ormconfig.json
{
  "type": "mysql",
  "host": "localhost",
  "port": 3306,
  "username": "username",
  "password": "password",
  "database": "database",
  "entities": [
    "src/entidades/*.ts"
  ],
  "logging": false,
  "synchronize": true
}
```

Figura A.2 - Descrição do arquivo ormconfig.json a ser criado

Para a execução do app, como já mencionado na Tabela A.2, é necessária uma máquina com sistema operacional *MacOs* e o aplicativo *Xcode* instalado, visto que o App é para o *iOs*, será necessário também instalar o simulador juntamente com o *Xcode*.

Com o *Xcode* aberto, deve ser selecionado o modelo do aparelho utilizado, que deve ter no mínimo a versão 15 do sistema operacional, então pressionar a opção de executar, conforme a Figura A.16, posteriormente o App funcionará no simulador caso tenha sido instalado ou no aparelho físico, via conexão direta com o cabo. Para encerrar o aplicativo basta selecionar a opção encerrar no menu principal do *Xcode* e a aplicação será encerrada.

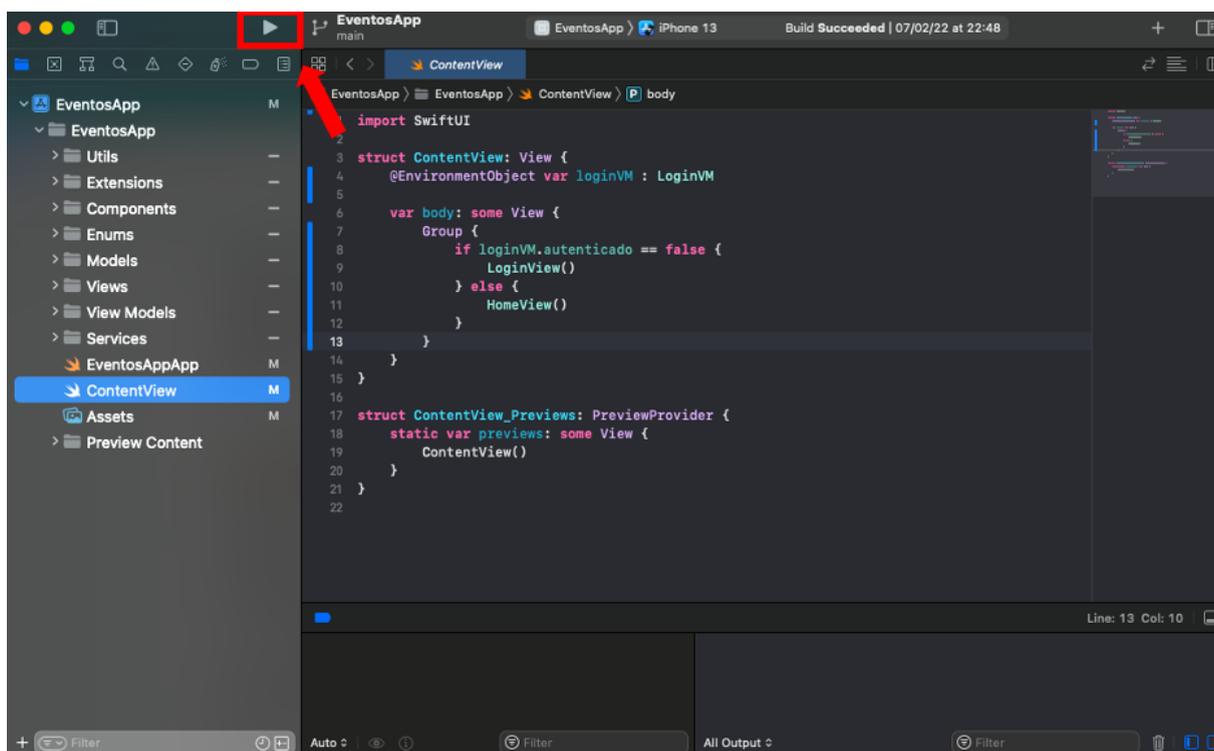


Figura A.3 - Executar o App.