

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

WILLIAN JOSE PIANTKOSKI

**PROGRESSIVE WEB APP PARA BUSCA DE LOCAIS PARA PRÁTICA  
DE FUTEBOL RECREATIVO**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO  
2021

WILLIAN JOSE PIANTKOSKI

**PROGRESSIVE WEB APP PARA BUSCA DE LOCAIS PARA PRÁTICA  
DE FUTEBOL RECREATIVO**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Campus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Vinicius Pegorini

PATO BRANCO  
2021



Ministério da Educação  
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
UTFPR - CAMPUS PATO BRANCO  
DIRETORIA GERAL - CÂMPUS PATO BRANCO  
DIR. DE GRAD. E EDUCAÇÃO PROFISSIONAL-PB  
DEP. ACADÊMICO DE INFORMÁTICA DAINF-PB



## TERMO DE APROVAÇÃO

### TRABALHO DE CONCLUSÃO DE CURSO

Progressive Web App para Busca de Locais para Prática de Futebol Recreativo

POR

William José Piantkoski

Este trabalho de conclusão de curso foi apresentado no dia 14 de maio, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Prof. Msc Vinicius Pegorini  
Professor orientador

Prof. Msc Andreia Scarlot Beulke  
Professora convidada

Prof. Dr. Robison Cris Brito  
Professor convidado

Assinam também:

Prof. Dr. Edilson Pontarolo  
Coordenador do Curso de Tecnologia em Análise e  
Desenvolvimento de Sistemas

Prof. Dra. Mariza Miola Dosciatti  
Responsável pela Atividade de Trabalho  
de Conclusão de Curso



Documento assinado eletronicamente por (Document electronically signed by) ANDREIA SCARIOT BEULKE, PROFESSOR ENS BASICO TECN TECNOLOGICO, em (at) 17/05/2021, às 12:23, conforme horário oficial de Brasília (according to official Brasília-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



Documento assinado eletronicamente por (Document electronically signed by) ROBISON CRIS BRITO, PROFESSOR ENS BASICO TECN TECNOLOGICO, em (at) 17/05/2021, às 18:55, conforme horário oficial de Brasília (according to official Brasília-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



Documento assinado eletronicamente por (Document electronically signed by) VINICIUS PEGORINI, PROFESSOR ENS BASICO TECN TECNOLOGICO, em (at) 17/05/2021, às 21:12, conforme horário oficial de Brasília (according to official Brasília-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



Documento assinado eletronicamente por (Document electronically signed by) MARIZA MIOLA DOSCIATTI, PROFESSOR ENS BASICO TECN TECNOLOGICO, em (at) 17/05/2021, às 23:12, conforme horário oficial de Brasília (according to official Brasília-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



Documento assinado eletronicamente por (Document electronically signed by) EDILSON PONTAROLO, COORDENADOR(A) DE CURSO/PROGRAMA, em (at) 18/05/2021, às 09:00, conforme horário oficial de Brasília (according to official Brasília-Brazil time), com fundamento no (with legal based on) art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



A autenticidade deste documento pode ser conferida no site (The authenticity of this document can be checked on the website) [https://sei.utfpr.edu.br/sei/controlador\\_externo.php?acao=documento\\_confir&id\\_orgao\\_acesso\\_externo=5](https://sei.utfpr.edu.br/sei/controlador_externo.php?acao=documento_confir&id_orgao_acesso_externo=5), informando o código verificador (informing the verification code) 2036768 e o código CRC (and the CRC code) 06C94DA3.

## **AGRADECIMENTOS**

Deixo aqui meus mais sinceros agradecimentos, sem citar nomes, para não cometer nenhuma injustiça, àqueles que, seja com uma simples carona, um café, uma ajuda qualquer em sala de aula e, inclusive, um pedido de ajuda, também às pessoas mais importantes da minha vida, minha família e minha namorada, pois, todos citados me condicionaram a alcançar este feito.

Deixo também um grande obrigado à Universidade Tecnológica Federal do Paraná, aos Professores e a todos os colaboradores que nesta Universidade trabalham para possibilitar a todos que elas frequentam o bem mais valioso de todos, o conhecimento.

## RESUMO

Neste projeto foi desenvolvida uma *Progressive Web App* junto do modelo de negócios *Online to Offline*, visando uma melhor experiência ao agendar partidas de futebol recreativo e encontrar pessoas para essa prática, utilizando a localização do usuário e tecnologias atuais do mercado de desenvolvimento web. A prática esportiva tem papel fundamental no bem estar humano trazendo melhoras, por exemplo, na saúde cardíaca, qualidade do sono e fortalecimento da musculatura. Todas as tecnologias utilizadas foram escolhidas com a intenção de melhorar a performance e tornar a aplicação mais acessível a todos os tipos de usuários. Como resultado na aplicação os usuários puderam encontrar espaços adequados para a prática do futebol recreativo e as empresas puderam gerenciar suas agendas e espaços de forma mais organizada e dinâmica.

**Palavras-chave:** Progressive Web App. JavaScript. Online to Offline. Geolocalização.

## **ABSTRACT**

In this project a Progressive Web App was developed using Online to Offline business model, aiming a better user's experience to schedule soccer matches and find people to play together, based on geolocation using the current web software development technologies. Sports practice is fundamental to human welfare improving health conditions such as a stronger heart, sleep quality and musculature strengthening. Every technology used was chosen to improve the performance and to make the application more accessible to all users. It is possible to conclude that users can find better places to play soccer and companies can manage schedule and places dynamically.

**Keywords:** Progressive Web App. JavaScript. Online to Off-line. Geolocation.

## LISTA DE FIGURAS

Figura 1 - Proxy reverso .....	23
Figura 2 - Terminal ssl.....	23
Figura 3 - Separação de conceitos cliente-servidor .....	25
Figura 4 - Representação de um servidor e cliente <i>stateless</i> com cache .....	25
Figura 5 - Arquitetura multicamadas .....	27
Figura 6 - Exemple de um product backlog .....	30
Figura 7 - Diagrama de um sprint.....	30
Figura 8 - Ciclo de teste tdd .....	32
Figura 9 - Caso de uso do usuário jogador .....	34
Figura 10 - Caso de uso do usuário do tipo empresa.....	34
Figura 11 - Caso de uso do usuário administrador .....	35
Figura 12 - Diagrama de atividades de um usuário jogador.....	36
Figura 13 - Diagrama de atividades do usuário empresa.....	37
Figura 14 - Diagrama de atividades do usuário administrador.....	38
Figura 15 - Autenticação de usuários .....	39
Figura 16 - Tela inicial do usuário administrador.....	40
Figura 17 - Formulário de pré-cadastro de empresas .....	41
Figura 18 - Formulário de cadastro de empresas.....	42
Figura 19 - Formulário de endereço da empresa .....	42
Figura 20 - Seleção da localização da empresa .....	43
Figura 21 - Agenda da empresa no primeiro acesso.....	44
Figura 22 - Menu de usuário empresa .....	44
Figura 23 - Lista de locais cadastrados.....	45
Figura 24 - Formulário de cadastro de locais .....	46
Figura 25 - Página de detalhes de um local cadastrado .....	46
Figura 26 - Agendamento de partidas. a) Formulário de agendamento. b) Lista de partidas agendadas.....	47
Figura 27 - Detalhes de uma partida agendada. a) Partida disponível para inscrições. b) Partida cancelada automaticamente pela falta de jogadores.....	48
Figura 28 - Formulário de informações básicas da empresa .....	49
Figura 29 - Formulário de endereço e localização da empresa .....	50

Figura 30 - Alteração de e-mail e senha .....	51
Figura 31 - Formulário de redefinição de senha.....	52
Figura 32 - Cadastro de usuário jogador. a) Formulário de cadastro. b) Cadastro com conta google .....	53
Figura 33 - Última etapa de criação do perfil de jogador.....	54
Figura 34 - Tela inicial do usuário jogador.....	55
Figura 35 - Listas de partidas disponíveis e locais. a) Lista de partidas disponíveis. b) Lista de locais.....	56
Figura 36 - Detalhes de uma partida. a) Aba de informações gerais. b) Aba de jogadores inscritos. c) Aba de localização .....	57
Figura 37 - Pagamento de inscrição na partida. a) Seleção do cartão de crédito. b) Cadastro de novo cartão .....	58
Figura 38 - Detalhes de um local.....	59
Figura 39 - Lista de avaliações de um local .....	60
Figura 40 - Localização de um estabelecimento .....	60
Figura 41 - Lista de partidas agendadas no local.....	61
Figura 42 - Formulário de agendamento de partidas .....	62
Figura 43 - Avaliação de uma partida .....	63
Figura 44 - Menu para usuários jogadores .....	64
Figura 45 - Histórico de partidas do usuário .....	64
Figura 46 - pagamentos de usuário. a) Lista de pagamentos realizados. b) Detalhes de um pagamento.....	65
Figura 47 - Cartões de crédito. a) Lista de cartões cadastrados. b) Formulário de cartão de crédito .....	66
Figura 48 - Relatório de avaliação de sites do <i>lighthouse</i> .....	67
Figura 49 - Instalação em dispositivo. a) Sugestão de instalação. b) Confirmação de instalação. c) Atalho de acesso à aplicação na lista de aplicativos do dispositivo .....	68
Figura 50 - Tela de abertura pwa .....	68
Figura 51 - Pedido de permissão de acesso a localização do dispositivo.....	69



## **LISTA DE QUADROS**

Quadro 1 - Materiais que serão utilizados no projeto.....	18
---	----

## LISTA DE ACRÔNIMOS

API	<i>Application Program Interface</i>
B2B	<i>Business to Business</i>
B2C	<i>Business to Consumer</i>
CEP	Código de Endereçamento Postal
CNPJ	Cadastro Nacional da Pessoa Jurídica
DNS	<i>Domain Name Service</i>
DOM	<i>Document Object Model</i>
ECMA	<i>European Computer Manufacturers Association</i>
HTML	<i>Hyper Text Markup Language</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IP	<i>Internet Protocol</i>
NPM	<i>Node Package Manager</i>
O2O	<i>Online to Offline</i>
PHP	<i>PHP: Hypertext Processor</i>
PWA	<i>Progressive Web App</i>
REST	<i>Representational State Transfer</i>
SPA	<i>Single Page Application</i>
SSL	<i>Secure Sockets Layer</i>
TDD	<i>Test Driven Development</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>12</b>
1.1 CONSIDERAÇÕES INICIAIS .....	12
1.2 OBJETIVOS .....	13
1.2.1 Objetivo Geral .....	13
1.2.2 Objetivos Específicos .....	13
1.4 ESTRUTURA DO TRABALHO .....	14
<b>2 REFERENCIAL TEÓRICO .....</b>	<b>15</b>
2.1 <i>Online-to-offline</i> .....	15
2.2 Progressive Web Application.....	16
<b>3 MATERIAIS E MÉTODO.....</b>	<b>19</b>
3.1 MATERIAIS .....	19
3.1.1 Javascript .....	19
3.1.2 Node.js .....	20
3.1.3 MongoDB .....	21
3.1.4 Nginx .....	22
3.1.5 Vue.JS .....	23
3.1.6 REST API .....	24
3.1.7 AWS Elastic Computing .....	27
3.1.8 AWS Route53 .....	28
3.1.9 Firebase Authentication .....	28
3.1.10 Firebase Hosting .....	28
3.2 MÉTODO .....	28
3.2.1 Scrum.....	28
3.2.2 Test driven development.....	31
<b>4 RESULTADOS.....</b>	<b>33</b>
4.1 ESCOPO DO SISTEMA .....	33
4.2 MODELAGEM DO SISTEMA .....	34
4.3 APRESENTAÇÃO DO SISTEMA .....	38
4.3.1 Autenticação .....	38
4.3.2 Usuário administrador .....	39
4.3.3 Usuário empresa .....	41

4.3.4 Usuário jogador .....	52
4.3.5 <i>Progressive Web App</i> .....	66
<b>5 CONCLUSÃO.....</b>	<b>70</b>
<b>REFERÊNCIAS .....</b>	<b>71</b>

## 1 INTRODUÇÃO

A evolução tecnológica trouxe diversos problemas de saúde pela diminuição do esforço em algumas atividades, porém, a tecnologia pode ser usada para a melhoria da saúde e aptidão física. Este capítulo apresenta uma introdução de uma solução que utiliza um modelo de negócio inovador e tecnologias que possibilitam acesso a qualquer pessoa que possua conexão com a Internet.

### 1.1 CONSIDERAÇÕES INICIAIS

Em pesquisa feita pelo Instituto Brasileiro de Geografia e Estatística (IBGE), no ano de 2015, com 356.904 (trezentos e cinquenta e seis mil e novecentas e quatro) pessoas, o esporte mais praticado no Brasil, por pessoas com idade a partir de 15 anos, é o futebol, correspondendo a 39,3% dos participantes. Glanner (2003) afirma que a revolução industrial contribuiu para a redução dos esforços físicos em atividades cotidianas, pois antes tais atividades exigiam 90% do físico humano e, atualmente as pessoas utilizam apenas 1% da capacidade física. Tal redução diminui a capacidade física humana, trazendo uma série de complicações para a saúde, até mesmo a morte.

A mesma evolução tecnológica que trouxe a perda da aptidão física pode ser usada para a busca de melhorias da saúde e aptidão física. Atletas de todos os níveis e de diversas modalidades fazem uso da tecnologia para o tratamento de lesões e análise de desempenho, melhorando suas estratégias (FILHO; SCHATTENBERG; STOLLMEIER, 2013).

O uso de dispositivos capazes de se conectarem com a Internet já se consolidou no cotidiano da população brasileira, principalmente os telefones móveis. De acordo com o IBGE (2016), das pessoas com idade a partir de 10 anos, que acessaram a Internet no mesmo ano, 94,6% fizeram a partir de um telefone móvel.

Neste trabalho, buscando usar a tecnologia e o uso de dispositivos capazes de se conectarem a Internet, bem como o modelo de negócios *Online-to-Offline* (O2O), que visa melhorar a experiência do consumo de serviços *offline* utilizando a Internet, para a melhoria da saúde e aptidão física humana, foi desenvolvida uma *Progressive Web App* (PWA). Essa aplicação permite que os usuários possam encontrar o melhor local e pessoas para a prática do futebol recreativo, de acordo com sua localização, bem como a diminuição da ociosidade dos estabelecimentos que oferecem local adequado para a prática do futebol.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Auxiliar na melhoria da saúde e aptidão física de praticantes de futebol recreativo, na busca por locais adequados e companhia para a prática do futebol, bem como, diminuir a ociosidade dos estabelecimentos que oferecem espaços para a prática do futebol recreativo.

### 1.2.2 Objetivos Específicos

- Facilitar ao usuário a busca pelo local mais próximo e com o melhor preço para a prática do futebol.
- Melhorar o gerenciamento dos horários dos locais que fornecem espaço adequado para a prática do futebol.
- Desenvolver uma aplicação que consuma poucos dados móveis.
- Desenvolver uma aplicação que possa ser utilizada com ou sem conexão com a Internet.

## 1.3 JUSTIFICATIVA

O avanço tecnológico trouxe diversos dispositivos, já consolidados na vida da população, que oferecem a capacidade de o ser humano executar mais tarefas, em qualquer lugar, ao longo do dia, muitas vezes impossibilitando o uso do tempo para a prática de atividades que tragam benefícios a saúde e aptidão física, como, por exemplo, o futebol.

Os mesmos dispositivos que oferecem maior capacidade de trabalho e uso do tempo, podem ser usados para o desenvolvimento de soluções que ajudem seus usuários a retomar a boa saúde e aptidão física, facilitando a busca e a prática de atividades que tragam tal benefício.

Neste trabalho foi desenvolvida uma PWA. Segundo Pontes (2018), uma PWA segue o paradigma de *Progressive Enhancement* (do Português, melhoria progressiva), o que condiciona uma aplicação a ser utilizável pelo maior número de pessoas possível, sob qualquer aspecto, sem nenhum tipo de restrição tecnológica.

Podemos colocar da seguinte forma: desenvolver com melhoria progressiva é um paradigma em que a aplicação deverá estar disponível para todos, sejam

usuários de microcomputadores ou *smartphones*, com *browsers* atualizados ou obsoletos, com conexão a Internet ou não (PONTES, 2018, p.1).

Também neste trabalho, o modelo de negócios O2O será aplicado. Este modelo está crescendo no mercado chinês, principalmente em compras realizadas em grupo, em que plataformas fornecem a comodidade e facilidade de compra de um comércio eletrônico por produtos e serviços consumidos *offline*, melhorando, assim, a experiência de compra do usuário.

As principais tecnologias utilizadas foram a linguagem de programação JavaScript, um servidor em Node.js e o banco de dados MongoDB.

#### 1.4 ESTRUTURA DO TRABALHO

Este trabalho foi organizado em diferentes capítulos. Neste primeiro capítulo foi apresentado o contexto do problema a ser resolvido, a justificativas e os objetivos. O capítulo seguinte foca no referencial teórico do projeto, descrevendo como um modelo de negócios inovador pode ser aplicado com tecnologias simples e eficientes. No Capítulo 3 estão descritas as ferramentas e tecnologias utilizadas na implementação do projeto. No Capítulo 4 são apresentados os resultados obtidos aplicando os conceitos e utilizando as ferramentas descritas nos capítulos anteriores. Finalizando com os últimos capítulos, em que são expostas as conclusões e referências utilizadas.

## 2 REFERENCIAL TEÓRICO

Neste capítulo está descrito o modelo de negócio do projeto, que vai balizar a implementação e como os usuários usufruirão da solução, bem como os conceitos de uma PWA, que foi o ponto central durante a modelagem do projeto, garantindo vasta acessibilidade ao seu uso.

### 2.1 *Online-to-offline*

A origem do modelo O2O vem dos Estados Unidos, ele foi criado por Alex Rampbell em 2010, que fez a junção do fluxo de informações e dinheiro *online* com o fluxo de comércio e serviços *offline* (TIANSHENG; JIONG, 2015).

Com o desenvolvimento de tecnologias para comércio eletrônico, os modelos de negócio tradicionais, *Business to Business* (B2B) e *Business to Consumer* (B2C), não são mais capazes de atender as necessidades de todas as áreas do comércio eletrônico. Principalmente movido pelo crescente uso das mídias sociais e a atualização constante das tecnologias voltadas para aplicativos desenvolvidos para plataforma móvel (ZAO; MENGYAO, 2014).

Por meio do uso do modelo O2O e dispositivos móveis atuais, é possível identificar quem são os usuários que acessam o aplicativo, suas localizações e outras características para estimar um número de consumidores em potencial (ZAO; MENGYAO, 2014).

O modelo O2O aplicado aos comércios eletrônicos combinam negócios *offline* com a Internet. As plataformas que usam o modelo, atraem usuários *online*, com a intenção de que seus produtos e serviços sejam consumidos *offline* (DU; TANG, 2014).

As compras em grupo elevaram o modelo O2O a um novo estágio. No mercado chinês, que está em constante crescimento, foi possível integrar serviços e comércio *offline* à dispositivos móveis, em um mercado com grande capacidade de crescimento, consolidando o modelo e desenvolvendo a indústria (TIANSHENG; JIONG, 2015).

A lógica aplicada no modelo O2O, é que usuários de plataformas *online* paguem antecipadamente por algo que desejam consumir *offline*. Para que um negócio se adeque ao modelo, é preciso que o serviço ou comércio oferecido tenha uma plataforma *online* e *offline*, no qual as duas plataformas trabalhem de tal forma que atenda às necessidades da Internet, com tudo centralizado no usuário (TIANSHENG; JIONG, 2015).



O modelo será aplicado seguindo a sua premissa de que, o negócio deve ser centralizado aos usuários, portanto, será desenvolvido uma PWA, que é capaz de trazer boa experiência de uso sem restrições tecnológicas, possibilitando o alcance à uma maior quantidade de usuários.

## 2.2 Progressive Web Application

De acordo com Ater (2017), PWA é um novo tipo de aplicação *web* que é capaz de combinar os benefícios de uma aplicação nativa com as características principais de uma aplicação *web*, tais como o uso multi-plataforma, baixa exigência por desempenho e desenvolvimento produtivo.

Nas primeiras interações do usuário, uma PWA se parece com uma simples aplicação *web* e vai oferecendo novos recursos progressivamente, de acordo com a frequência de uso feita pelo usuário, passando de uma simples aplicação *web* para algo muito mais parecido com uma aplicação nativa (ATER, 2017). A medida em que o usuário aumenta a frequência de uso da aplicação, novas funcionalidades são oferecidas, como a opção de criar um *link* de acesso a aplicação em sua tela inicial, bem como o recebimento de notificações.

As funcionalidades citadas acima, seguindo o paradigma *Progressive Enhancement* (do português, Melhoria Progressiva), condicionam a aplicação a atender o maior número de pessoas, independente de qual seja o dispositivo, tipo e velocidade de conexão ou ausência da mesma (PONTES, 2018).

Segundo Tal (2017), a instalação de aplicações nativas vem se tornando cada mais difícil de acontecer e mais custosa para convencer um usuário a instalar a aplicação. O fato de que, uma aplicação nativa, toma um lugar em definitivo no dispositivo ao qual for instalado e a quantidade de passos necessários para a instalação, torna o uso de uma PWA mais vantajoso, pois apenas um *link* de acesso é criado, economizando espaço de armazenamento do dispositivo.

Para Pontes (2018), para uma aplicação *web* ser reconhecida como uma PWA, ela precisa atender a alguns requisitos:

- *Service worker*: É um *script* que pode ser registrado para controlar páginas de um site, formando uma camada entre a página.

- E a *web*, que monitora e age de acordo com eventos originados das páginas que controla, respondendo às requisições de tais eventos, independentemente da existência de conexão com a rede (ATER, 2017).
- JavaScript desabilitado: Existem usuários que desabilitam o suporte à JavaScript de seus navegadores. Em uma PWA, mesmo que o usuário tenha desabilitado tal recurso, algum tipo de conteúdo deve ser exibido, mesmo que seja uma mensagem informativa, ao usuário, de que a aplicação está impossibilitada ao uso (PONTES, 2018).
- HTTPS: O *Hyper Text Transfer Protocol Secure* (HTTPS) é um protocolo de rede que criptografa os dados de uma requisição, garantindo mais segurança ao usuário. Uma PWA não deve funcionar sem o uso deste protocolo, caso um usuário tente acessar a mesma utilizando o protocolo *Hyper Text Transfer Protocol* (HTTP), o servidor da aplicação deve redirecionar o usuário para o canal HTTPS (PONTES, 2018).
- Carregar rapidamente: A aplicação deve estar disponível para uso em, no máximo, 10 segundos (PONTES, 2018).
- Possibilidade de instalação: De acordo com a quantidade de acessos de um usuário a uma PWA, a mesma é capaz de oferecer a possibilidade de criar um link de acesso rápido à aplicação na tela inicial do dispositivo do usuário (PONTES, 2018).
- Tela de abertura personalizada: Uma tela de abertura personalizada, com o logotipo de aplicação, deve ser exibida enquanto a aplicação completa sua inicialização (PONTES, 2018).
- Colorir o navegador: Para uma experiência semelhante ao de uma aplicação nativa, a PWA deve colorir a barra de endereço do navegador com uma cor semelhante ao restante da aplicação.
- Páginas que se adaptam: As páginas de uma PWA devem adaptar-se a diversos tamanhos de telas, para tal, se faz necessário o uso da *metatag* HTML (Hyper Text Markup Language) chamada *viewport*, que possibilita a aplicação ajustar-se de acordo com o tamanho da área disponível na tela do dispositivo (PONTES, 2018).
- Compatibilidade: Com as diversas opções de navegadores disponíveis para o usuário, uma PWA deve ser *cross-browser*, ou seja, ser capaz de funcionar em diversos navegadores diferentes (PONTES, 2018).

- Transições de telas: No momento em que se faz necessário a troca de telas na aplicação, tal ação deve ser completada de forma transparente, ou seja, sem que o nenhum tipo de carregamento de faça o usuário esperar. Para tal, faz-se necessário o uso de uma PWA *Single Page Application* (SPA), em que todos os recursos da aplicação são carregados no primeiro acesso, evitando que novos carregamentos ocorram (PONTES, 2018).
- URL única: Cada página de uma PWA deve conter uma *Uniform Resource Locator* (URL) única, o que possibilita ao usuário a inclusão de determinada página à lista de páginas favoritas em seu navegador, agilizando acessos futuros à tal página (PONTES, 2018).

### 3 MATERIAIS E MÉTODO

Nesse capítulo serão apresentados os materiais e o método utilizado no desenvolvimento deste projeto.

#### 3.1 MATERIAIS

No Quadro 1 estão listados os materiais que foram utilizados durante o desenvolvimento da aplicação.

**Quadro 1 – Materiais que foram utilizados no projeto**

MATERIAL	VERSÃO	FINALIDADE
Javascript	ES6	Linguagem de programação
Node.js	14	Framework back-end
MongoDB	4	Persistência de dados
Nginx	1.18	Proxy reverso
VueJS	2	Framework Front-end
REST API		Metodologia de desenvolvimento
Adobe XD Starter		Design
AWS EC2		Servidor cloud
Firebase Authentication		Autenticação
Firebase Hosting		Hospedagem
AWS Route53		DNS

**Fonte: Autoria própria.**

##### 3.1.1 Javascript

Javascript foi criado pela Netscape em 1995, levando a linguagem em 1996 e 1997 até a *European Computer Manufacturers Association* (ECMA) para criação de especificações e padrões (LECHETA, 2018).

A ascensão da Internet provocou o crescimento e grande adoção da linguagem, inicialmente utilizada para a manipulação de elementos de páginas *web* (STEFANOV, 2013).

Javascript é uma linguagem extremamente produtiva, dinâmica, não tipada (não precisa declarar os tipos de variáveis) e interpretada, além de possuir suporte a Orientação a Objetos (LECHETA, 2018, p.14).

Chegou a ser considerada por muitos desenvolvedores uma linguagem incompleta, a linguagem evoluiu a ponto de outras linguagens de programação, como Java e *PHP: Hypertext Processor* (PHP) a adotarem funcionalidades, como *closures* e funções anônimas, que já eram utilizadas há muito tempo em JavaScript (STEFANOV, 2013).

Atualmente, ainda em supervisão da ECMA, está na versão 6, recebendo nome oficial de ECMAScript 6. Uma das melhorias que chamou mais a atenção, foi a adoção do motor Javascript V8 do Chrome, que possibilitou que a linguagem deixasse de ser apenas interpretada e passasse a ser compilada (LECHETA, 2018).

A constante melhoria da linguagem permitiu que novas ferramentas fossem criadas, possibilitando seu uso na construção de soluções para diversas plataformas, como *desktop*, aplicações móveis híbridas e servidores, tornando-se a linguagem mais popular do mundo e uma das mais proeminentes na Internet (GRONER, 2017).

### 3.1.2 Node.js

Node.js é baseado no motor Javascript V8 do Chrome e utiliza JavaScript como linguagem de programação. Vem sendo adotado no mundo todo por sua facilidade para desenvolver soluções e por ter boa performance (LECHETA, 2018).

Node.js é o responsável por trazer a linguagem JavaScript ao servidor, podendo ser utilizado nas mais diversas e complexas soluções, como manipulação de arquivos, execução de linhas de comando, bem como o desenvolvimento em plataformas de *hardware* para soluções robóticas e Internet das coisas (WILSON, 2018).

Node.js usa um modelo de E/S (entrada e saída ou I/O, em inglês) não bloqueante, é orientado a eventos (*Event Driven*) e extremamente leve, rápido e eficiente (LECHETA, 2018, p.14).

Diferente de outras linguagens de programações, Node.js faz uso de uma arquitetura não bloqueante, ou seja, assíncrona, permitindo que novas requisições sejam processadas sem que se faça necessário aguardar o término de processamento de outras (LECHETA, 2018).

Em uma arquitetura bloqueante, também chamada síncrona, para cada nova requisição, uma nova *thread* é criada. Em aplicações que recebem um grande número de requisições, este

tipo de arquitetura pode se tornar um problema caso o limite de *threads* do servidor seja utilizado, pois tal condição, fará com que novas requisições só possam ser processadas quando alguma *thread* já usada por outra requisição termine seu processamento, causando um gargalo no servidor (LECHETA, 2018).

A arquitetura do Node.js faz uso de uma *thread* principal chamada *Event Loop* (do português, loop de eventos) que fica monitorando eventos disparados por requisições, ao receber tais eventos, o *Event Loop* enfileira os mesmos e delega seu processamento à *threads* secundárias, chamadas de *workers* (do português, trabalhadores), as quais, após concluir o processamento, retornam uma resposta com o resultado do mesmo (LECHETA, 2018).

### 3.1.3 MongoDB

O MongoDB é um banco de dados *Open Source* (do português, código aberto), não relacional e orientado a documentos, que faz uso da linguagem de programação JavaScript para manipular seus objetos e é totalmente compatível com o Node.js (LECHETA, 2018), podendo ser utilizado para soluções de qualquer propósito (CHODOROW, 2013).

Tornou-se um banco de dados poderoso devido a sua boa performance por meio da combinação de sua flexibilidade e recursos como índices secundários, *range queries*, classificação, agregações e índices geoespaciais (CHODOROW, 2013).

Por ser orientado a documentos, ao invés das tradicionais linhas de tabelas dos bancos de dados relacionais, possui documentos, permitindo o uso de hierarquias complexas em apenas um único registro (CHODOROW, 2013).

No MongoDB, não se faz necessário o uso de qualquer tipo de modelagem ou predefinição de tipos e/ou tamanho de dados, tornando seu uso fácil e produtivo (CHODOROW, 2013). Apesar de tal possibilidade, ainda é possível que se faça uma espécie de modelagem dos dados de um documento do MongoDB, por meio de uma aplicação Node.js, utilizando um pacote do *Node Package Manager* (NPM) chamado Mongoose.

Com a melhora das conexões de rede e o baixo custo para armazenar grandes quantidades de dados, criou-se um ambiente perfeito para que aplicações utilizem cada vez mais dados, tornando necessário o uso de tecnologias com capacidade de escalar rapidamente. O MongoDB foi projetado para escalar de forma horizontal, ou seja, com vários computadores pequenos trabalhando em conjunto, formando um *cluster*. O uso de um *cluster* costuma gerar um maior trabalho no que se diz respeito a manutenção. Porém, o MongoDB é capaz de lidar com a distribuição de seus dados e redirecionamento de requisições por dados de forma

automática, desta forma, quando um *cluster* não possuir mais capacidade para operar e necessitar escalar, basta que novas máquinas sejam adicionadas ao mesmo e o MongoDB tomará conta do resto (CHODOROW, 2013).

Para atingir bons índices de performance, o MongoDB usa o máximo de memória RAM possível como *cache* e ao escolher o índice correto na execução de *queries* (CHODOROW, 2013).

Por ser projetado a deixar lógicas de negócio a serem tratadas do lado da aplicação, o MongoDB é capaz de atingir uma excelente performance (CHODOROW, 2013).

### 3.1.4 Nginx

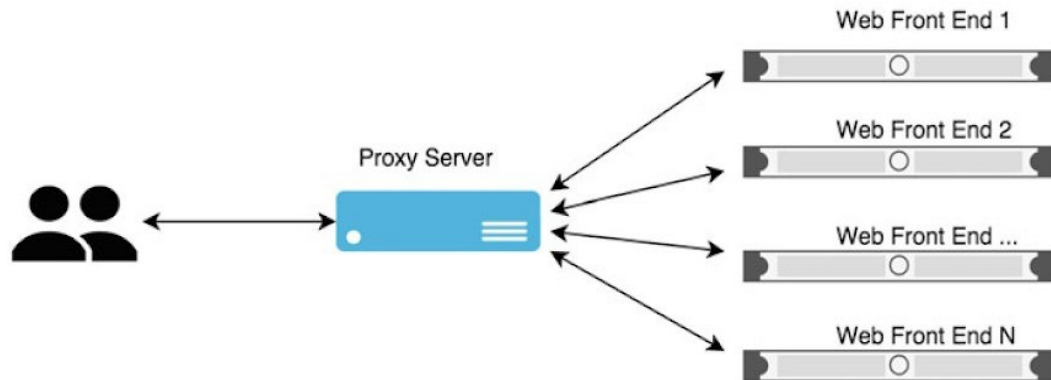
O Nginx, antes de ser considerado um simples servidor *web*, pode ser considerado um *event based reverse proxy server* (do português, servidor de *proxy* reverso baseado em eventos) e que resolveu problemas de performance relacionado a grandes cargas de requisições concorrentes devido a sua arquitetura assíncrona (SONI, 2016).

Com o tempo de espera para o carregamento de uma página se tornando cada vez mais importante, sendo esta medida utilizada, inclusive, para o ranqueamento de páginas em mecanismos de busca, servidores *web* passaram a enfrentar dificuldades para atender este requisito, devido a sua arquitetura bloqueante, que faz uso de múltiplas *threads*, causando um bloqueio ao atingir o número total de *threads* presentes no *hardware* em que o serviço se encontra. O Nginx resolveu o problema de performance na *web*, apresentando bom resultado, mesmo sob alta demanda (SONI, 2016).

Muito da performance oferecida pelo Nginx vem da sua arquitetura assíncrona e orientada a eventos, assim como o Node.js, não cria uma nova *thread* para cada requisição, delega o processamento de novas requisições a outras *threads*, tornando-se não bloqueante. Logo, foi considerado capaz o problema de trabalhar com grandes quantidades de requisições simultâneas, frequentemente chamado de *10.000 concurrent connections* (C10K) (do português, 10.000 conexões simultâneas) (SONI, 2016).

Um *proxy reverso* é um tipo de servidor de *proxy* que trabalha a frente de um servidor *web* comum, decidindo como proceder de acordo com a *Uniform Resource Identifier* (URI) (SONI, 2016), como o exemplo dado na Figura 1.

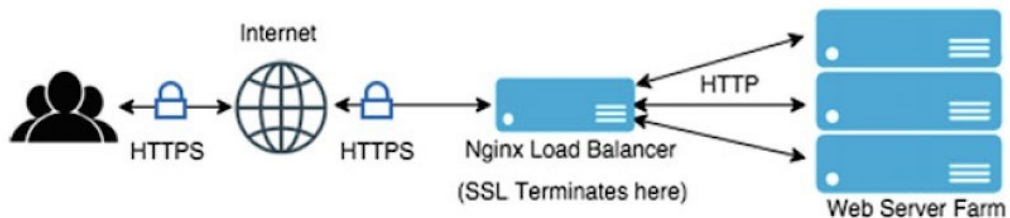
**Figura 1 - Proxy reverso**



Fonte: Soni (2016).

O Nginx também fornece recursos relacionados à segurança, atuando como um terminal *Secure Sockets Layer* (SSL), permitindo que requisições com conteúdo criptografado sejam recebidas. Esta camada adicional permite que o Nginx faça a descriptografia do conteúdo, como apresentado no diagrama da Figura 2, entregando tal conteúdo pronto para os servidores *web*, possibilitando que trabalhem com uma carga menor de processamento (SONI, 2016).

**Figura 2 - Terminal SSL**



Fonte: Soni (2016).

Neste projeto, o Nginx foi utilizado como um servidor de *proxy* reverso, redirecionando requisições HTTP para HTTPS, bem como requisições HTTPS para a camada SSL, entregando o conteúdo da requisição ao servidor Node.js.

### 3.1.5 Vue.JS

Vue.JS é uma biblioteca JavaScript que possibilita o desenvolvimento de componentes reativos na web, podendo ser utilizado em qualquer projeto que possui *front-end* (INCAU, 2017).



Os componentes reativos são pequenos fragmentos de páginas *web* que possuem marcação, estilo e JavaScript próprio. Para que um componente possa considerado reativo, sua marcação deve reagir de acordo com a mudança de informações presentes (INCAU, 2017).

É muito flexível, pois usa JavaScript puro e não limita o desenvolvedor a seguir um único padrão, bem como possui uma ótima performance, por fazer uso do chamado *Virtual DOM* (INCAU, 2017).

Evan You foi o criador desta biblioteca. Enquanto trabalhava no Google Creative Labs, sentiu a necessidade de uma ferramenta que permitisse construir seus protótipos de forma leve e flexível. Tão logo, percebeu que tal ferramenta não existia no mercado, o que deu origem ao Vue.JS, o qual cresceu de tal maneira, que, além de protótipos, pode ser utilizado em aplicações de todos os tamanhos (INCAU, 2017).

### 3.1.6 REST API

O termo REST foi criado no ano de 2000 pelo americano Roy Fielding, este que foi um dos principais autores do protocolo HTTP e coautor de um dos servidores *web* mais conhecidos, o Apache (DOGLIO, 2015).

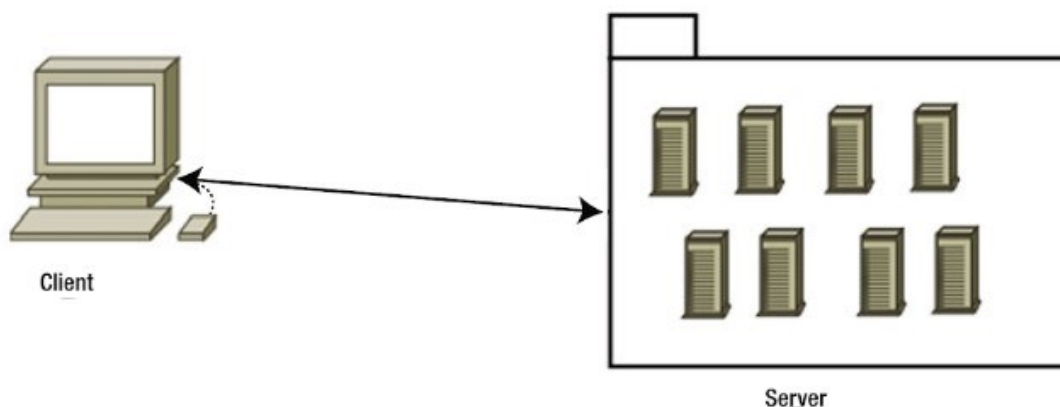
Segundo Doglio (2015), *Representational State Transfer* (REST) é um estilo de arquitetura definido para criar e organizar sistemas distribuídos, ou seja, não é um manual ou um padrão rígido com regras. A ideia principal deste estilo, é que um sistema distribuído organizado de acordo com o REST possa melhorar nos seguintes aspectos, de acordo com DOGLIO (2015):

- Performance: o meio de comunicação proposto pelo REST é eficiente e simples.
- Escalabilidade na interação de componentes: qualquer sistema distribuído deve ser capaz de lidar com este aspecto, o padrão REST permite isso.
- Simplicidade da interface: uma interface simples permite uma comunicação simples.
- Modificação de componentes: a separação de conceitos, *front-end* e *back-end*, proposto pelo REST, permite a modificação de componentes de forma independente com o risco e custo mínimo.
- Portabilidade: REST não depende de nenhuma tecnologia ou linguagem de programação específica, podendo ser aplicado e consumido por qualquer sistema.

- Confiabilidade: por ser *stateless*, permite a recuperação rápida quando falhas acontecem.
- Visibilidade: novamente, por ser *stateless*, permite que sistemas de monitoramento não precisem mais do que uma requisição para determinar o estado desta requisição.

Ao construir um serviço REST, o primeiro ponto a ser aplicado em REST é a separação de conceitos, em que cliente e servidor trabalham separados. O servidor é o responsável por manipular um conjunto de serviços com base nas requisições feitas pelo cliente, retornando o resultado desta manipulação (DOGLIO, 2015). A Figura 3 mostra um exemplo desta separação.

**Figura 3 - Separação de conceitos cliente-servidor**

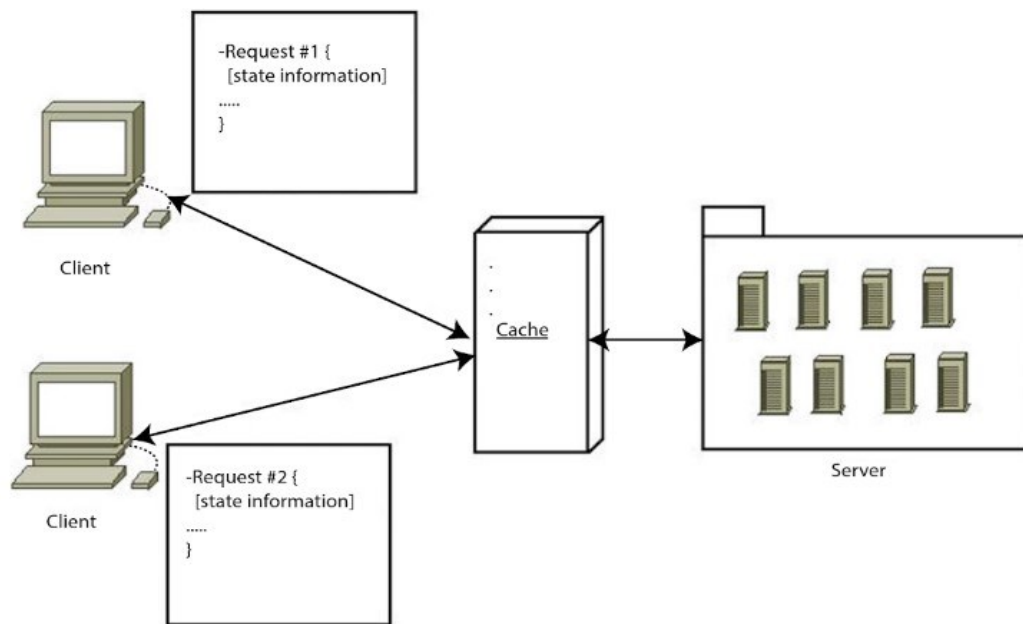


Fonte: Doglio (2015).

A comunicação entre cliente e servidor deve ser *stateless* (do português, ausência de estado), ou seja, toda requisição feita pelo cliente deve conter todas as informações necessárias pelo servidor, para que possa atender a requisição sem a necessidade de dados armazenados (DOGLIO, 2015).

O padrão REST também propõe que toda resposta à uma requisição deve ser cacheada, ou seja, seus dados devem ser armazenados, como mostra a Figura 4. Esse aspecto evita que requisições repetidas não precisem ser processadas pelo servidor, bem como, diminui a quantidade de requisições necessárias pelo cliente (DOGLIO, 2015).

**Figura 4 - Representação de um servidor e cliente *stateless* com cache**

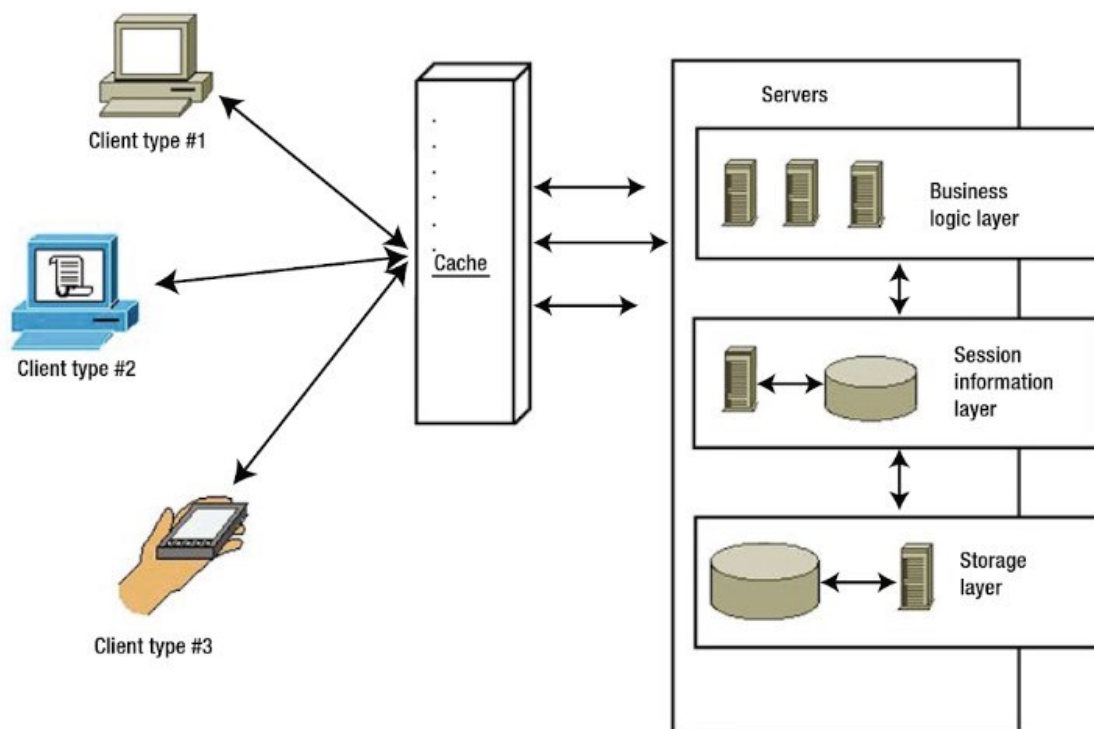


Fonte: Doglio (2015).

Uma interface uniforme se faz necessária no REST, simplificando as interações do cliente com a *Application Programming Interface* (API), possibilitando uma implementação independente e entregando um conjunto de regras claro ao cliente, ou seja, clientes de diversos tipos poderão interagir com facilidade com o serviço REST (DOGLIO, 2015).

REST foi desenvolvido para a Internet, por isso, espera-se que um serviço REST seja capaz de trabalhar com um grande tráfego de requisições. Para alcançar este objetivo, deve-se separar os componentes em camadas de forma que uma camada use somente outra camada abaixo e responda somente outra camada acima, simplificando a complexidade geral do serviço, beneficiando serviços que estão sempre crescendo (DOGLIO, 2015). A Figura 5 traz um diagrama de como as camadas são organizadas.

Figura 5 - Arquitetura multicamadas



Fonte: Doglio (2015).

A API provê um meio de conexão, entre duas partes, para o acesso a dados e serviços. No mercado de desenvolvimento, uma API é interpretada como se fosse um contrato, permitindo que desenvolvedores a usem com confiança. Uma API também faz a conexão entre um consumidor e um provedor de dados com muito mais eficiência, uma vez que suas interfaces são documentadas, consistentes e previsíveis (JACOBSON; BRAIL; WOODS, 2012).

Uma REST API é um servidor *web* com um propósito específico que atende as necessidades de uma página *web* ou uma aplicação. O cliente usa uma API para se comunicar com o serviço *web*, ou seja, uma API expõe um conjunto de dados e funções que facilitam a interação entre aplicações, permitindo a troca de informações (MASSÉ, 2011).

### 3.1.7 AWS Elastic Computing

O *Amazon Web Services (AWS) Elastic Computing (EC2)* é um serviço de virtualização de servidores em *Cloud*.

A utilização deste serviço possibilita a eliminação de hardware físico, aumentando a agilidade de implementação de soluções tecnológicas que dependem de servidores para sua execução e tornando-as mais seguras.

### 3.1.8 AWS Route53

O AWS *Route53* é um serviço em *Cloud* para o uso de *Domain Name Service* (DNS) com alta disponibilidade e capacidade de escala.

Foi projetado para facilitar o direcionamento de usuários de aplicativos, convertendo domínios para seus respectivos endereços de *Internet Protocol* (IP). Sua principal função é conectar solicitações de usuários aos mais diversos serviços AWS aos seus respectivos endereços de IP, inclusive de serviços que não utilizam a plataforma AWS, facilitando o gerenciamento de tráfego.

### 3.1.9 Firebase Authentication

É um serviço disponibilizado de forma gratuita para reconhecimento de identidade de usuários. Fornece bibliotecas tanto para uso no *back-end* como no *front-end*.

Sua utilização permite que usuários possam fazer *login* em aplicações, integrando vários métodos de *login*, como Google, Facebook, Twitter e diversos outros provedores.

### 3.1.10 Firebase Hosting

O Firebase Hosting é considerado um serviço de hospedagem de aplicações, conteúdo e micros serviços.

Disponibiliza ferramentas de fácil uso aos seus usuários, capacitando-os a disponibilizar suas soluções na *web* com comandos simples, conexão segura e alta disponibilidade e performance.

## 3.2 MÉTODO

Neste projeto foi aplicado o método Scrum para planejamento de tarefas e prazos em conjunto do método de desenvolvimento *Test Driven Development* (TDD).

### 3.2.1 Scrum

O Scrum existe desde a década de 1990, porém, só ficou mundialmente conhecido no ano 2000, se tornando na forma mais popular de se trabalhar no desenvolvimento de software (SABBAGH, 2013).

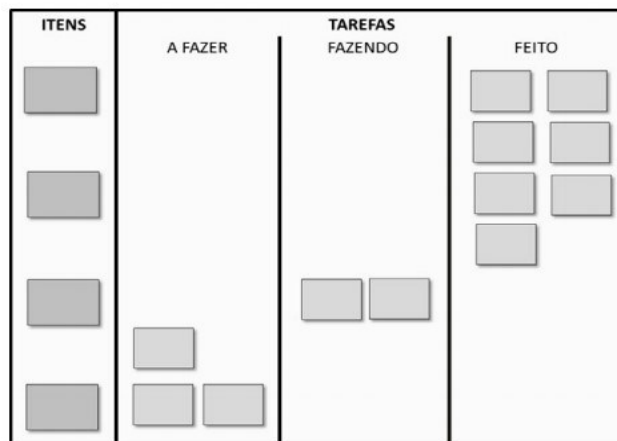
Vem sendo aplicado por empresas de diversos tamanhos, desde multinacionais à *startups* desconhecidas, pois, se aplicado de forma correta, permite entregar valor aos clientes desde o início do projeto, lidar com mudanças inesperadas no projeto, reduzir riscos de insucesso, aumentar a qualidade dos produtos e a produtividade da equipe (SABBAGH, 2013).

- Entrega de valor contínua: Scrum permite, desde o início do projeto e com alta frequência, entregar partes funcionais do sistema, permitindo que o produto seja usado e disponibilizado rapidamente no mercado, gerando retorno ao investimento dos clientes.
- Riscos: os riscos são reduzidos com a produção do produto em ciclos curtos e as entregas frequentes, partindo das partes mais importantes do sistema às menos importantes, bem como a colaboração e envolvimento dos clientes com o projeto, deixando *feedbacks* sobre as partes entregues.
- Qualidade: o time que trabalhar com Scrum é o responsável por garantir a qualidade do produto, pois possui habilidades e técnicas para aplicar as validações necessárias, como por exemplo, testes automatizados, entregando cada parte do produto funcionando corretamente.
- Mudanças: as necessidades do negócio e especificação não são todas definidas no início do projeto, são descobertas e avaliadas no decorrer do desenvolvimento, garantindo que o produto entregue somente aquilo que tem valor real ao cliente.
- Progresso do projeto: o Scrum garante transparência aos clientes e envolvidos, garantindo que os mesmos possam visualizar o progresso, no mínimo, ao final de cada ciclo de desenvolvimento.
- Desperdícios: o time que usa Scrum deve simplificar o desenvolvimento e utilizar somente aquilo que é realmente necessário para o projeto, causando uma redução nos desperdícios do projeto.
- Produtividade: o Scrum possui fatores que melhoram a produtividade do time, sendo alguns deles o trabalho em equipe e a autonomia na realização dos trabalhos, a remoção de impedimentos, melhoria contínua dos processos de trabalho e um ritmo sustentável de trabalho.

Dentre as práticas propostas pelo Scrum, neste projeto foram utilizadas a prática do *Product Backlog* na organização de prioridade das tarefas e *Sprints* para a definição de prazos e entregas.

No *Product Backlog*, exibido na Figura 6, são definidas quais as atividades são necessárias para atender ao escopo do projeto. Estas atividades são organizadas de acordo com sua importância, ou seja, quanto maior a importância de uma atividade, mais alta será sua posição e maior será a quantidade de detalhes. A importância das atividades também define a ordem para implementação das mesmas (SABBAGH, 2013).

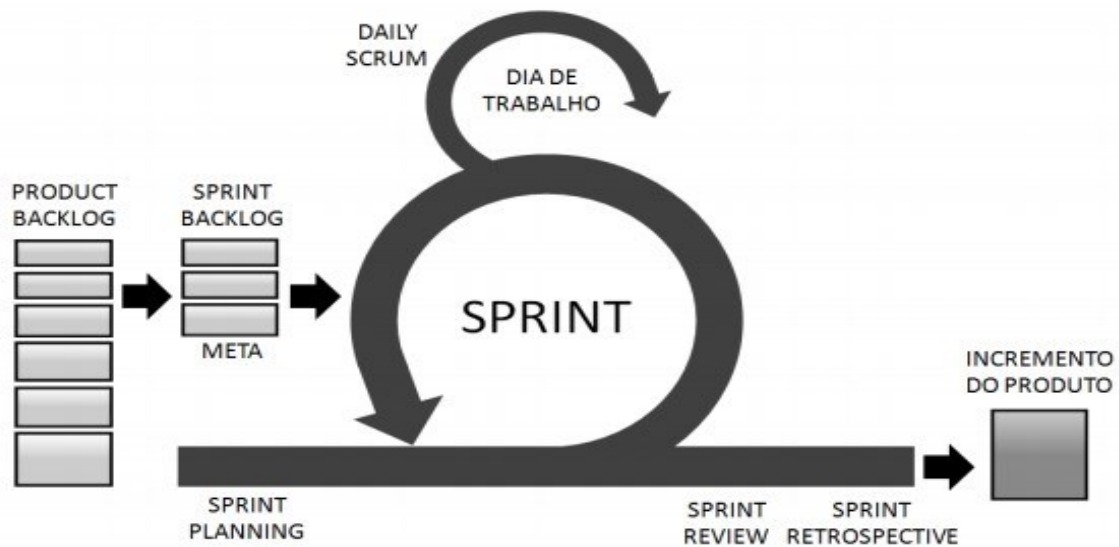
**Figura 6 - Exemple de um Product Backlog**



**Fonte: Sabbagh (2013).**

Quanto aos *Sprints*, são ciclos de desenvolvimento em que o time, com o apoio do *Scrum Master*, se reúne para definir qual atividade do *Product Backlog* será implementada, bem como a meta do *Sprint*. Ao término de um *Sprint*, uma parte do produto deve ser entregue, parte esta que deve ser totalmente funcional, entregando valor ao cliente continuamente (SABBAGH, 2013). Na Figura 7 é apresentado um exemplo do diagrama do ciclo de um *Sprint*.

**Figura 7 - Diagrama de um Sprint**



Fonte: Sabbagh (2013).

### 3.2.2 Test driven development

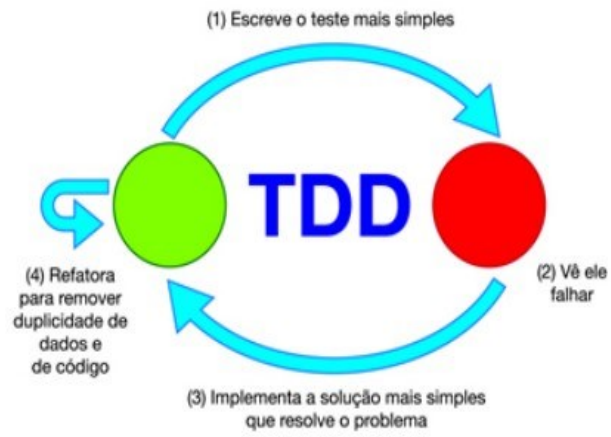
*Test Drive Development* (TDD) é uma das práticas de desenvolvimento de software recomendada por metodologias ágeis. O TDD tem como objetivo a aplicação de testes automatizados no decorrer de todo o desenvolvimento. A principal diferença, se comparado a outras práticas, é os testes são escritos antes da implementação (ANICHE, 2012).

De acordo com Aniche (2012), esta simples inversão no ciclo de desenvolvimento permite que uma maior quantidade de testes seja aplicada, conseqüentemente, uma maior abrangência dos mesmos, além disso, forçará o desenvolvedor a escrever códigos de maior qualidade.

O TDD, neste projeto, será aplicado por meio de testes de unidade, ou seja, as funções que serão implementadas serão testadas de forma individual, bem como testes de acoplamento, que testará funções trabalhando em conjunto. Todos os testes seguirão o ciclo definido pelo TDD, mostrado na Figura 8.



**Figura 8 - Ciclo de teste TDD**



**Fonte: Aniche (2013).**

## 4 RESULTADOS

Neste capítulo serão apresentados o escopo deste projeto, a modelagem dos dados, imagens e a implementação.

### 4.1 ESCOPO DO SISTEMA

A aplicação desenvolvida fornece aos usuários as opções de cadastro via conta do Google ou e-mail. Ao abrir a aplicação pela primeira vez, o usuário deverá ser questionado se deseja instalá-la. Um novo usuário que decidir entrar no sistema utilizando seu e-mail pessoal, deverá preencher um formulário e com seu *e-mail*, nome completo, senha e se deseja ser do tipo jogador ou empresa. Ao completar o registro inicial, o usuário do tipo jogador deve preencher um segundo formulário contendo seus dados de jogador, como a posição em que joga e seu estilo de jogo. Ao concluir o cadastro, esse usuário será redirecionado para a tela principal da aplicação, na qual encontrará informações como o total de partidas em andamento, em aberto e de estabelecimentos cadastrados.

Ao acessar uma das listas pela primeira vez, o usuário deve fornecer permissão de acesso à sua localização. A lista de locais e/ou partidas deve ser ordenada de acordo com a distância referente a posição atual do usuário, de forma crescente.

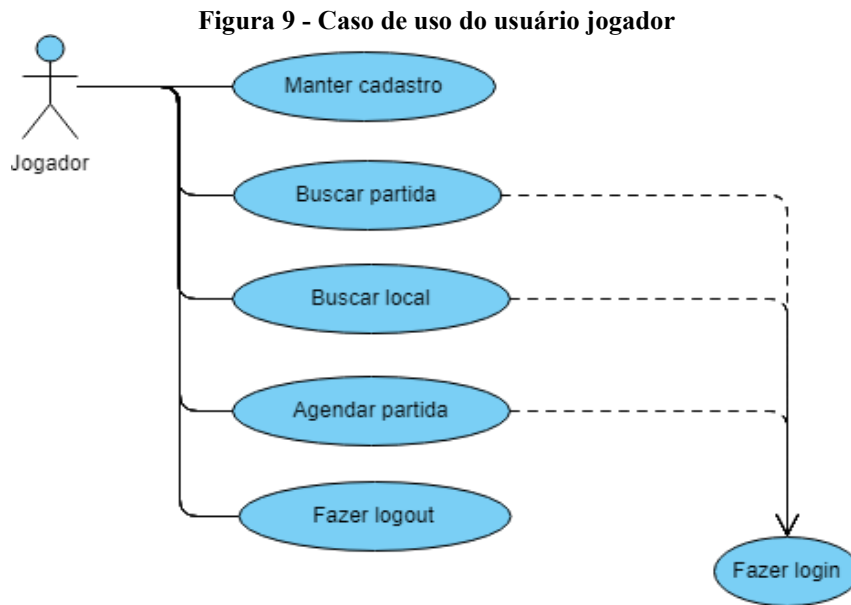
Quando acessar a página de detalhes de uma partida, o usuário terá acesso a informações do horário e local da partida, pessoas inscritas, bem como um mapa para ver o caminho. Para realizar a inscrição em uma partida, caso seja paga, o usuário deve pagar somente com cartão de crédito. Nos casos em que o usuário cancelar sua inscrição de uma partida, perderá pontos de reputação. Um usuário não poderá fazer inscrições simultâneas em mais de uma partida. Após o término de uma partida, o usuário não poderá participar de outra partida sem ter preenchido o formulário de avaliação da partida anterior que participou.

Quando acessar a página de detalhes de um local, o usuário terá acesso às informações gerais do local, agenda e mapa para ver o caminho.

O usuário do tipo empresa, após preencher o formulário de registro inicial, deverá preencher o formulário de cadastro do local, bem como a disponibilização de fotos do mesmo. Usuários empresa não podem cadastrar mais de um local. O usuário empresa poderá cancelar uma partida, com no máximo, 12 horas de antecedência. Ao cancelar uma partida, o local perderá pontos de reputação. O usuário empresa tem autonomia para definir a agenda de funcionamento das partidas realizadas no local.

## 4.2 MODELAGEM DO SISTEMA

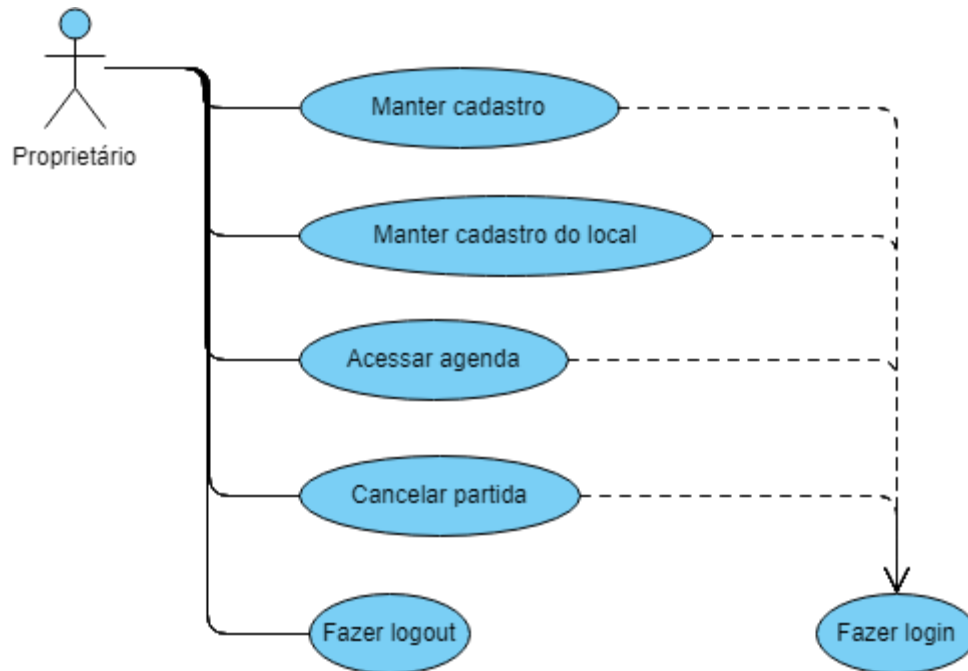
Na Figura 9 é apresentado o diagrama de casos de uso da aplicação na visão do usuário do tipo jogador. Nesse diagrama as atividades do ator são as de cadastro, buscar partida, buscar locais, agendar uma nova partida ou sair da aplicação.



Fonte: Autoria própria.

O caso de uso do usuário do tipo empresa é apresentado na Figura 10. A empresa poderá realizar o próprio cadastro, o cadastro do estabelecimento e acessar a agenda de horários do local, bem como sair da aplicação.

**Figura 10 - Caso de uso do usuário do tipo empresa**



Fonte: Autoria própria.

Na Figura 11 é apresentado o caso de uso do usuário administrador. O administrador é responsável por realizar e gerir o pré-cadastro dos usuários do tipo empresa.

Figura 11 - Caso de uso do usuário administrador



Fonte: Autoria própria.

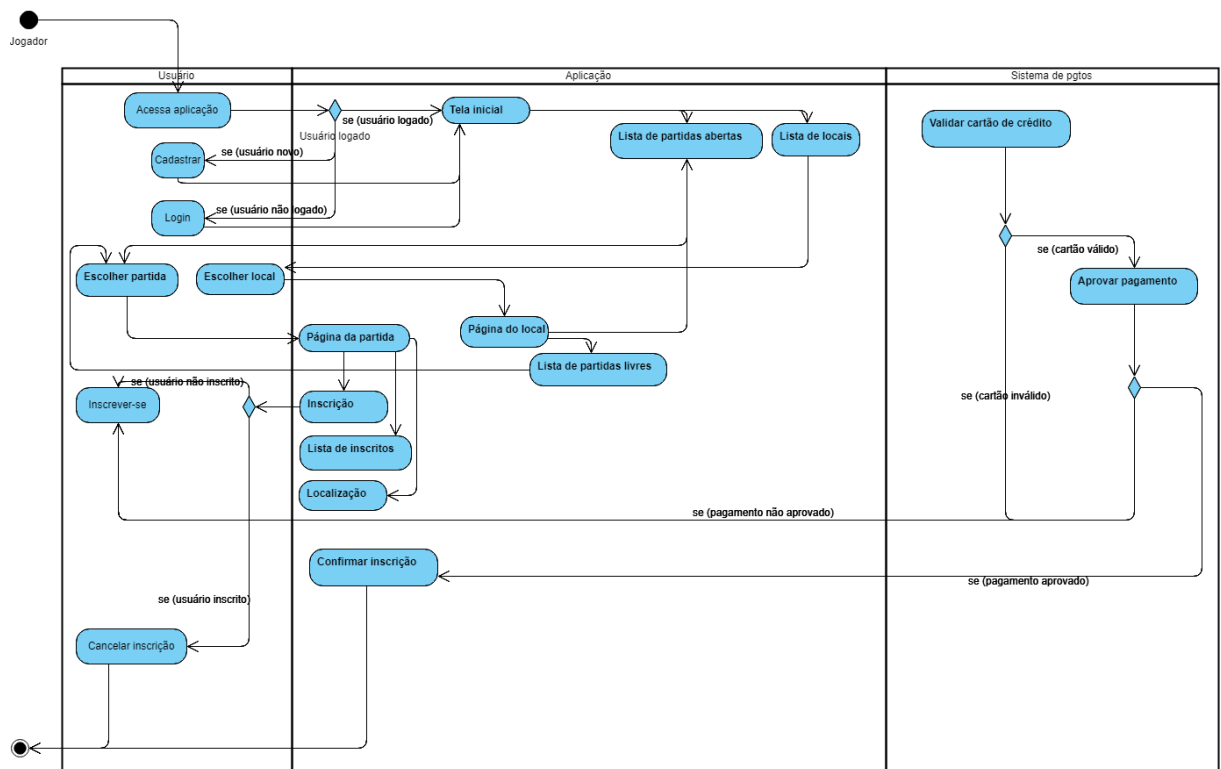
A Figura 12 apresenta o diagrama de atividades do processo de busca e inscrição de um usuário jogador em uma partida. O fluxo de atividades principal de um usuário do tipo jogador inicia ao acessar a tela principal, em que ele poderá decidir se buscará por locais ou por partidas. Em ambos os casos o usuário será redirecionado à uma página contendo uma lista dos itens que deseja buscar, podendo ter acesso a mais detalhes de cada item ao acessar a página de detalhes de um dos itens.

Caso a lista de partidas abertas seja acessada, a página de detalhes de uma partida dá ao usuário as opções de acesso a quem são os outros usuários inscritos na partida selecionada, bem

como a localização no mapa do local onde a partida ocorrerá. Na mesma página de detalhes de uma partida, existe a possibilidade de o usuário realizar sua inscrição na partida, a qual exigirá que usuário complete o pagamento da sua inscrição utilizando seu cartão de crédito.

O processo de pagamento fica a encargo de um sistema de pagamentos independente. Tão logo o pagamento seja confirmado ou negado, o usuário será informado da situação de seu pagamento e terá sua inscrição confirmada ou rejeitada, redirecionando-o para a página de detalhes da partida.

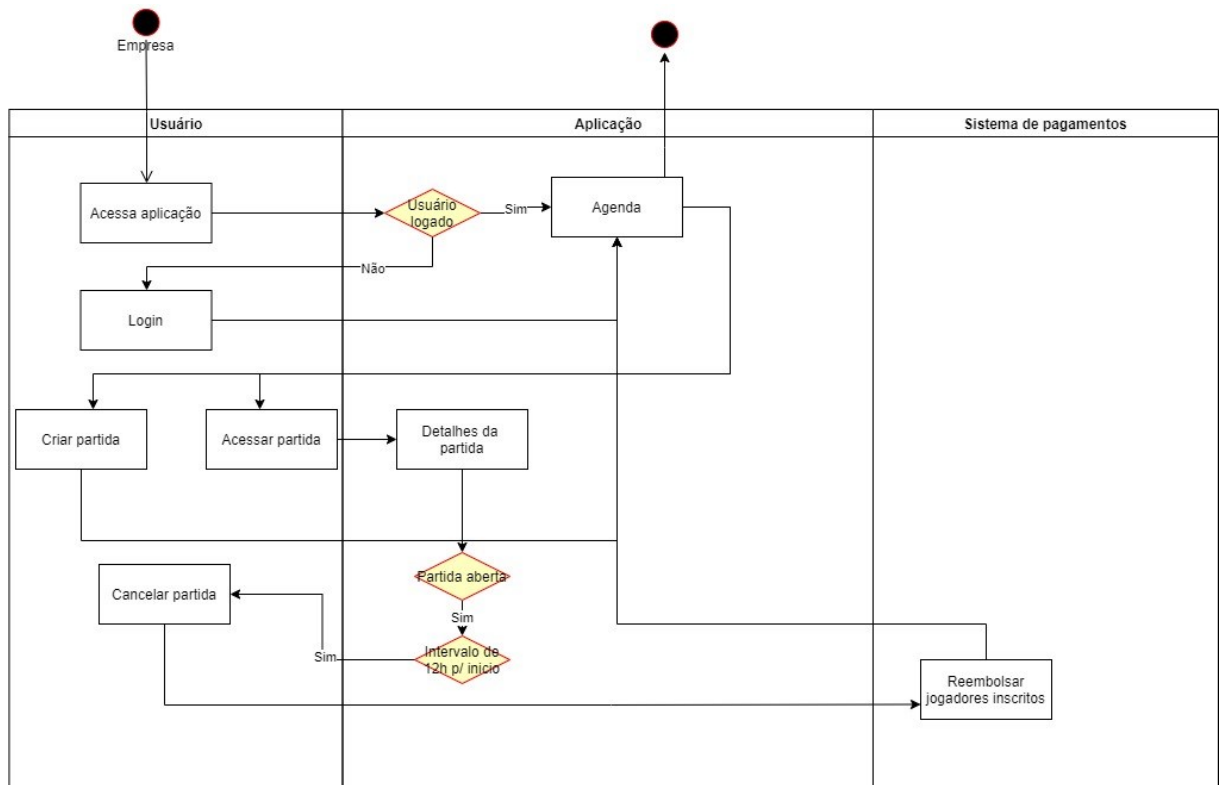
**Figura 12 - Diagrama de atividades de um usuário jogador**



**Fonte: Autoria própria.**

Na Figura 13 é apresentado o processo de agendamento de partidas por parte dos usuários do tipo empresa. O fluxo de uso do usuário empresa inicia na agenda, onde poderá agendar novas partidas ou acompanhar as inscrições e detalhes ou cancelar partidas agendadas previamente. Caso execute o cancelamento da partida, o sistema de pagamentos reembolsará os jogadores inscritos automaticamente.

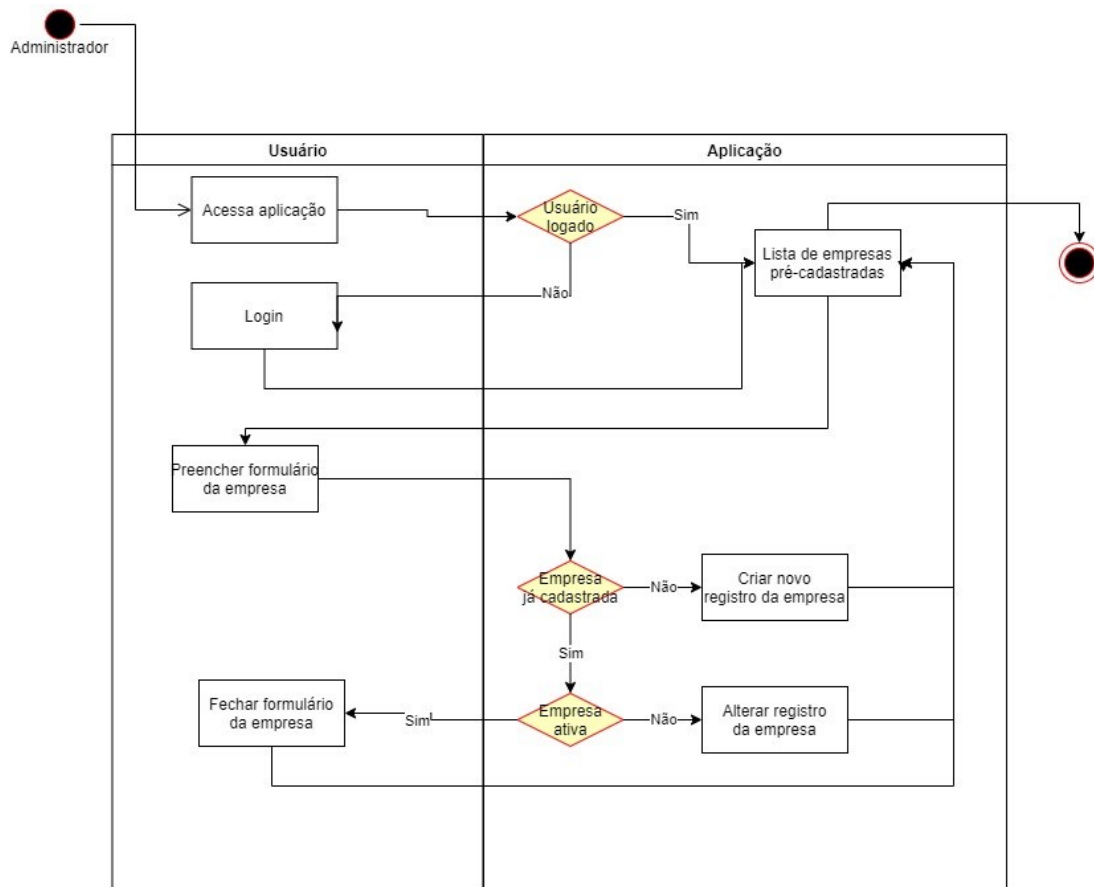
**Figura 13 - Diagrama de atividades do usuário empresa**



**Fonte: Autoria própria.**

A Figura 14 representa o diagrama de atividades dos usuários do tipo administrador. As atividades representadas no fluxo do diagrama representam o processo de criação e alteração do pré-cadastro de uma empresa.

**Figura 14 - Diagrama de atividades do usuário administrador**



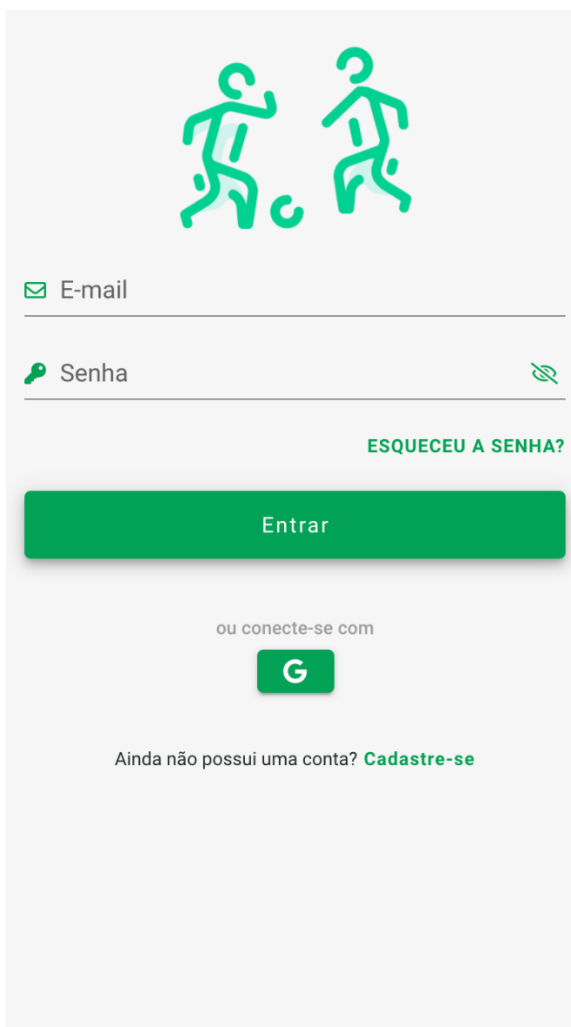
Fonte: Autoria própria.

### 4.3 APRESENTAÇÃO DO SISTEMA

Nesta seção serão apresentadas as imagens das funcionalidades implementadas na PWA para usuários do tipo administrador, empresa e jogador.

#### 4.3.1 Autenticação

A Figura 15 apresenta a tela de autenticação, que pode ser realizada de duas maneiras, utilizando e-mail e senha ou por meio de uma conta Google, que também pode ser utilizada para criar um usuário do tipo jogador. Todos os tipos de usuários da plataforma utilizam este formulário para acessá-la.

**Figura 15 - Autenticação de usuários**

A tela de autenticação de usuários apresenta um design limpo e moderno. No topo, há dois ícones de jogadores de futebol em cor verde. Abaixo, há dois campos de entrada: um para E-mail com um ícone de envelope e um para Senha com um ícone de chave e um ícone de olho desativado. Um link "ESQUECEU A SENHA?" está posicionado à direita do campo de senha. Um botão verde com o texto "Entrar" está centralizado. Abaixo do botão, o texto "ou conecte-se com" precede um ícone verde com a letra "G" (Google). Na base da tela, há o texto "Ainda não possui uma conta? [Cadastre-se](#)".

**Fonte: Autoria própria.**

#### 4.3.2 Usuário administrador

O usuário do tipo administrador possui uma única função dentro da aplicação, que é realizar o pré-cadastro de empresas, geri-las e desativar ou reativar o acesso à plataforma destes usuários.



**Figura 16 - Tela inicial do usuário administrador**

The screenshot displays the administrator interface for managing pre-registered companies. On the left is a green sidebar with navigation icons for profile, home, and settings. The main content area is titled 'Empresas pré-cadastradas' and includes a search bar and a table of company records. A green button labeled 'Adicionar empresa' is located in the top right corner.

Nome	Razão social	CNPJ	Ações
Bednar, Nolan And Lakin	Bednar, Nolan And Lakin	84.441.546/2418-84	
Borer, Yundt And Rempel	Borer, Yundt And Rempel	27.686.241/2020-62	
Daugherty Group	Daugherty Group	88.078.581/3048-50	
Empresa De Testes	Empresa De Testes Ltda	34.839.996/0001-45	
Empresa De Testes 2	Empresa De Testes 2 Me	31.655.745/0001-12	
Fahey, Hauck And Metz	Fahey, Hauck And Metz	52.417.727/5348-00	
Feil, Wolff And Streich	Feil, Wolff And Streich	58.576.753/2712-39	
First Company	First Company Ltda	61.136.926/0001-01	
Gleichner - Tremblay	Gleichner - Tremblay	64.113.858/3541-17	
Green - Blick	Green - Blick	85.660.140/1010-93	

**Fonte: Autoria própria.**

Todo o uso da aplicação, por usuários do tipo empresa, é centralizado na página inicial, ilustrada na Figura 16, a qual contém a lista das empresas cadastradas. Essa lista permite que o administrador edite o cadastro ou remova uma empresa. Contudo, para que realmente seja possível realizar a remoção, é necessário que esta empresa ainda não possua um usuário ativo.

Para efetuar o pré-cadastro de uma empresa, é necessário que o administrador clique no botão “Adicionar empresa” e preencha o formulário exibido na Figura 17. Após concluir esta etapa, o administrador torna a empresa apta a utilizar a plataforma.

**Figura 17 - Formulário de pré-cadastro de empresas**

The image shows a modal window titled "Adicionar empresa" with a close button (X) in the top right corner. The modal contains three input fields, each with a green icon to its left: "CNPJ" (with a document icon), "Razão social" (with a document icon), and "Nome fantasia" (with a document icon). A green "Salvar" button is located at the bottom right of the modal. In the background, a table is visible with columns for "Nome", "CNPJ, razão social ou nome fantasia", and "Ações". The table contains several rows of company data, including "Bednar, Nolan And Laki", "Borer, Yundt And Remp", "Daugherty Group", "Empresa De Testes", "Empresa De Testes 2", "Fahey, Hauck And Metz", and "Feil, Wolff And Streich".

**Fonte: Autoria própria.**

### 4.3.3 Usuário empresa

Usuários do tipo empresa são os responsáveis por disponibilizar os locais para prática recreativa do futebol.

Para que estes usuários possam utilizar a plataforma, é necessário primeiramente possuir o pré-cadastro na aplicação, que é realizado pelo administrador. Com o pré-cadastro concluído, basta que inicie a criação de seu perfil na plataforma.

Na tela de autenticação, na Figura 15, deve-se clicar no botão “Cadastre-se” e escolher a aba “Empresa”, exibida na Figura 18, para que tenha acesso ao formulário de cadastro de empresas. Ao preencher seu CNPJ seus dados serão importados automaticamente do pré-cadastro, restando ao usuário preencher seu e-mail e senha de autenticação.

**Figura 18 - Formulário de cadastro de empresas**

← JOGADOR EMPRESA

CNPJ

Razão social

Nome fantasia

E-mail

Senha

Repita a senha

Continuar

**Fonte: Autoria própria.**

Após concluir esta etapa, o usuário terá que completar outras duas etapas, que são: preencher o formulário de endereço, apresentado na Figura 19 e apontar a localização no mapa, como ilustrado na Figura 20.

**Figura 19 - Formulário de endereço da empresa**

1 Qual seu endereço?  
Digite seu CEP e nós faremos o resto

85505-140 Rua Tocantins

321 Próximo ao Supermercado

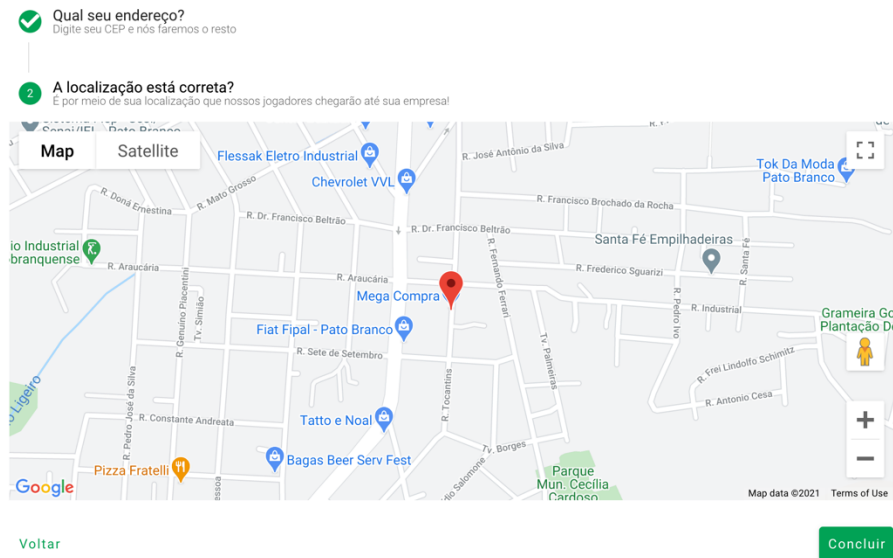
Baixada Paraná Pato Branco

Próximo

2 A localização está correta?  
É por meio de sua localização que nossos jogadores chegarão até sua empresa!

**Fonte: Autoria própria.**

**Figura 20 - Seleção da localização da empresa**

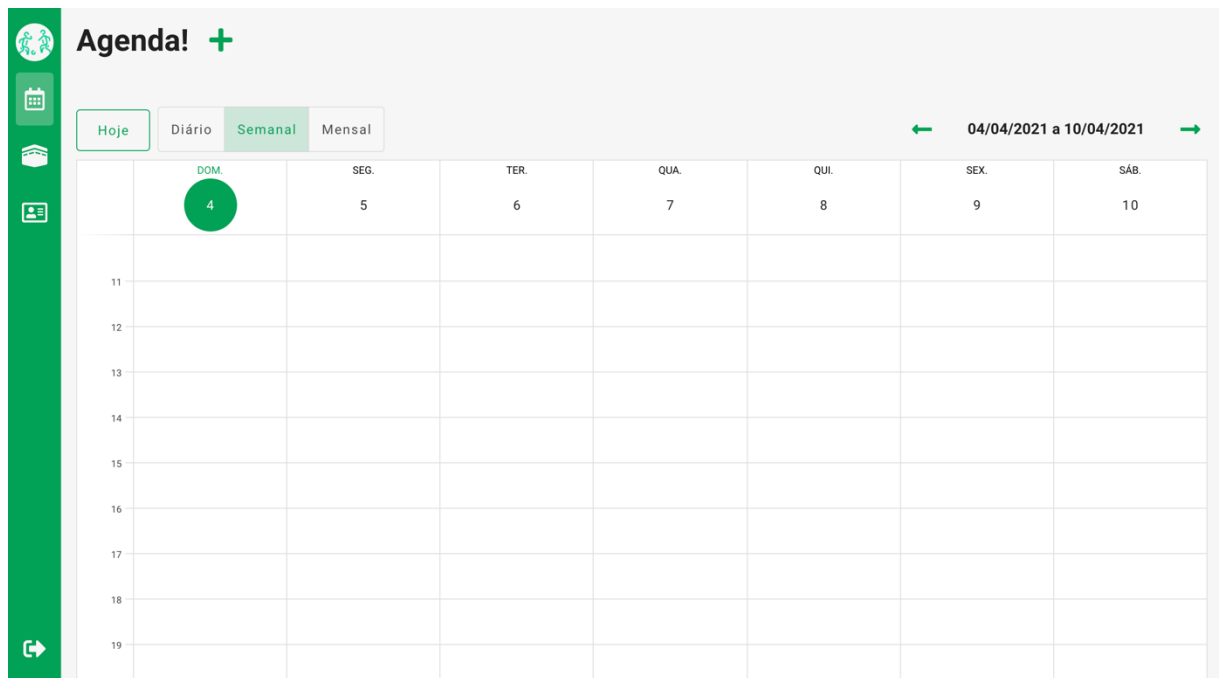


**Fonte: Autoria própria.**

No formulário de endereço, é necessário que o CEP seja informado para que o restante das informações sejam auto completadas, restando apenas o número do local e o complemento, se necessário. A localização é selecionada a partir do endereço preenchido. O usuário poderá ajustar a localização caso a seleção não seja realizada de forma precisa.

Após clicar no botão “Concluir” o usuário será redirecionado para a tela inicial da empresa e sua agenda será exibida pela primeira vez, como pode ser observado na Figura 21. Por padrão, a agenda é iniciada em modo de exibição semanal, selecionando o período da semana atual, referente a data de acesso, de Domingo a Sábado. Além do modo semanal, o usuário pode trocar o modo de exibição para diário, selecionando uma data específica, e modo Mensal, exibindo a agenda completa de acordo com o mês selecionado.

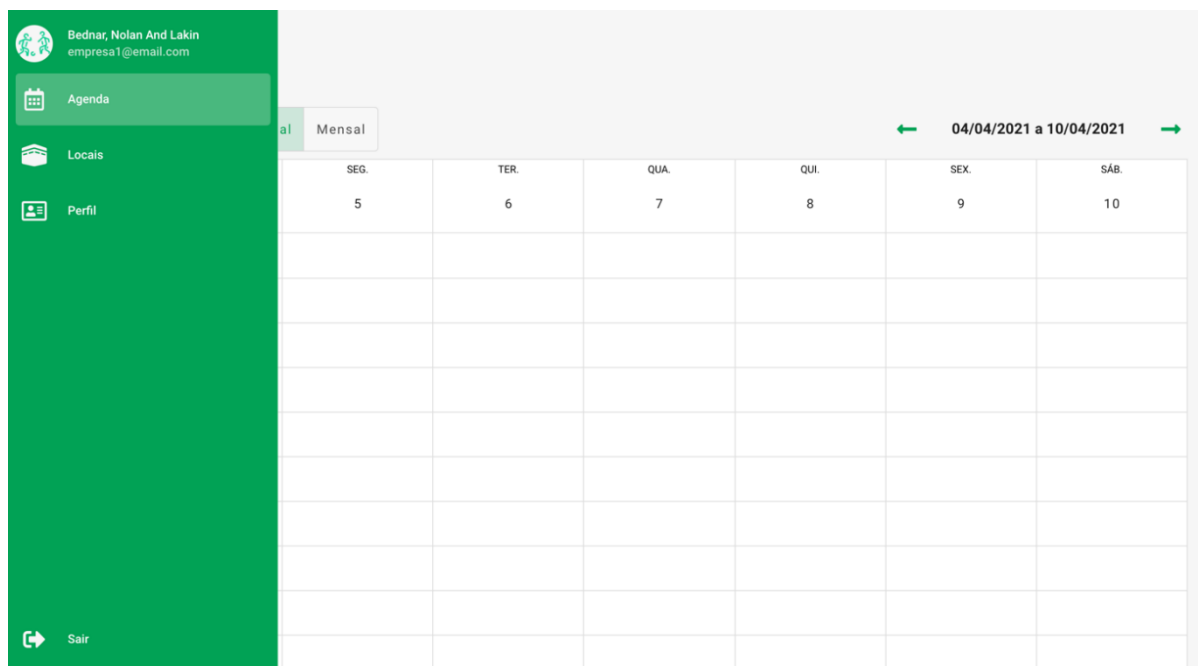
**Figura 21 - Agenda da empresa no primeiro acesso**



**Fonte: Autoria própria.**

Na Figura 22, está exibido o menu disponível para usuários do tipo empresa. Neste menu existem apenas quatro opções, sendo elas: Agenda, Locais, Perfil e Sair, respectivamente. É por meio desse menu que os usuários terão acesso às funcionalidades da plataforma.

**Figura 22 - Menu de usuário empresa**

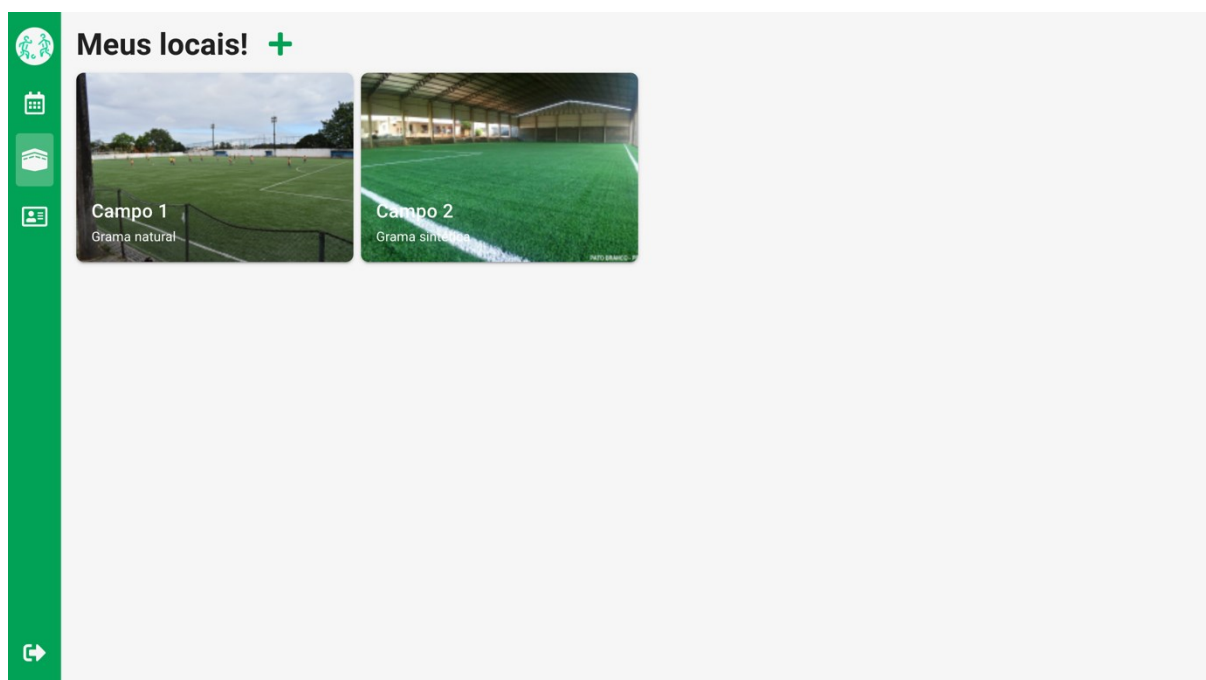


**Fonte: Autoria própria.**

A opção “Agenda”, exibida por completo na Figura 21, é a tela inicial de um usuário do tipo empresa. Já a opção “Locais” redireciona para a lista de locais que o usuário cadastrou. Os detalhes desta funcionalidade serão tratados adiante, assim como a opção Perfil, que também fará um redirecionamento para outra tela da aplicação, que contém formulários que possibilitam o usuário do tipo empresa manter seus dados cadastrais sempre atualizados. O botão “Sair”, localizado na parte inferior do menu, é utilizado para encerrar a sessão do usuário ativo. Este botão está disponível para todos os usuários da aplicação.

A primeira ação de um usuário do tipo empresa dentro da aplicação deve ser cadastrar os espaços que pode disponibilizar para a prática de futebol. Essa ação é obrigatória, pois, caso não haja nenhum espaço de prática cadastrado, nenhuma partida poderá ser agendada nesta empresa. Na Figura 23 está a tela em que este usuário deve utilizar para a gestão de seus espaços. Os locais são exibidos de tal forma que sejam identificados facilmente e que a lista se adapte a qualquer dispositivo.

**Figura 23 - Lista de locais cadastrados**



**Fonte: Autoria própria.**

Para adicionar um novo local é necessário que o usuário clique no botão “+” que está ao lado do título da página, “Meus locais!”. Será exibido um formulário em que todas as informações são de preenchimento obrigatório. Como exibido na Figura 24, o formulário é intuitivo e de fácil entendimento, com dados importantes para ajudar os jogadores a escolherem o local em que desejam praticar o futebol recreativo.

**Figura 24 - Formulário de cadastro de locais**

**Adicionar Local!**

Nome: > Campo 2 | Tipo: > Grama sintética

Dias de disponibilidade: Segunda-Feira, Terça-Feira, Domingo, Quarta-Feira, Quinta-Feira, Sexta-Feira, Sábado

Abre as: 16:00 | Fecha as: 22:00 | Total de jogadores: 10

Preço/h: R\$ 150,00

Imagens: campo-society-1.webp (1 arquivo(s))

Cancelar | Salvar

**Fonte: Autoria própria.**

Após realizar o cadastro dos locais, o usuário poderá acessar detalhes do local cadastrado. Na Figura 25, é exibida a página de um local cadastrado, contendo detalhes de sua operação, como dias de funcionamento, valor e quantidade de jogadores e também as avaliações deixadas por jogadores que já participaram de uma partida de futebol naquele espaço.

**Figura 25 - Página de detalhes de um local cadastrado**

**Campo 2**

- Grama natural (Tipo)
- Seg, Ter, Qua, Qui, Sex (Dias de funcionamento)
- 18:00 às 23:00 (Horários de funcionamento)
- R\$ 300,00 (Preço/h)

**Avaliações**

USUÁRIO DE APRESENTAÇÃO

★★★★★

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

**Fonte: Autoria própria.**

É por meio da página de detalhes de um local que o usuário realizará alterações nas configurações de funcionamento do local ou irá desativá-lo. Um local, quando desativado, se tornará indisponível para o agendamento de partidas dentro da aplicação.

Após conclusão do cadastro dos locais, o usuário do tipo empresa está habilitado a criar eventos e/ou receber agendamentos a partir de usuários jogadores.

Para disponibilizar eventos, o usuário do tipo empresa deve abrir o formulário de evento utilizando o botão ao lado do título da página inicial, que contém sua agenda. Ao clicar neste botão deverá preencher o formulário com as informações necessárias, como na Figura 26(a), e então disponibilizará uma partida disponível na lista de partidas em que todos os usuários jogadores da aplicação tem acesso.

**Figura 26 - Agendamento de partidas. a) Formulário de agendamento. b) Lista de partidas agendadas**

**(a) Formulário de agendamento:**

Evento

Local

> Campo 1

Data

> 10/04/2021

Hora

> 15:00

Valor

> R\$ 200,00

Cancelar Agendar

**(b) Lista de partidas agendadas:**

Hoje Diário Semanal Mensal

10/04/2021

SÁB. 10

15	Partida em Second Company no Campo 1, 2 PM - 3 PM
16	Partida em Second Company no Campo 1, 3 PM - 4 PM
17	Partida em Second Company no Campo 1, 4 PM - 5 PM
18	Partida em Second Company no Campo 1, 5 PM - 6 PM
19	Partida em Second Company no Campo 2, 7 PM - 8 PM
20	Partida em Second Company no Campo 2, 8 PM - 9 PM
21	Partida em Second Company no Campo 2, 9 PM - 10 PM
22	Partida em Second Company no Campo 2, 10 PM - 11 PM

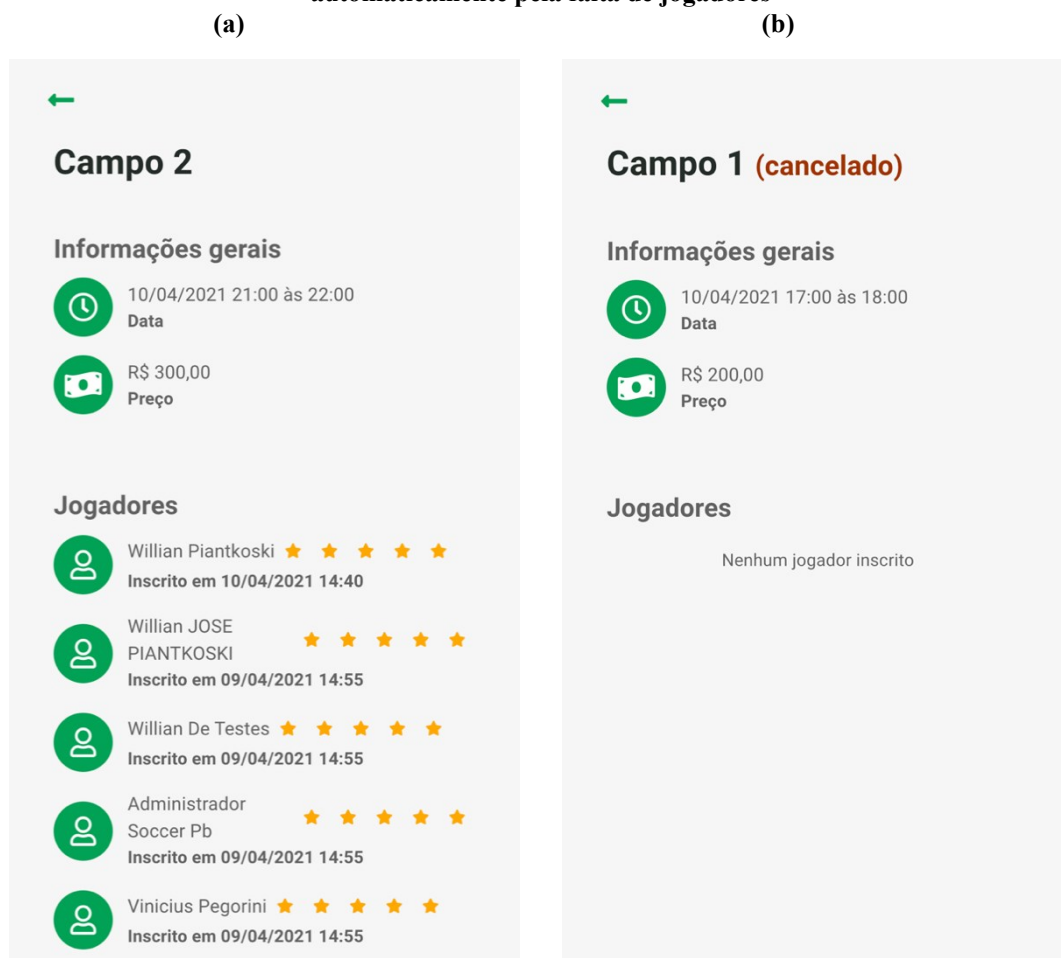
Fonte: Aatoria própria.

Todas as partidas agendadas ficarão disponíveis na agenda da empresa, como na Figura 26(b), independente se foram agendadas por jogadores ou pela própria empresa.



Cada item na agenda representa uma partida agendada que, para ver seus detalhes, basta que o usuário clique em um dos itens e terá acesso aos dados da partida, bem como a lista de jogadores que já estão inscritos nela, como na Figura 27(a).

**Figura 27 - Detalhes de uma partida agendada. a) Partida disponível para inscrições. b) Partida cancelada automaticamente pela falta de jogadores**



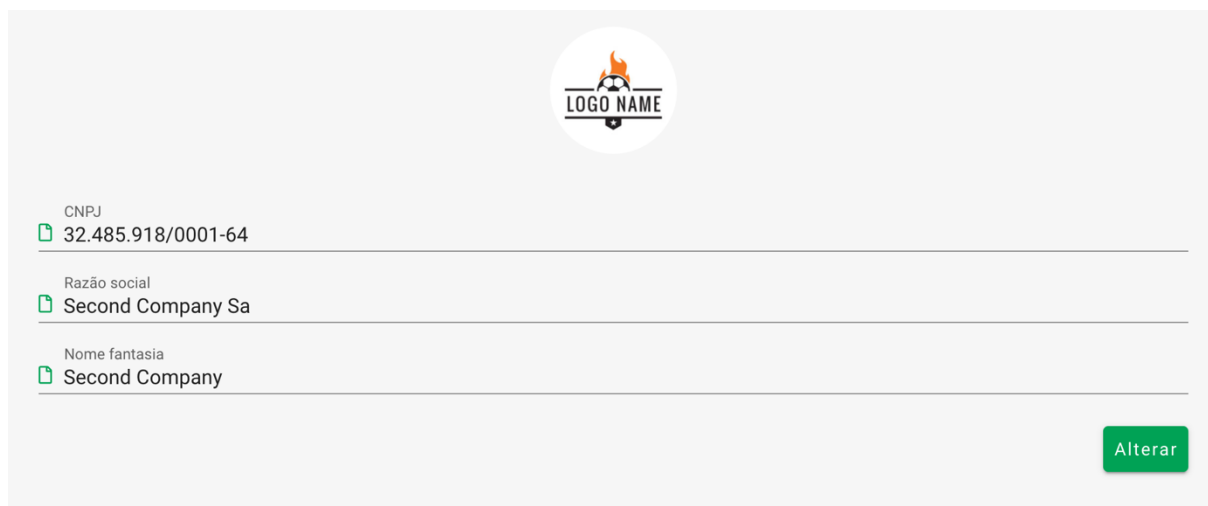
**Fonte: Autoria própria.**

Na Figura 27(b) é exibido um caso em que uma partida não atingiu o número mínimo de jogadores inscritos. O cancelamento é realizado de forma automática ou pela própria empresa e, caso seja uma partida que exija um pagamento por parte dos jogadores, estes serão reembolsados. Quanto aos cancelamentos de partidas pela empresa, só poderá ser executado com 12 (doze) horas de antecedência ao horário de início da partida.

Para que todos os agendamentos sejam precisos, é necessário que a empresa mantenha seus dados cadastrais sempre atualizados, para isso, basta que acesse seu perfil, utilizando o menu.

O perfil, como exibido nas figuras 28, 29 e 30, é constituído por formulários que são responsáveis por atualizar determinadas informações. O primeiro formulário, na Figura 28, é responsável pelas informações básicas da empresa, como nome, CNPJ e razão social.

**Figura 28 - Formulário de informações básicas da empresa**



The image shows a web form for updating company information. At the top center is a circular logo placeholder with a soccer ball icon and the text "LOGO NAME". Below the logo are three input fields, each with a small document icon to its left. The first field is labeled "CNPJ" and contains the value "32.485.918/0001-64". The second field is labeled "Razão social" and contains "Second Company Sa". The third field is labeled "Nome fantasia" and contains "Second Company". In the bottom right corner of the form area, there is a green button with the text "Alterar".

**Fonte: Autoria própria.**

A seguir, está o formulário que detém os dados de endereço e localização da empresa. Este formulário funciona exatamente como o formulário ilustrado na Figura 19.

**Figura 29 - Formulário de endereço e localização da empresa**

### Endereço

CEP	Logradouro	
85506-140	Rua Bartolomeu Bueno	
Número	Complemento	
12	Teste	
Bairro	Estado	Cidade
Pinheirinho	Paraná	Pato Branco

**Alterar**

### Localização

**Alterar**

Fonte: Autoria própria.

Finalizando a tela de perfil, na Figura 30, está o formulário para alteração de e-mail. Basta que o usuário preencha seu novo e-mail, aguarde a validação do mesmo e confirme a atualização. Se alterado com sucesso, o usuário terá sua sessão encerrada e poderá realizar *login* novamente, utilizando o novo e-mail.

**Figura 30 - Alteração de e-mail e senha**

The image shows a user interface for account management, divided into two sections: "Alterar e-mail" and "Alterar senha".

**Alterar e-mail**

E-mail atual  
secondcompany@email.com

Novo e-mail

Alterar

**Alterar senha**

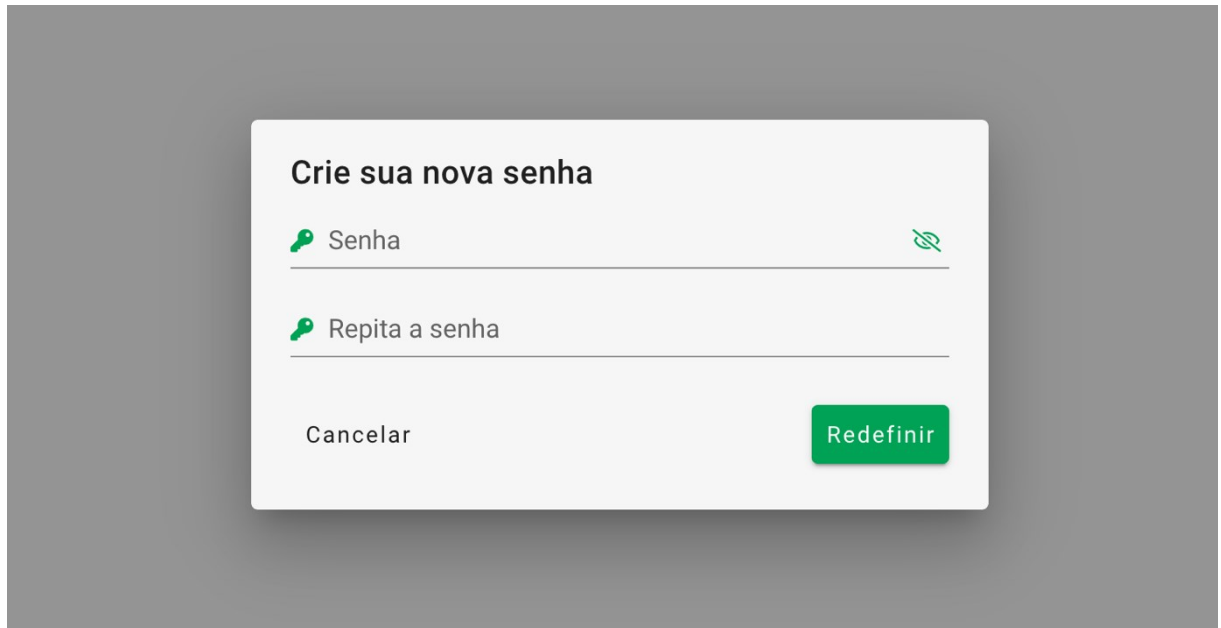
Envie um link de redefinição de senha para o email secondcompany@email.com.

Enviar link

**Fonte: Aatoria própria.**

Além dos formulários citados, também há a possibilidade de solicitar a alteração da senha. Ao clicar no botão “Enviar link”, um link de redefinição de senha é enviado para o e-mail do usuário. Ao clicar neste link, o usuário será redirecionado para a plataforma para que crie sua nova senha, como na Figura 31. Ao criar sua senha o usuário terá sua sessão encerrada e poderá realizar *login* novamente, utilizando a nova senha.

As funcionalidades de alterar de e-mail e senha, são iguais para todos os tipos de usuários e possuem os mesmos comportamentos.

**Figura 31 - Formulário de redefinição de senha**A imagem mostra um formulário de redefinição de senha em um modal branco sobre um fundo cinza escuro. O título do modal é "Crie sua nova senha". Abaixo do título, há dois campos de entrada de texto. O primeiro campo é rotulado "Senha" e possui um ícone de chave verde à esquerda e um ícone de olho desligado verde à direita. O segundo campo é rotulado "Repita a senha" e também possui um ícone de chave verde à esquerda. Na base do modal, há dois botões: "Cancelar" em texto cinza e "Redefinir" em um botão verde com texto branco.

**Fonte: Autoria própria.**

#### 4.3.4 Usuário jogador

Os jogadores são todos os usuários que utilizam a plataforma para buscar locais mais próximos e adequados às suas preferências em que possam praticar o futebol de forma recreativa.

Para iniciar o uso da plataforma os usuários jogadores podem fazer seu cadastro de duas maneiras: utilizando o formulário de cadastro, ilustrado na Figura 32(a), ou utilizando uma conta do Google, como na Figura 32(b).

Se desejar utilizar o formulário, o jogador deve clicar no botão “Cadastre-se” ou, se desejar utilizar sua conta Google, deve clicar no botão que possui a logo do Google. Ambos os botões estão posicionados no centro da tela de autenticação, exibida na Figura 15.

**Figura 32 - Cadastro de usuário jogador. a) Formulário de cadastro. b) Cadastro com conta Google**

The image displays two screenshots related to user registration. Screenshot (a) shows a mobile application interface with a navigation bar at the top containing 'JOGADOR' (highlighted in green) and 'EMPRESA'. Below the navigation bar are four input fields: 'Nome completo' (with a person icon), 'E-mail' (with an envelope icon), 'Senha' (with a key icon and a toggle for visibility), and 'Repita a senha'. A green button labeled 'Cadastrar' is positioned at the bottom. Screenshot (b) shows a web browser window titled 'Sign in - Google Accounts'. The address bar shows 'accounts.google.com/o/oauth2/auth/oauthchoo...'. The main content area says 'Sign in with Google' and 'Choose an account to continue to soccer-pb-dev.firebaseio.com'. Three accounts are listed: 'Willian Piantkoski' (piantkoskiwillian@gmail.com), 'Willian Piantkoski' (wjpiantkoski@gmail.com), and 'Willian JOSE PIANTKOSKI' (piantkoski@alunos.utfpr.edu.br). There is also a 'Use another account' option. At the bottom, it states: 'To continue, Google will share your name, email address, language preference, and profile picture with soccer-pb-dev.firebaseio.com.' and includes links for 'English (United States)', 'Help', 'Privacy', and 'Terms'.

**Fonte: Autoria própria.**

Ao optar pelo cadastro utilizando o formulário, o usuário deverá preencher todas as informações solicitadas. Caso optar por utilizar uma conta do Google, ao clicar no botão, será redirecionado para uma tela onde todas as suas contas já disponíveis em seu navegador serão listadas, com a possibilidade de adicionar outra conta, caso esta não esteja presente na lista. Após a seleção da conta que deseja utilizar ou a conclusão do preenchimento do formulário, o usuário será redirecionado para a próxima etapa de criação do perfil de jogador, como ilustrado na Figura 33. Nessa tela o jogador deverá selecionar em que setor da partida prefere jogar e qual é seu estilo de jogo.

**Figura 33 - Última etapa de criação do perfil de jogador**

**Bem vindo(a), Willian!**  
Antes de continuar, gostaríamos de saber mais sobre voce.

Onde voce mais gosta de jogar?

Na defesa

Como é seu estilo de jogo?

Ofensivo

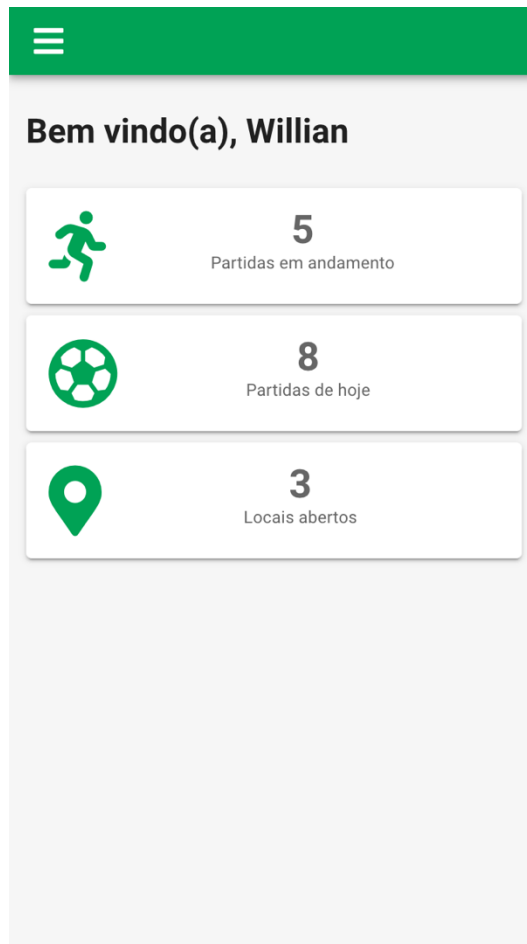
Concluir

A imagem mostra uma interface de usuário para a criação de perfil. No topo, há uma saudação personalizada: 'Bem vindo(a), Willian!'. Abaixo, um texto explica que o sistema quer saber mais sobre o usuário antes de continuar. Há duas perguntas com menus suspensos: 'Onde voce mais gosta de jogar?' com a opção 'Na defesa' selecionada, e 'Como é seu estilo de jogo?' com a opção 'Ofensivo' selecionada. Um botão verde 'Concluir' está visível no canto inferior direito da seção de perguntas.

**Fonte: Autoria própria.**

Após concluir as etapas de cadastro, o usuário será redirecionado para a tela inicial da plataforma, ilustrada na Figura 34. Esta tela possui três *cards* com informações em tempo real sobre as partidas e locais cadastrados.

Figura 34 - Tela inicial do usuário jogador



Fonte: Autoria própria.

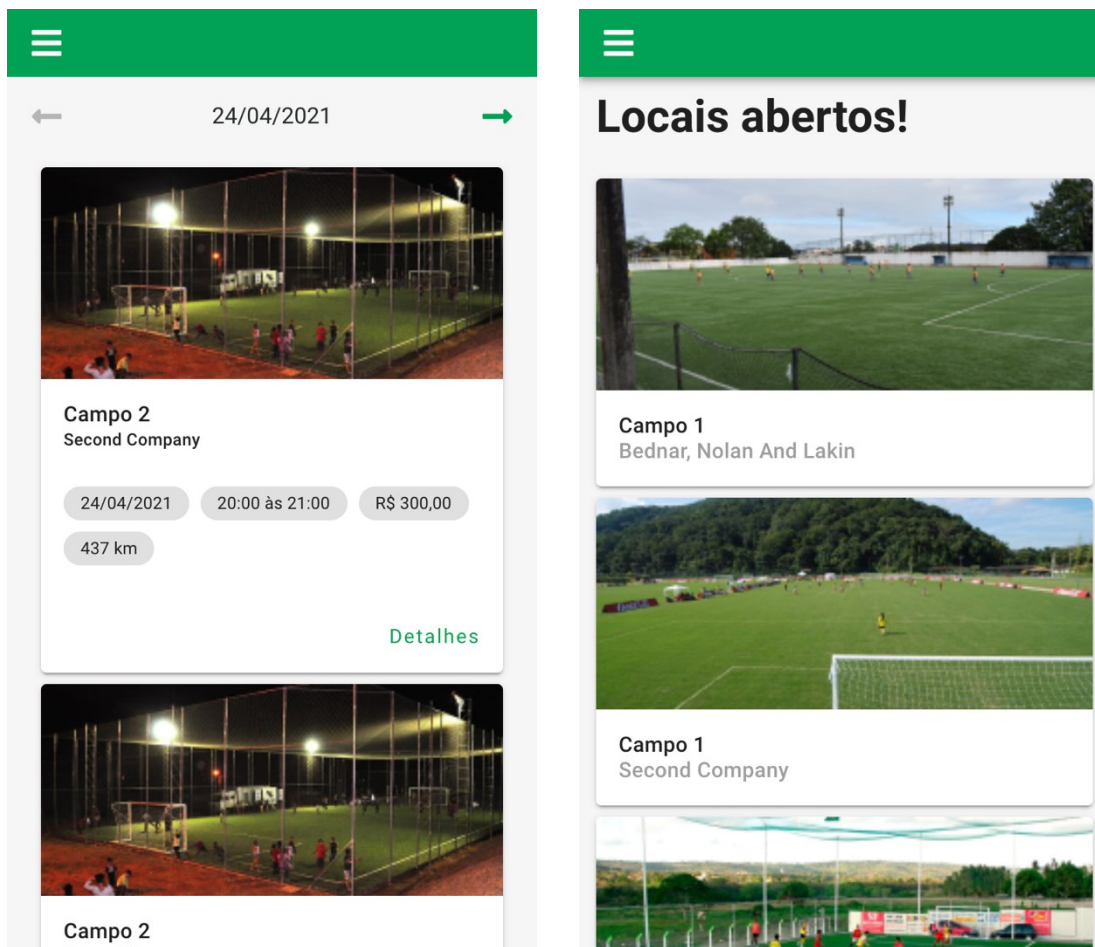
O primeiro *card*, com o título “Partidas em andamento”, exibe o total de partidas que estão ocorrendo naquele momento do acesso. Este *card* não possui nenhuma outra função além de exibir este dado.

O segundo *card*, “Partidas hoje”, representa o total de partidas disponíveis para inscrição na data. Ao clicar no *card*, o usuário será redirecionado para a lista de partidas, como exibido na Figura 35(a), disponíveis no momento do acesso.

O terceiro e último *card*, “Locais abertos”, se refere ao número de espaços para prática de futebol recreativo que estão disponíveis no momento do acesso. Clicando no *card*, o usuário visualizará a tela que contém a lista de locais disponíveis, como apresentado na Figura 35(b), para agendamento de partidas.



**Figura 35 - Listas de partidas disponíveis e locais. a) Lista de partidas disponíveis. b) Lista de locais**



**Fonte: Autoria própria.**

Na lista de partidas disponíveis, Figura 35(a), cada *card* representa uma partida e exibe informações importantes, para que o usuário possa escolher a partida, sendo elas a foto do local, nome da empresa, data e horário da partida, preço total e distância em relação a localização atual do jogador.

Para realizar a inscrição e/ou saber mais detalhes sobre a partida, é necessário clicar no botão “Detalhes”. Na tela de detalhes da partida, existirão informações mais completas que estarão divididas em três abas.

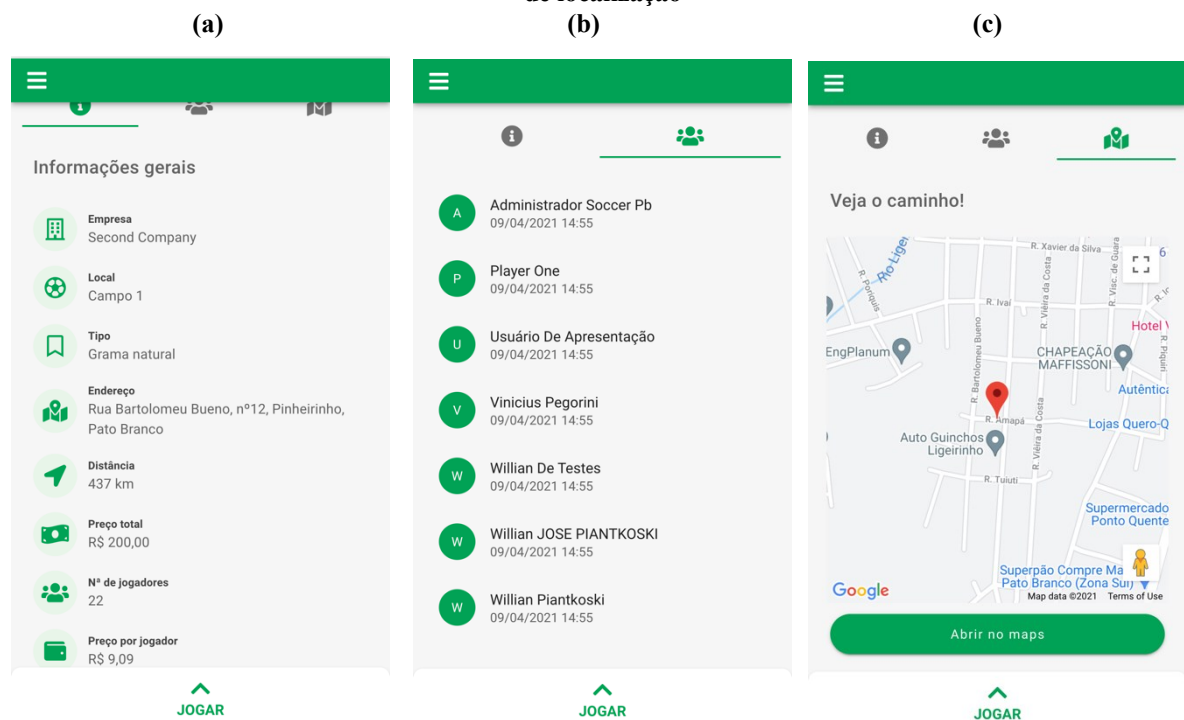
A primeira aba, disponível na Figura 36(a), contém informações adicionais como o endereço completo, quantidade de jogadores necessários e o preço que cada jogador deverá pagar pela inscrição, caso haja custos. O preço é formado por uma divisão entre o preço total e o número de jogadores necessários, portanto, todos os jogadores pagam o mesmo preço.

Na segunda aba, denominada “Jogadores”, exibida na Figura 36(b), são listados os jogadores que fizeram a inscrição na partida escolhida. Nesta lista, cada jogador é representado

por sua foto, caso tenha sido adicionada pelo usuário, nome, data e hora de quando realizou a inscrição.

Finalizando as abas disponíveis na página de detalhes da partida, existe a aba chamada “Localização”, ilustrada na Figura 36(c), que é responsável por exibir a localização exata do local possibilitando ao usuário abrir seu aplicativo de mapas e encontrar o melhor caminho para chegar de acordo com sua localização atual.

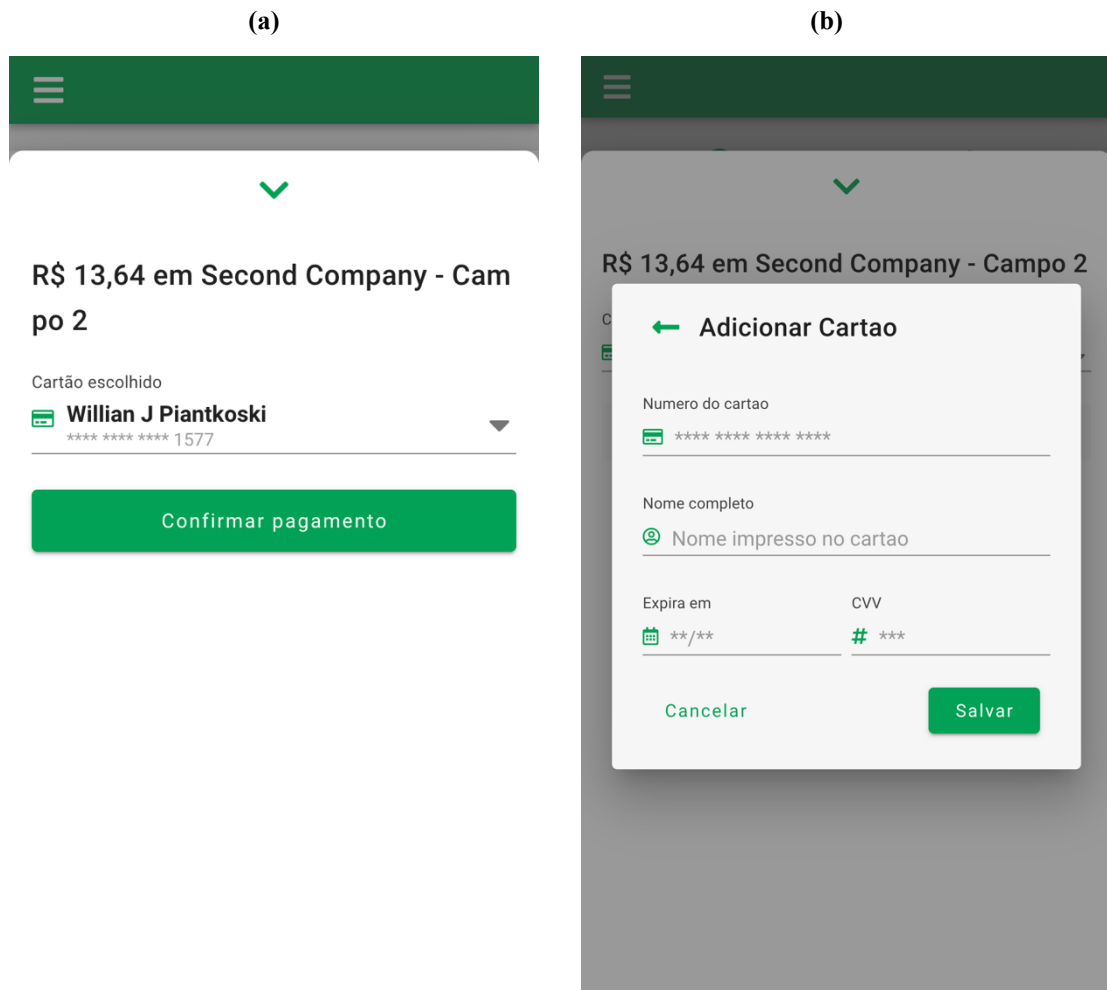
**Figura 36 - Detalhes de uma partida. a) Aba de informações gerais. b) Aba de jogadores inscritos. c) Aba de localização**



**Fonte: Autoria própria.**

Na parte inferior das telas de detalhes de uma partida está localizado o botão “Jogar”, é por meio deste botão que o jogador poderá fazer sua inscrição na partida. Ao clicá-lo, caso a partida exija um pagamento, o formulário de seleção de cartão de crédito para pagamento será exibido, como na Figura 37(a), bastando ao usuário escolher seu cartão e confirmar. Caso o usuário ainda não possua um cartão de crédito cadastrado na plataforma, basta clicar no botão “Cadastrar cartão” e inserir os dados de seu cartão no formulário, como na Figura 37(b).

**Figura 37 - Pagamento de inscrição na partida. a) Seleção do cartão de crédito. b) Cadastro de novo cartão**



**Fonte: Autoria própria.**

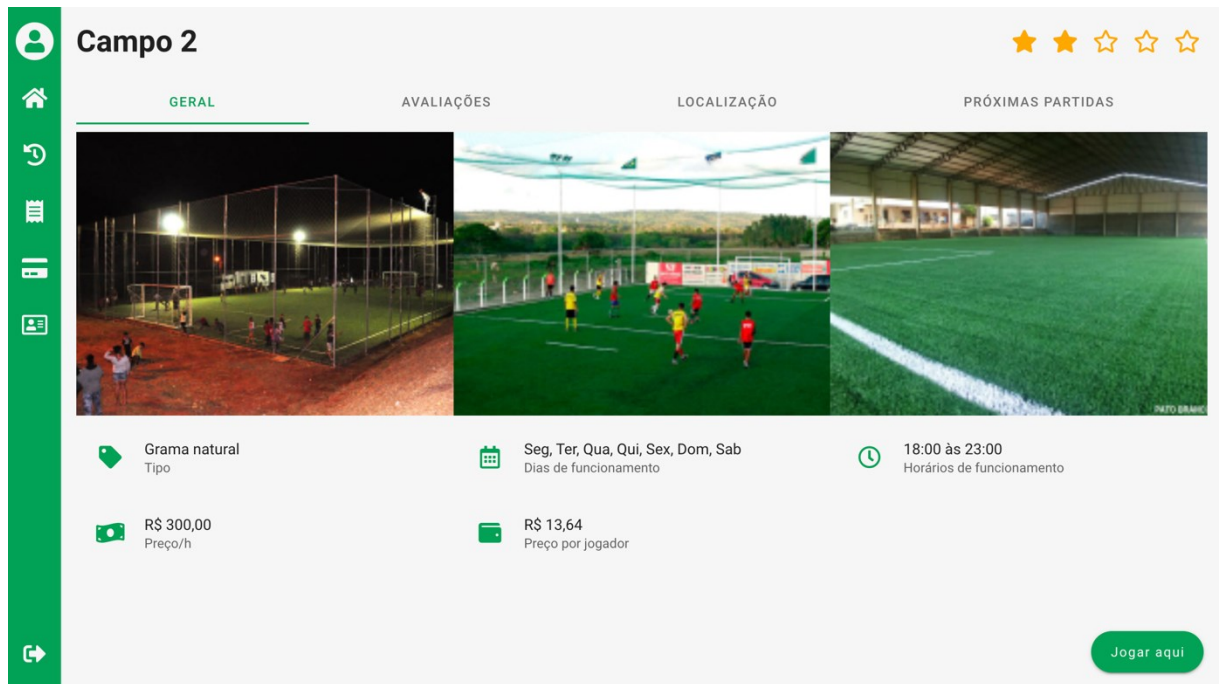
Após aprovação do pagamento, o usuário terá sua inscrição confirmada, restando apenas comparecer ao local na data e horário agendados.

Caso necessário, o usuário pode cancelar sua inscrição. Esta ação pode ser realizada em até doze horas de antecedência ao horário da partida e causará a perda de pontos de reputação do jogador. Há também a possibilidade de cancelamento da partida por parte da empresa que, assim como os jogadores, pode cancelar em até doze horas de antecedência ao horário da partida e perderá pontos de reputação.

A reputação de jogadores e de empresas, em casos de cancelamento, recebe um desconto de 0,25 pontos no seu total. Os pontos também sofrem mudanças quando um jogador conclui a avaliação da última partida que disputou, ocorrendo um acréscimo de 0,25 pontos em sua pontuação e atualizando a pontuação da empresa utilizando um cálculo simples de soma entre a pontuação atual e a nota dada pelo jogador e, com o total calculado, fazendo uma divisão simples por 2. A pontuação máxima, tanto para jogadores, como para empresas, é de 5 pontos.

Existe também uma forma alternativa de buscar partidas, por meio da lista de locais aberto, exibida na Figura 35(b). Ao escolher um local da lista e clicar em sua opção, o usuário será redirecionado para a página de detalhes do local, ilustrada na Figura 38. Esta página possui quatro abas que contém informações que ajudam o usuário escolher o local em que deseja jogar.

**Figura 38 - Detalhes de um local**

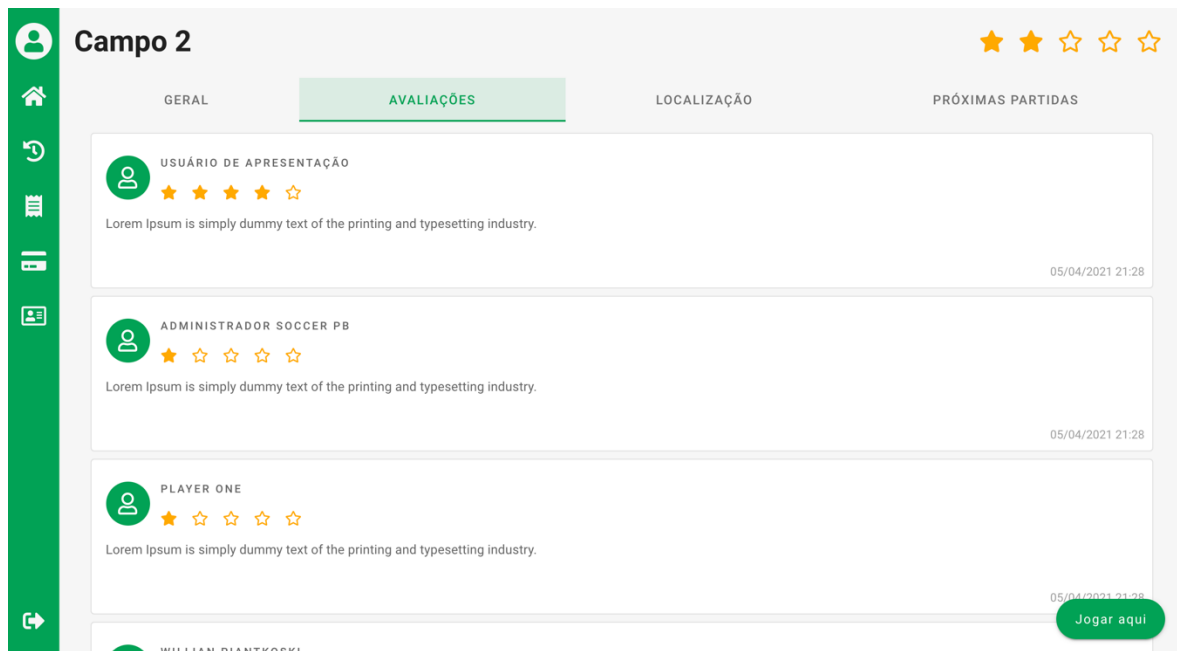


**Fonte: Autoria própria.**

Na aba “Geral” estão contidas informações como as fotos, tipo do terreno, dias e horários de funcionamento, preço total e preço por jogador.

Na aba “Avaliações”, exibida na Figura 39, estão listadas as avaliações de jogadores que já jogaram no local. A nota dessas avaliações é que forma a nota do local.

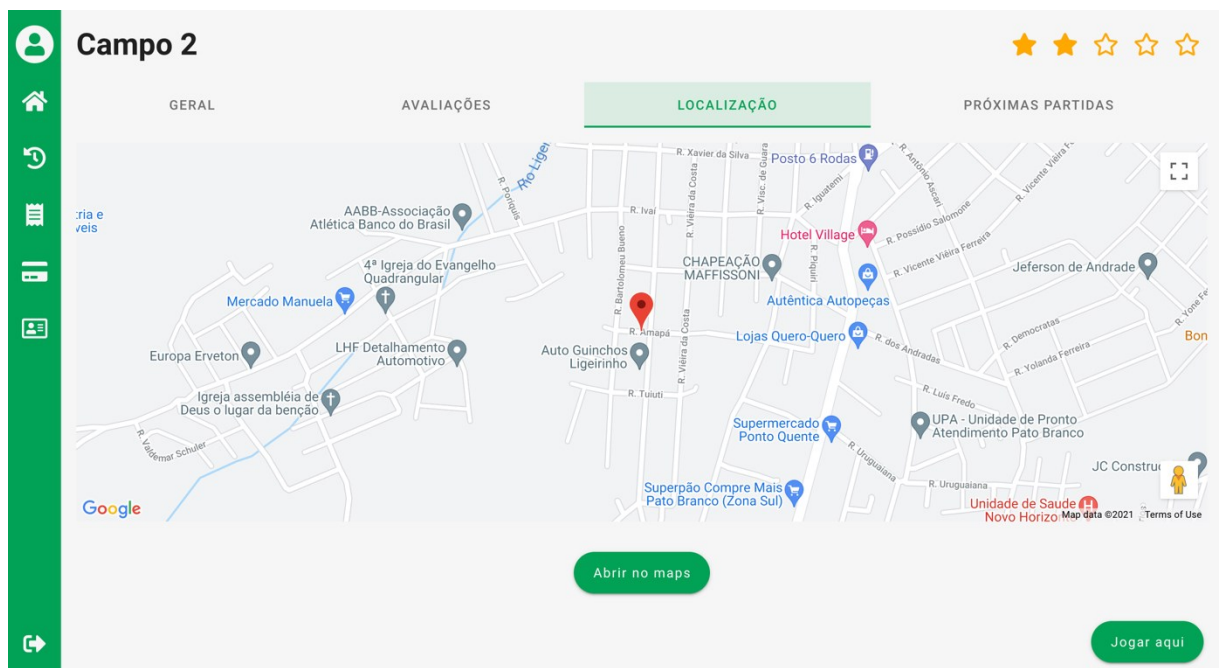
**Figura 39 - Lista de avaliações de um local**



Fonte: Autoria própria.

Na aba “Localização” é exibido o mapa com a localização exata do local, como na Figura 40, e um botão localizado na parte inferior da tela que possibilita a abertura de um aplicativo de mapas para que o usuário busque o melhor caminho até o local com base em sua localização atual.

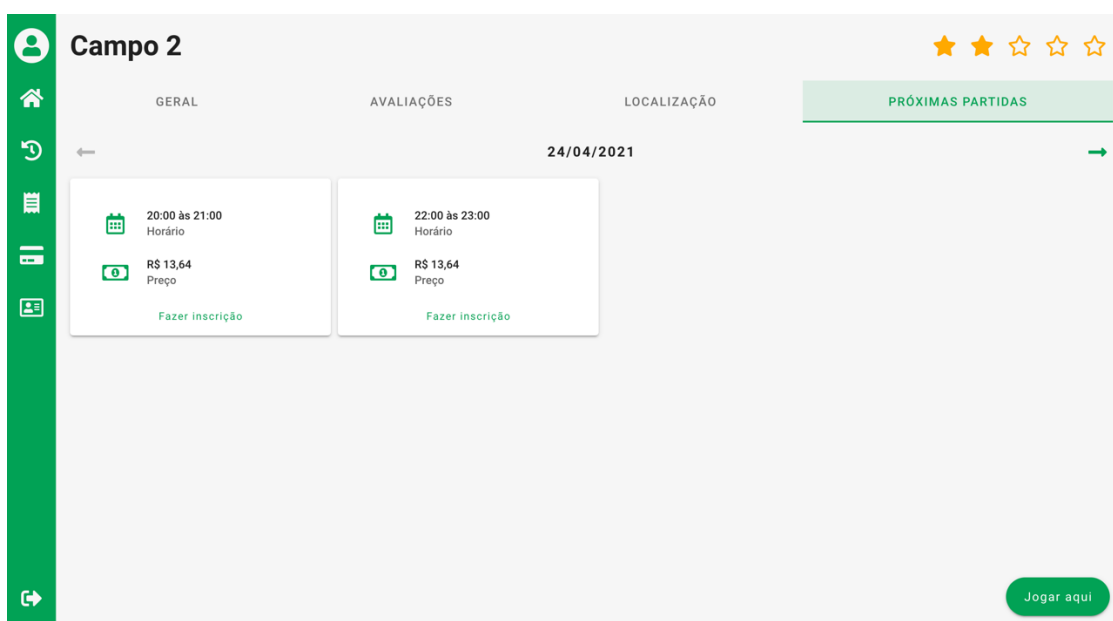
**Figura 40 - Localização de um estabelecimento**



Fonte: Autoria própria.

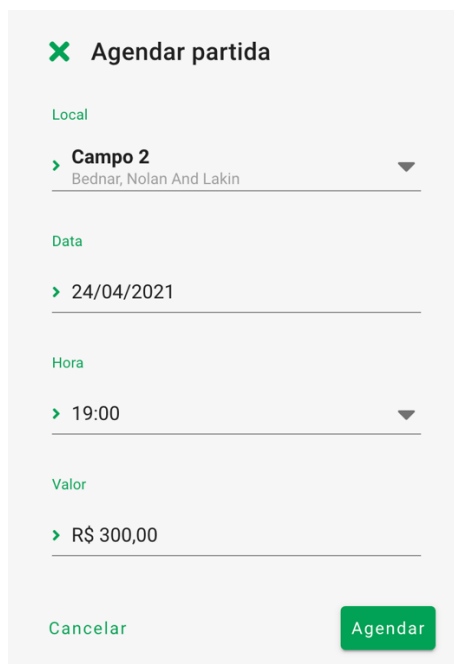
Na aba “Próximas partidas”, ilustrada na Figura 41, são listadas as partidas abertas no local e disponíveis para inscrição. Estas partidas podem ter sido agendadas pela própria empresa ou por outros jogadores. Nesta lista, cada partida é representada por uma opção da lista que contém o preço, data, hora e o botão para realizar a inscrição. Assim como na página de detalhes de uma partida, listada na Figura 37, ao clicar no botão de inscrição o usuário deverá selecionar seu cartão de crédito e confirmar a inscrição.

**Figura 41 - Lista de partidas agendadas no local**



**Fonte: Autoria própria.**

Caso o usuário não encontre uma partida que seja compatível com suas preferências neste local, poderá agendá-la por conta própria clicando no botão “Jogar aqui” que fica localizado a direita inferior da tela de detalhe do local. Este botão é responsável por abrir o formulário de agendamento, como pode ser observado na Figura 42, em que o usuário deverá selecionar a data e hora em que deseja jogar.

**Figura 42 - Formulário de agendamento de partidas**

O formulário, intitulado "Agendar partida", apresenta os seguintes campos e opções:

- Local:** Campo com o texto "Campo 2" e o nome "Bednar, Nolan And Lakin" abaixo dele.
- Data:** Campo com o texto "24/04/2021".
- Hora:** Campo com o texto "19:00".
- Valor:** Campo com o texto "R\$ 300,00".

Na base do formulário, há dois botões: "Cancelar" e "Agendar".

**Fonte: Autoria própria.**

Ao concluir esta etapa, assim como se estivesse realizando uma inscrição em uma partida agendada previamente, basta selecionar seu cartão de crédito e confirmar a inscrição. A partida passará a ser disponibilizada para que outros jogadores cadastrados na plataforma também possam inscrever-se e participar.


Após concluir uma partida, em seu próximo acesso à plataforma, o usuário deve avaliar sua última partida por meio do formulário de avaliação, exibido na Figura 43. Enquanto não completar a avaliação, não poderá utilizar a aplicação. Essa ação é necessária para garantir que todos que estiveram inscritos na partida avaliem e formem a pontuação de reputação do local, cooperando para a escolha de outros jogadores da plataforma e ajudando as empresas melhorarem seus serviços.

**Figura 43 - Avaliação de uma partida**

**Avalie sua última partida!**

Avaliar as partidas que participa lhe darão mais pontos de reputação e permitirão que voce continue jogando.

Qual nota voce daria com base em sua experiencia durante o jogo?



Escreva abaixo o que voce achou!

Mussum Ipsum, cacilds vidis litro abertis.  
Detraxit consequat et quo num tendi nada.

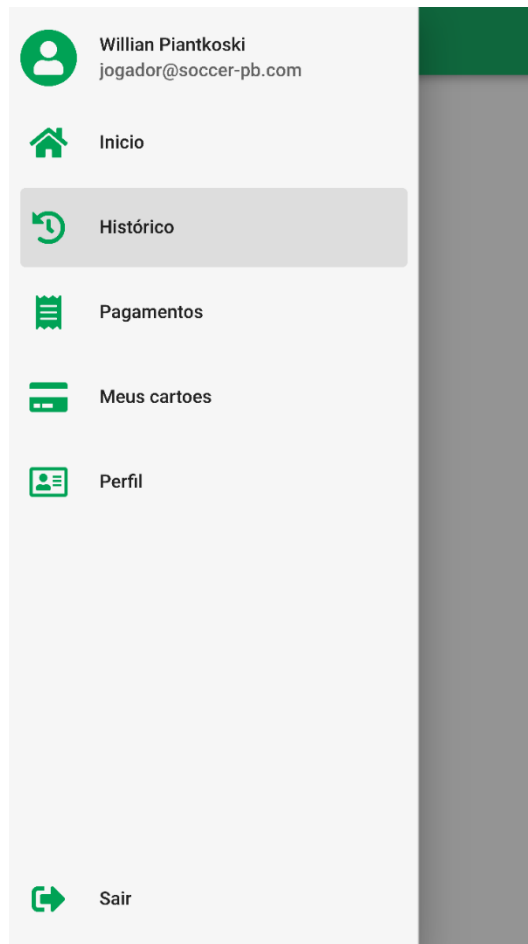
Utilize, no máximo, 150 caracteres

**Fonte: Autoria própria.**

Todo o uso da aplicação para os usuários jogadores gira em torno dos locais e das partidas, no entanto, se acessar o menu, exibido na Figura 44, o usuário encontrará outras opções de acesso que o ajudam a saber mais sobre seus agendamentos e pagamentos.



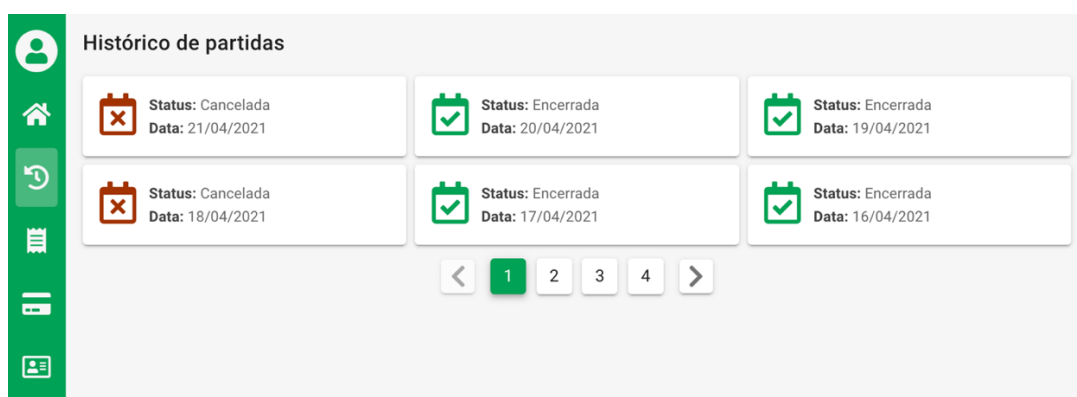
**Figura 44 - Menu para usuários jogadores**



Fonte: Autoria própria.

A opção “Histórico” redirecionará o usuário ao seu histórico de partidas. Nesta lista, ilustrada na Figura 45, cada partida é representada por um *card* que informa qual foi o status final daquele agendamento e a data. Se clicar em algum destes *cards*, o usuário será redirecionado para a tela de detalhes da partida, como já foi exibido na Figura 36.

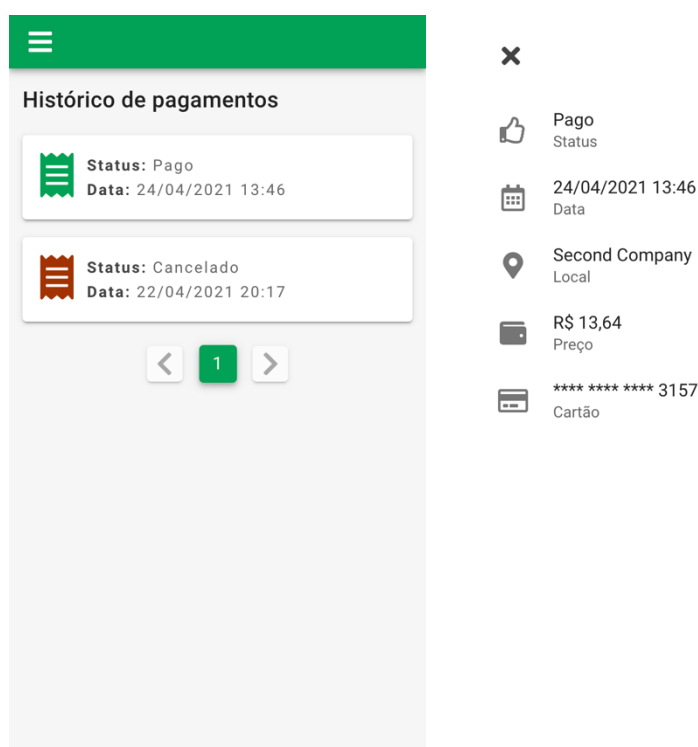
**Figura 45 - Histórico de partidas do usuário**



Fonte: Autoria própria.

A terceira opção do menu, “Pagamentos”, dá acesso a lista de todos os pagamentos realizados pelo usuário na plataforma. Assim como no histórico de partidas, cada pagamento é representado por um *card*, como na Figura 47(a), e contém informações como o status do pagamento, data e hora em que foi realizado. Ao clicar em um item da lista, serão apresentados mais detalhes sobre aquela transação, ilustrados na Figura 47(b).

**Figura 46 - Pagamentos de usuário. a) Lista de pagamentos realizados. b) Detalhes de um pagamento**

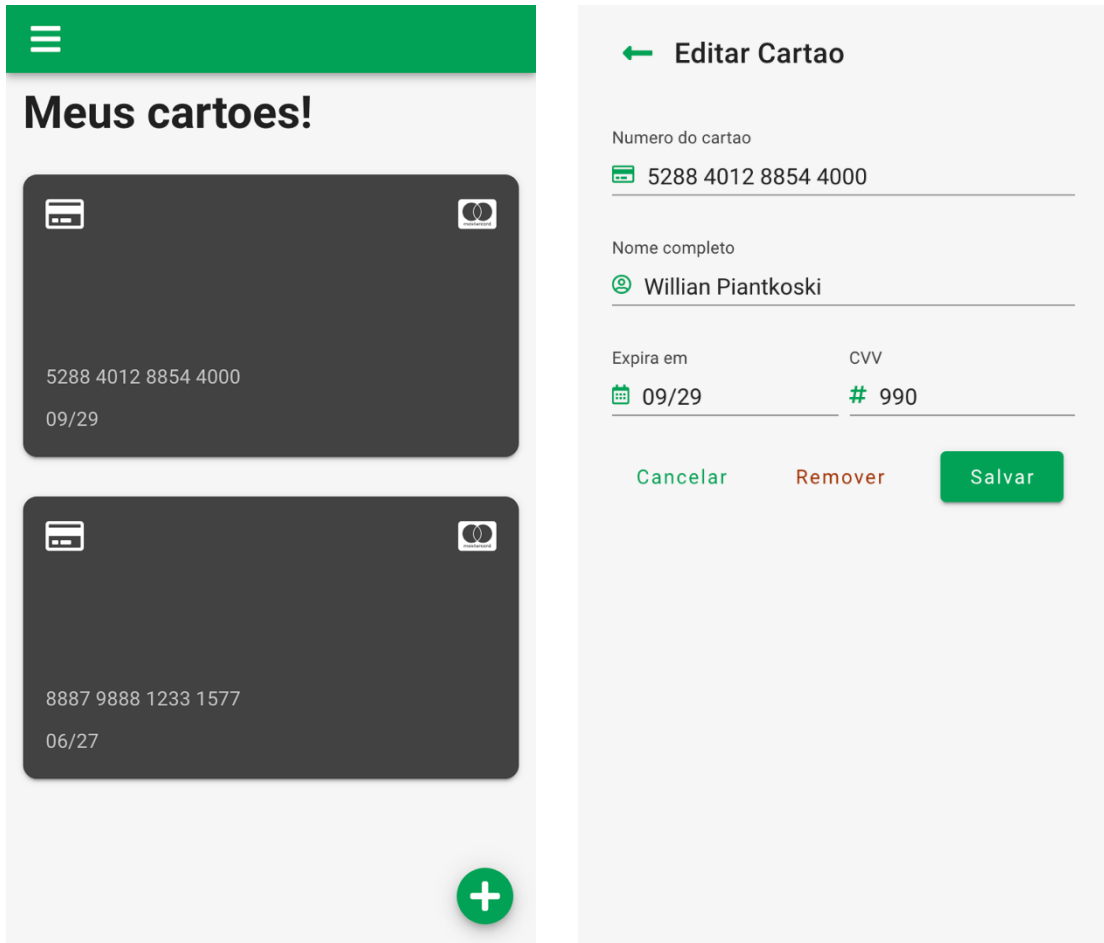


**Fonte: Autoria própria.**

Em casos que uma partida é cancelada de forma automática pela plataforma, por falta de jogadores, pela empresa ou pelo próprio jogador a transação recebe o *status* de “Cancelada” e o reembolso é realizado. A fim de facilitar esta operação de reembolso, a plataforma recebe apenas pagamentos com cartão de crédito.

Continuando com as opções disponíveis no menu, a opção chamada “Meus cartões” redireciona o usuário para sua lista de cartões cadastrados na aplicação, como exibido na Figura 48(a). Cada item da lista representa um cartão de crédito e a manipulação deste pode ser feita por meio do formulário de cartão, ilustrado na Figura 48(b).

Figura 47 - Cartões de crédito. a) Lista de cartões cadastrados. b) Formulário de cartão de crédito

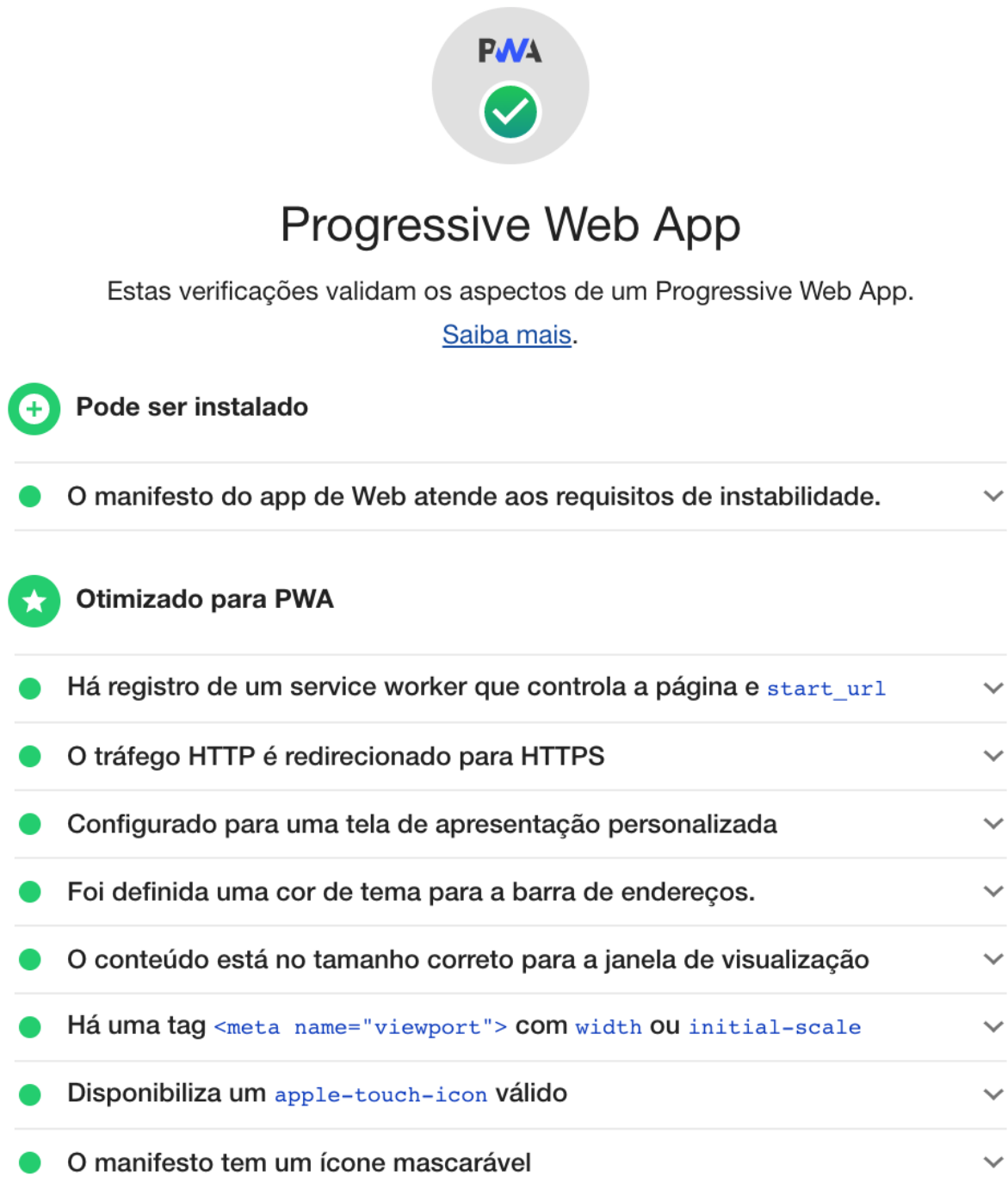


Fonte: Autoria própria.

#### 4.3.5 Progressive Web App

De acordo com o diagnóstico de sites do *Lighthouse*, exibido na Figura 48, ferramenta Google para avaliar a qualidade de páginas web, esta aplicação atende a todos os requisitos que definem uma PWA.

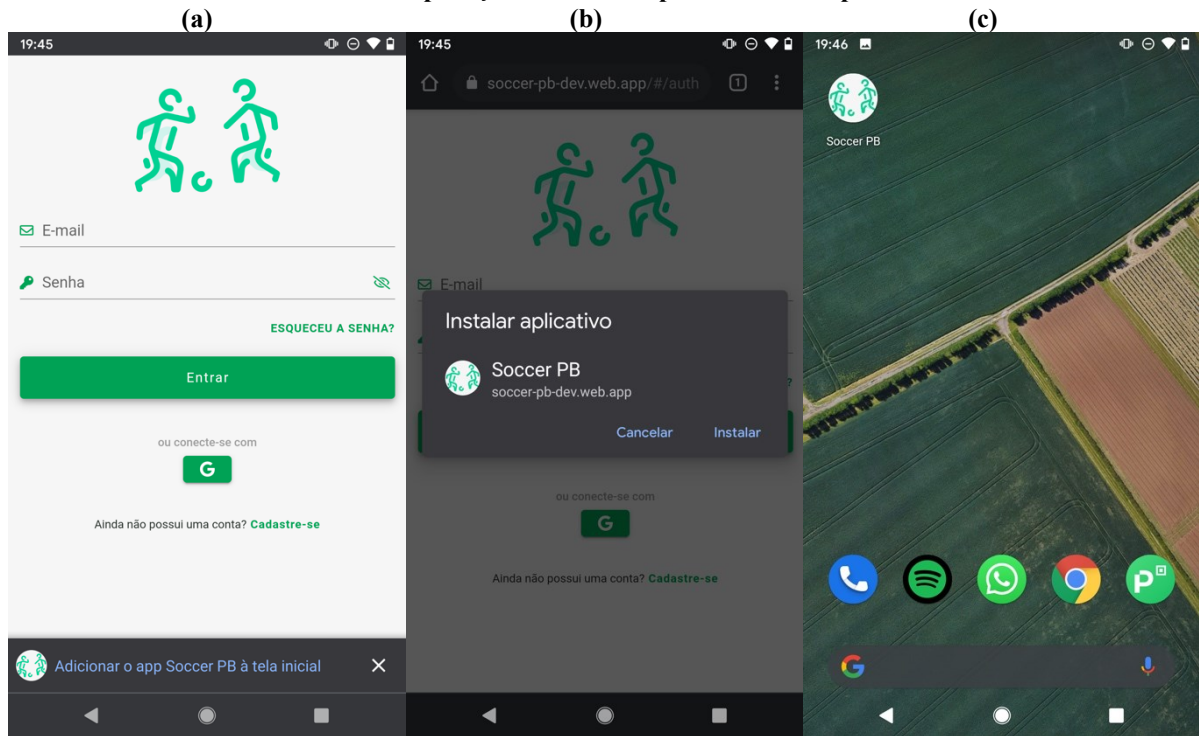
Figura 48 - Relatório de avaliação de sites do *Lighthouse*



Fonte: Autoria própria.

Além de atender aos requisitos, a aplicação também atende às boas práticas de uma PWA, oferecendo melhoria contínua da experiência do usuário durante o uso, como por exemplo, a possibilidade de instalar a aplicação em seu dispositivo Android como se fosse um aplicativo nativo.

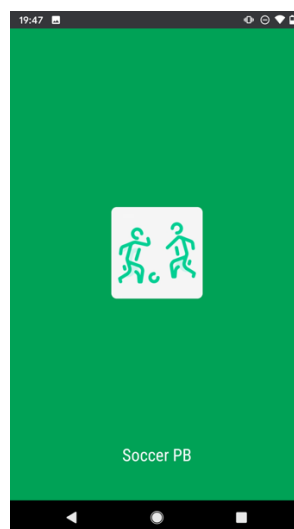
**Figura 49 - Instalação em dispositivo. a) Sugestão de instalação. b) Confirmação de instalação. c) Atalho de acesso à aplicação na lista de aplicativos do dispositivo**



**Fonte: Autoria própria.**

Outra boa prática sugerida no desenvolvimento de uma PWA, a fim de trazer uma experiência mais próxima ao de uma aplicação nativa, é exibido uma tela de abertura que indica ao usuário que sua requisição de abertura da aplicação está sendo processada, como na Figura 50.

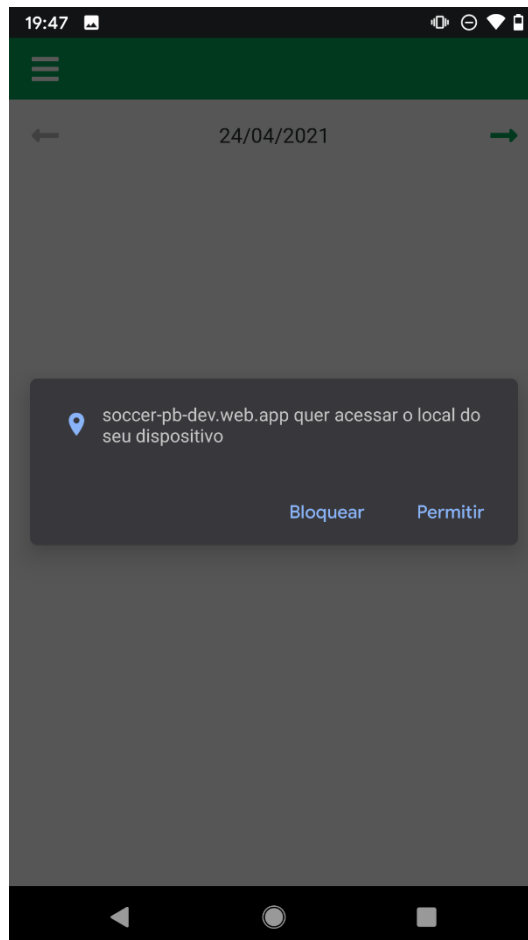
**Figura 50 - Tela de abertura PWA**



**Fonte: Autoria própria.**

Atendendo às necessidades da plataforma, também é preciso fazer o uso da localização atual do usuário. Porém, é boa prática solicitar ao usuário permissão para tal, por isso, como exibido na Figura 51, a aplicação solicita a acesso a localização logo no primeiro acesso do usuário na lista de partidas ou locais abertos. O acesso a localização garante que o cálculo de distância entre o usuário e o local em que deseja jogar seja realizado.

**Figura 51 - Pedido de permissão de acesso a localização do dispositivo**



**Fonte: Autoria própria.**

## 5 CONCLUSÃO

Neste trabalho foi desenvolvida uma aplicação com o intuito de facilitar e incentivar a prática de futebol recreativo e cooperar com empresas que fornecem espaços adequados para a prática na tentativa de diminuir sua ociosidade.

Durante o desenvolvimento algumas escolhas não cooperaram muito para a agilidade e andamento do projeto, como o uso do TDD, que foi descartado. Seu uso em determinado ponto passou a retardar o avanço da implementação pela grande necessidade de código adicional que precisa ser desenvolvido e mantido.

Outro ponto a ser observado ao trabalhar com o *framework frontend* VueJS é a falta de ferramentas *open source* relacionadas às APIs de localização do Google Maps, que carece de apoio da comunidade, bem como a implementação dos recursos de uma PWA. O VueJS fornece ferramentas que automatizam a criação de PWAs, porém, não se adequaram a este projeto, pois foram necessárias diversas adaptações externas ao *framework* para que fosse possível alcançar os objetivos de uma PWA.

Em relação ao funcionamento do conjunto *frontend e backend* a aplicação apresenta um desempenho satisfatório devido ao uso de boas práticas de desenvolvimento, arquitetura e comunicação entre as partes.

Fica claro, levando em consideração o aprendizado adquirido durante a implementação do projeto, que existem diversos pontos de melhorias no projeto, como o uso de dados em tempo real, implementação de tecnologias e ferramentas com foco em geolocalização, um módulo de controle financeiro para empresas e uma funcionalidade que permitisse aos jogadores a criação de times próprios e o envio de convites para amigos dentro da plataforma.

## REFERÊNCIAS

- AMAZON Route53. **AWS**, 2021. Disponível em: <https://aws.amazon.com/pt/route53>. Acesso em: 27 de mar. de 2021.
- ANICHE, Mauricio. **Test-Driven Development teste e design no mundo real**. São Paulo: Casa do Código, 2012.
- ATER, Tal. **Building progressive web apps bringing the power of native to the browser**. California: O'Reilly Media, 2017.
- AWS. O que é Amazon EC2?. **AWS Documentation**, 2021. Disponível em: [https://docs.aws.amazon.com/pt\\_br/AWSEC2/latest/WindowsGuide/concepts.html](https://docs.aws.amazon.com/pt_br/AWSEC2/latest/WindowsGuide/concepts.html). Acesso em: 27 de mar. de 2021.
- CHODOROW, Kristina. **MongoDB the definitive guide**. California: O'Reilly Media, 2013.
- DOGLIO, Fernando. **Pro RESP API development with Node.js**. Uruguai: Apress, 2015.
- DU, Yingsheng. Tang, Youchun. **Study on the development of O2O e-commerce plataforma of China from the perspective of offline servisse quality**. Shanghai, China. International Journal of Business and Social Science, vol.5 n° 4, 2014.
- FILHO, João Carlos Barth. SCHATTENBERG, Lucas Daniel. STOLLMEIER, Nicolas. **Tecnologias Esportivas Auxiliando no Esporte**. Revista Eletrônica do Alto Vale do Itajaí. Brasil. 2013.
- FIREBASE authentication. **Firestore Documentação**, 2021. Disponível em: <https://firebase.google.com/docs/auth>. Acesso em: 27 de mar. de 2021.
- FIREBASE hosting. **Firestore Documentação**, 2021. Disponível em: <https://firebase.google.com/docs/hosting>. Acesso em: 27 de mar. de 2021.
- GLANER, Maria Fátima. **Importância da aptidão física relacionada a saúde**. Revista Brasileira da Cine-antropometria e Desempenho Humano. Distrito Federal, Brasil. 2003.
- GRONER, Loiane. **Estrutura de dados e algoritmos em JavaScript**. São Paulo: Novatec, 2017.
- INCAU, Caio. **Vue.js construa aplicações incríveis**. São Paulo: Casa do Código, 2017.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal**. Brasil. 2016.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Práticas de esporte e atividade física**. Brasil. 2015.
- JACABSON, Daniel. BRAIL, Greg. WOODS, Dan. **APIs a strategy guide**. California: O'Reilly Media, 2012.



MASSE, Mark. **REST API design rulebook**. California: O'Reilly Media, 2011.

LECHETA, Ricardo R. **Node Essencial.1. Ed.** São Paulo: Novatec, 2018.

PONTES, Guilherme. **Progressive web apps construa aplicações progressivas com React**. São Paulo: Casa do Código, 2018.

SABBAGH, Rafael. **Scrum gestão ágil para projetos de sucesso**. São Paulo: Casa do Código, 2013.

SONI, Rahul. **Nginx from beginner to pro**. Nova Iorque: Apress, 2016.

STEFANOV, Stoyan. **Padrões Javascript**. São Paulo: Novatec, 2011.

TIANSHENG, Xu. JIONG, Zhang. **A Development Strategy of O2O Business in China**. Beijing, China. International Conference on Computer Science and Intelligent Communication, 2015.

WILSON, Jim. **Node.js 8 the right way practical, server-side JavaScript that scales**. Pragmatic Bookshelf, 2018.

ZHAO, Jun. MENGYAO, Wang. **Investigation and analysis of the Shenyang local O2O business model**. Shenyang, China. International Conference on Mechatronics, Electronic, Industrial and Control Engineering, 2014.