



MATHEUS DOS ANJOS DE OLIVEIRA

**HOOP TIME: DESENVOLVIMENTO DE UM FRONT-END WEB PARA  
BUSCA DE QUADRAS DE BASQUETE**

CAMPINAS

2025

MATHEUS DOS ANJOS DE OLIVEIRA

**HOOP TIME: DESENVOLVIMENTO DE UM FRONT-END WEB PARA BUSCA DE  
QUADRAS DE BASQUETE**

Trabalho de Conclusão de Curso apresentado  
como parte dos requisitos para obtenção do  
diploma do Curso de Tecnologia em Análise e  
Desenvolvimento de Sistemas do Instituto  
Federal de Educação, Ciência e Tecnologia  
Campus Campinas.

Orientador: Prof. Ms. Emilio Carlos Rodrigues

CAMPINAS  
2025

**Ficha Catalográfica**

Instituto Federal de São Paulo – Campus Campinas

Biblioteca – Pedro Augusto Pinheiro Fantinatti

Rosangela Gomes – CRB 8/8461

Oliveira, Matheus dos Anjos de.

Hoop Time: desenvolvimento de um front-end web para busca de quadras de basquete.  
/ Matheus dos Anjos de Oliveira. – Campinas, SP: [s.n.], 2025.

46 f. : il.

Orientador: Emilio Carlos Rodrigues.

Trabalho de Conclusão de Curso (graduação) – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, 2025.

1. Basquetebol. 2. Espaços esportivos. 3. Tecnologia da informação. 4. Sistemas de informação. 5. Agendamento eletrônico. I. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas. II. Título.

**ATA N.º 3/2025 - TEINFO-CMP/DAE-CMP/DRG/CMP/IFSP**

**Ata de Defesa de Trabalho de Conclusão de Curso - Graduação**

Na presente data, realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulado **HOOP TIME: DESENVOLVIMENTO DE UM FRONT-END WEB PARA BUSCA DE QUADRAS DE BASQUETE**, apresentado(a) pelo(a) aluno(a) **Matheus dos Anjos de Oliveira (CP3020762)** do Curso **SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS** (Campus Campinas). Os trabalhos foram iniciados às **20h37m** pelo(a) Professor(a) presidente da banca examinadora, constituída pelos seguintes membros:

Membros	Instituição	Presença (Sim/Não)
Emilio Carlos Rodrigues (Presidente/Orientador)	IFSP-CMP	SIM
Cecilia Sosa Arias Peixoto	IFSP-CMP	SIM
Antonio Queiroz da Silva Neto	IFSP-CMP	SIM

**Observações:**

A banca examinadora, tendo terminado a apresentação do conteúdo da monografia, passou à arguição do(a) candidato(a). Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo(a) aluno(a), tendo sido atribuído o seguinte resultado:

Aprovado(a)       Reprovado(a)

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu lavrei a presente ata que assino em nome dos demais membros da banca examinadora.

Campus Campinas, 11 de junho de 2025

Documento assinado eletronicamente por:

- **Emilio Carlos Rodrigues, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 11/06/2025 08:52:50.

Este documento foi emitido pelo SUAP em 21/05/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/autenticar-documento/> e forneça os dados abaixo:

**Código Verificador:** 954565  
**Código de Autenticação:** f66ce75bdc



*Dedico este trabalho a todos que respiram basquete  
— entusiastas, amadores e profissionais —  
e compartilham comigo a paixão por esse esporte.*

## **AGRADECIMENTOS**

Agradeço à minha família, por todo o apoio, amor e força nos momentos em que pensei em desistir. Foram muitas dúvidas no caminho, mas o suporte incondicional deles — emocional, financeiro e em todos os sentidos — foi essencial para que eu pudesse continuar e concluir essa etapa.

Agradeço às empresas pelas quais passei durante a formação, que contribuíram não apenas para o meu crescimento profissional, mas também para que eu conseguisse me manter firme até aqui. Foram três anos percorrendo quilômetros, atravessando Campinas para cumprir horários, equilibrando responsabilidades acadêmicas e profissionais.

Agradeço ao meu orientador, Prof. Ms. Emilio Carlos Rodrigues, por sua paciência, orientação e apoio na realização deste trabalho.

Agradeço também a todos os professores, servidores, colegas e amigos do IFSP Campus Campinas, que, de diferentes formas, contribuíram para a minha formação e para a conclusão deste trabalho.

*“Eu posso aceitar a falha, todo mundo falha em alguma coisa.*

*Mas eu não posso aceitar não tentar.”*

*Michael Jordan*

## RESUMO

O basquete, historicamente relegado a segundo plano em relação ao futebol no Brasil, enfrentou desafios significativos em sua infraestrutura, resultando em escassez de espaços adequados para sua prática. Este trabalho teve como objetivo desenvolver um modelo front-end web para mapeamento e agendamento de quadras de basquete. Foram utilizadas tecnologias modernas, como React e TypeScript, com foco em boas práticas de desenvolvimento web. A metodologia adotada baseou-se na prototipagem de interfaces, visando validar e refinar a proposta visual e funcional da aplicação. A plataforma é voltada especialmente a praticantes amadores e entusiastas do basquete, que buscam facilidade para encontrar e reservar quadras disponíveis em suas regiões. Ela permite localizar quadras próximas, visualizar informações detalhadas, realizar agendamentos e compartilhar avaliações, contribuindo para a democratização do acesso ao esporte. A arquitetura do sistema foi projetada de forma modular e escalável, permitindo futura integração com um back-end para expansão das funcionalidades. A implementação de tecnologias digitais mostrou-se uma ferramenta eficaz para superar as barreiras históricas de infraestrutura do basquete no país, oferecendo uma solução prática e acessível que promove o acesso ao esporte e a inclusão social.

**Palavras-chave:** basquetebol; espaços esportivos; tecnologia da informação; sistemas de informação; agendamento eletrônico.

## ABSTRACT

Basketball, historically overshadowed by soccer in Brazil, has faced significant challenges regarding infrastructure, resulting in a lack of suitable spaces for its practice. This project aimed to develop a front-end web model for mapping and scheduling basketball courts. Modern technologies such as React and TypeScript were used, following best practices in web development. The methodology adopted was based on interface prototyping, aiming to validate and refine the visual and functional proposal of the application. The platform targets amateur players and basketball enthusiasts seeking an easier way to find and book available courts in their areas. It allows users to locate nearby courts, view detailed information, make reservations, and share reviews—contributing to the democratization of access to the sport. The system architecture was designed to be modular and scalable, allowing future integration with a back-end to expand its features. The implementation of digital technologies proved to be an effective tool to overcome the historical infrastructural barriers of basketball in the country, offering a practical and accessible solution that fosters both access to the sport and social inclusion.

**Keywords:** basketball; sports spaces; information technology; information systems; electronic scheduling.

## LISTA DE FIGURAS

Figura 1 – Página de exibição de quadras do Playfinder.....	20
Figura 2 – Tela de agendamento das quadras do aplicativo 156+POA.....	21
Figura 3 – Protótipo da página inicial.....	24
Figura 4 – Protótipo da página de <i>login</i> .....	25
Figura 5 – Protótipo da página de registro.....	25
Figura 6 – Protótipo da página de lista de quadras.....	26
Figura 7 – Protótipo da página de quadra.....	26
Figura 8 – Protótipo da página de mapa de quadras.....	27
Figura 9 – Diagrama de caso de uso.....	28
Figura 10 – Diagrama de Atividade.....	29
Figura 11 – Diretórios e Arquivos Principais.....	32
Figura 12 – Página inicial.....	38
Figura 13 – Página de <i>login</i> .....	39
Figura 14 – Página de registro.....	40
Figura 15 – Página de lista de quadras.....	40
Figura 16 – Página de detalhes da quadra.....	41
Figura 17 – Página de mapa.....	42
Figura 18 – Página de erro.....	42

## **LISTA DE QUADROS**

Quadro 1 – Requisitos funcionais.....	31
Quadro 2 – Requisitos não funcionais.....	31
Quadro 3 – Trecho de código da página de lista de quadras.....	33
Quadro 4 – Trecho de código da página de detalhes da quadra.....	34
Quadro 5 – Trecho de código da página de <i>login</i> .....	36

## **LISTA DE SIGLAS**

API	Application Programming Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
GUI	Graphical User Interface
HTML	HyperText Markup Language
JS	JavaScript
SEO	Search Engine Optimization
SPA	Single Page Application
UML	Unified Modeling Language
URL	Uniform Resource Locator

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>14</b>
<b>2 JUSTIFICATIVA</b>	<b>16</b>
<b>3 OBJETIVO</b>	<b>18</b>
3.1 Objetivo Geral	18
3.2 Objetivos Específicos	18
<b>4 FUNDAMENTAÇÃO TEÓRICA</b>	<b>19</b>
4.1 O Gerenciamento como diferencial na gestão de quadras esportivas	19
4.2 Soluções Tecnológicas no Esporte	19
4.3 Website Playfinder	19
4.4 Aplicativo 156+POA	20
4.5 Ferramentas de Desenvolvimento Front-end	21
4.5.1 <i>React</i>	21
4.5.2 <i>TypeScript</i>	22
4.5.3 <i>Styled Components</i>	22
4.5.4 <i>React Router</i>	22
4.5.5 <i>HTML</i>	22
4.5.6 <i>Node.js</i>	23
4.5.7 <i>NPM</i>	23
4.5.8 <i>Git</i>	23
4.5.9 <i>GitHub</i>	23
4.5.10 <i>Excalidraw</i>	24
<b>5 METODOLOGIA</b>	<b>28</b>
5.1 Diagramas UML	28
5.2 Requisitos	30
5.2.1 <i>Requisitos funcionais</i>	30
5.2.2 <i>Requisitos não funcionais</i>	31
5.3 Arquitetura e estrutura do projeto	32
5.3.1 <i>Estrutura do código da página de lista das quadras</i>	32
5.3.2 <i>Estrutura do código da página de detalhes da quadra</i>	34
5.3.3 <i>Estrutura do código da página de login</i>	35
<b>6 RESULTADOS</b>	<b>38</b>
6.1 Página Inicial	38
6.2 Página de login	39
6.3 Página de Registro	39
6.4 Página de Lista de Quadras	40
6.5 Página de Detalhes da Quadra	41
6.6 Página de Mapa	41
6.7 Página de Tratamento de Erro	42

<b>7 CONSIDERAÇÕES FINAIS</b>	<b>44</b>
<b>REFERÊNCIAS</b>	<b>45</b>

## 1 INTRODUÇÃO

O basquete no Brasil possui uma trajetória paradoxal: embora seja um esporte com expressiva tradição, sua democratização sempre esbarrou em barreiras históricas e estruturais. Desde sua introdução em 1894, o esporte foi associado a elites urbanas, praticado em "clubes privados, fechados às categorias populares" (Gaudin, 2007), enquanto o futebol se consolidava como fenômeno nacional. Esse cenário perpetuou uma lacuna de acesso: hoje, quadras públicas em regiões carentes frequentemente apresentam infraestrutura precária ou são subutilizadas por falta de visibilidade (Taheiji; Noda; Sales, 2022).

A falta de mapeamento e gestão integrada das quadras públicas e privadas, em várias localidades, dificulta o acesso da população. Nesse contexto, a tecnologia tem se mostrado uma ferramenta poderosa para democratizar o esporte, proporcionando acesso facilitado à informação e agendamento eficiente de espaços para a prática esportiva. O papel da tecnologia nos esportes mudou significativamente desde o início do século XXI, alinhado aos avanços tecnológicos e à digitalização de diversas indústrias (Frevel; Beiderbeck; Schmidt, 2022).

Este trabalho nasce da necessidade de transformar essa situação. O Hoop Time surge como uma plataforma digital que visa conectar jogadores às quadras de basquete disponíveis em suas cidades, rompendo com as barreiras históricas que sempre dificultaram o acesso ao esporte. Mais do que um simples sistema de busca, a plataforma pretende criar uma comunidade em torno do basquete, incentivando a prática regular e fortalecendo a cultura esportiva local.

A importância deste projeto vai além da conveniência tecnológica. Ao facilitar o encontro entre jogadores e espaços esportivos, o Hoop Time pode se tornar uma ferramenta de inclusão social, especialmente para jovens em áreas com poucas opções de lazer e desenvolvimento. A plataforma foi pensada para ser simples e acessível, garantindo que qualquer pessoa, independentemente de seu conhecimento tecnológico, possa encontrar locais para jogar basquete em sua região.

Nos próximos capítulos, serão apresentados os fundamentos deste trabalho. Primeiro, exploraremos as razões que justificam o desenvolvimento deste trabalho, mostrando como a falta de informações centralizadas sobre quadras prejudica o desenvolvimento do basquete no Brasil. Em seguida, detalharemos os objetivos do trabalho, tanto em sua versão atual quanto em suas possibilidades futuras.

A fundamentação teórica trará dados e pesquisas que comprovam a importância de iniciativas como esta, enquanto a metodologia explicará como o trabalho foi desenvolvido para garantir usabilidade e eficiência. Por fim, serão apresentados os resultados alcançados e como eles podem impactar positivamente a vida de milhares de praticantes de basquete em todo o país.

## 2 JUSTIFICATIVA

Embora o basquete tenha sido introduzido no Brasil no mesmo ano que o futebol (1894), sua trajetória foi marcada por uma assimilação lenta e elitizada. Enquanto o futebol se popularizou rapidamente como um esporte das massas, o basquete permaneceu restrito a clubes privados e classes médias urbanas, especialmente no eixo Rio-São Paulo (Gaudin, 2007). Essa discrepância histórica reflete-se ainda hoje na falta de quadras públicas acessíveis.

A precariedade na infraestrutura esportiva é um problema recorrente em muitas cidades brasileiras. A prática de atividades físicas e esportes é essencial para o bem-estar humano, mas frequentemente é dificultada por falta de políticas públicas adequadas e infraestrutura suficiente, especialmente em áreas urbanas. Essa falta de estrutura impacta diretamente a democratização do esporte, restringindo sua acessibilidade (Pellegrinotti, 2012).

Por outro lado, o uso de tecnologias tem se mostrado uma ferramenta eficaz para melhorar o acesso e a gestão de espaços esportivos. Plataformas digitais de agendamento e controle de quadras esportivas não apenas otimizam a utilização dos espaços, mas também facilitam a comunicação e a interação entre usuários e administradores. A tecnologia pode transformar a maneira como os esportes são praticados e gerenciados, aumentando sua eficiência e acessibilidade (Frevel; Beiderbeck; Schmidt, 2022).

Iniciativas como o aplicativo desenvolvido pela prefeitura de Porto Alegre para agendamento de quadras na Orla do Guaíba têm demonstrado resultados positivos em termos de adesão e satisfação dos usuários, evidenciando o potencial das soluções tecnológicas na gestão esportiva urbana (Martins; Dotto; Myskiw, 2023). Inspirado por esses exemplos, este trabalho busca criar uma plataforma que mapeie quadras públicas e privadas e permita o agendamento de horários. Essa solução visa não apenas facilitar o acesso aos espaços esportivos, mas também promover maior integração social e esportiva na cidade.

Vale destacar que a desigualdade no acesso ao basquete vai além da infraestrutura física. O esporte carrega uma carga simbólica de "modernidade" e "status" que acaba excluindo parcelas da população. A falta de referências acessíveis - como quadras em espaços públicos - reforça a ideia de que o basquete não é um esporte para todos. Essa percepção precisa ser combatida com ações concretas que tornem o esporte visível e acessível em todas as comunidades (Gaudin, 2007).

Essa solução visa não apenas facilitar o acesso aos espaços esportivos, mas também promover maior integração social e esportiva na cidade, rompendo com o histórico de elitização que marcou o basquete brasileiro desde suas origens.

### 3 OBJETIVO

#### 3.1 Objetivo Geral

Este trabalho tem como objetivo desenvolver uma plataforma *web front-end* voltada para facilitar o acesso à prática do basquete, oferecendo uma solução digital inovadora para jogadores e entusiastas. A aplicação será focada em proporcionar uma visualização clara e intuitiva das quadras públicas e privadas disponíveis em determinada região, permitindo que os usuários localizem facilmente os melhores locais para jogar. Através de um design responsivo e acessível, a plataforma garantirá uma boa experiência tanto em dispositivos móveis quanto em computadores, tornando o basquete mais acessível para todos.

Um dos principais recursos será a simulação de agendamento para quadras privadas, criando um sistema preparado para futura integração com um back-end completo. Os usuários poderão visualizar horários e simular reservas, enquanto a interface oferecerá filtros inteligentes para buscar quadras por localização e outras características relevantes. Além disso, a plataforma incluirá um sistema de avaliações e comentários, onde a comunidade poderá compartilhar experiências e recomendações sobre as quadras.

#### 3.2 Objetivos Específicos

Dessa forma, para atingir o objetivo proposto, este trabalho foi estruturado em metas específicas que orientaram o desenvolvimento da solução. A seguir, são apresentados os objetivos específicos do trabalho:

- a) Criar uma interface *web* responsiva e intuitiva que permita aos usuários visualizarem a localização de quadras públicas de basquete, com base em dados estáticos;
- b) Implementar uma simulação de agendamento para quadras privadas, permitindo ao usuário selecionar horários disponíveis;
- c) Exibir detalhes relevantes sobre cada quadra, como nome, endereço, avaliação, comentários e imagem da localização;
- d) Permitir a filtragem visual das quadras por critérios como bairro, tipo (pública ou privada), relevância, alfabética e avaliação;
- e) Utilizar recursos modernos de desenvolvimento *front-end* com React, aplicando boas práticas de organização, responsividade e usabilidade;
- f) Desenvolver uma funcionalidade de agendamento para quadras privadas, utilizando dados simulados para representar o fluxo de uso.

## 4 FUNDAMENTAÇÃO TEÓRICA

### 4.1 O Gerenciamento como diferencial na gestão de quadras esportivas

A gestão eficiente de quadras esportivas, especialmente em cidades com alta densidade populacional, é crucial para maximizar seu uso e acessibilidade. O uso de ferramentas tecnológicas para o gerenciamento de quadras esportivas é um diferencial que facilita o agendamento, melhora a experiência do usuário e otimiza a administração do espaço. Essas ferramentas permitem que administradores monitorem reservas, recebam feedbacks dos usuários e ajustem a oferta de acordo com a demanda. Entretanto, é importante reconhecer que as quadras podem funcionar sem sistemas de gerenciamento automático.

### 4.2 Soluções Tecnológicas no Esporte

O objetivo de tecnologias como as propostas neste projeto é agregar valor, tornando os processos mais acessíveis e eficientes, tanto para gestores quanto para os usuários finais.

### 4.3 *Website Playfinder*

Projetos semelhantes, como o Playfinder (Reino Unido), já conectam milhões de usuários a quadras esportivas, com reservas em tempo real e sistemas de avaliação. No Reino Unido, o Playfinder já disponibiliza mais de 7.000 instalações em sua plataforma (Bookteq, 2025). A figura 1 a seguir apresenta a página de exibição de quadras do *website*.

Figura 1 – Página de exibição de quadras do Playfinder

The screenshot shows the Playfinder website interface. At the top, there's a navigation bar with the Playfinder logo, a search icon, and links for 'MY ACCOUNT' and 'LIST YOUR VENUE'. Below the header, a blue banner says 'List your venue' and 'Connect your sport facilities with your local community! Learn more ». There's also a link 'VIEW MAP'.

In the main content area, a message states 'There are 18 places to play basketball within a 5 mile radius of your desired location'. A button 'VIEW MAP' is located to the right.

On the left, there are several filter options:

- Booking method:** Includes 'Playfinder Partners' and 'Playfinder Partners with Online Booking'.
- Formats:** Includes 'court', 'half court', 'indoor', and 'outdoor'.
- Surfaces:** Includes 'Astroturf', 'Hard (macadam)', 'Indoor', and 'Sports hall'.
- Radius:** Includes 'Within 1 mile', 'Within 3 miles', 'Within 5 miles' (which is selected), 'Within 10 miles', and 'Within 75 miles'.
- Facilities:** Includes 'Changing rooms'.

Two venue cards are displayed under the heading 'Playfinder Partner':

- Haberdashers' Hatcham College - Jerningham Site**: Located at Jerningham Road, London, London, SE14 5NY. Format: outdoor. Surface: Hard (Macadam). It is 4.4 miles away and has free parking. A green button 'EXPLORE VENUE' is available.
- Haberdashers' Hatcham College - Jerningham Site**: Located at Jerningham Road, London, London, SE14 5NY. Format: outdoor. Surface: Hard (Macadam). It is 4.4 miles away and has free parking. A purple button 'BOOK ONLINE' is available.

At the bottom of the page, there's a purple bar with the text 'Get more players'.

Fonte: Bookteq (2025)

#### 4.4 Aplicativo 156+POA

Em Porto Alegre, a Prefeitura disponibiliza o aplicativo 156+POA, que permite o agendamento de quadras esportivas na Orla do Guaíba, facilitando o acesso e a gestão desses espaços públicos. Essa iniciativa tem sido eficaz para aumentar a adesão da população às atividades esportivas e promover o uso adequado das quadras. O aplicativo está disponível para Android e também pode ser acessado pelo site oficial da Prefeitura (Prefeitura de Porto Alegre, 2021). A seguir, é apresentada pela figura 2, a tela de exibição de quadras públicas no aplicativo.

Figura 2 – Tela de agendamento das quadras do aplicativo 156+POA



Fonte: Prefeitura de Porto Alegre (2025)

#### 4.5 Ferramentas de Desenvolvimento *Front-end*

O termo *front-end* se refere à interface gráfica do usuário (GUI) com a qual as pessoas interagem, como menus de navegação, elementos de *design*, botões, imagens e gráficos. Em termos técnicos, uma tela ou página que o usuário vê com vários componentes da interface do usuário é chamada de modelo de objeto de documento (DOM) (Amazon Web Services, 2023).

##### 4.5.1 React

O React é uma biblioteca JavaScript amplamente utilizada para a construção de interfaces de usuário interativas. A versão 18 do React trouxe melhorias no gerenciamento de estado e renderização, permitindo a criação de uma aplicação mais responsiva e eficiente (Meta Open Source, [s.d.]).

Neste trabalho o React foi implementado como a biblioteca principal para construção da interface do usuário, sendo utilizado na criação de diversos componentes reutilizáveis. A biblioteca permitiu o gerenciamento do estado dos componentes e a implementação de uma SPA com múltiplas rotas.

#### **4.5.2 TypeScript**

TypeScript é um superset do JavaScript que adiciona tipagem estática ao código, o que ajudou a evitar erros durante o desenvolvimento e a tornar o código mais legível e fácil de manter. Com a tipagem forte, conseguimos garantir maior confiabilidade nas funções e componentes do sistema, além de melhorar a experiência de desenvolvimento (Microsoft, 2025).

Foi utilizado na definição de interfaces para *props* de componentes, na tipagem de variáveis e funções, e na criação de tipos personalizados para o tema do Styled Components.

#### **4.5.3 Styled Components**

Styled Components é uma biblioteca para React (e React Native) que permite estilizar componentes React utilizando CSS in JS (CSS dentro do JavaScript). Essa abordagem usa literais de modelo marcados para definir estilos, oferecendo vantagens como estilos escopados, reutilização de estilos e uma sintaxe mais expressiva, segundo a documentação (Styled Components, [s.d.]).

Foi implementado um sistema de temas através do arquivo theme.ts, possibilitando estilização responsiva com media queries, estilização baseada em *props* e criação de animações CSS.

#### **4.5.4 React Router**

O React Router é uma biblioteca JavaScript utilizada em aplicações React para gerenciar o roteamento do lado do cliente. Ele permite que os usuários naveguem por diferentes visualizações de uma aplicação, como se estivessem em um site tradicional, sem a necessidade de recarregar a página inteira. Em outras palavras, o React Router permite que você construa aplicações de página única (SPAs) com navegação dinâmica, onde diferentes componentes são renderizados com base na URL (Remix Software, [s.d.]).

A biblioteca foi utilizada para criar links de navegação, manipular parâmetros de URL e implementar rotas protegidas e redirecionamentos.

#### **4.5.5 HTML**

HTML é a linguagem de marcação de hipertexto que define o significado e a estrutura do conteúdo da web. É o bloco de construção mais básico da internet, utilizado para

estruturar páginas web com elementos como parágrafos, listas, imagens e tabelas (Mozilla Foundation, 2025).

Foi empregado na criação de formulários para entrada de dados, elementos de mídia e imagens, e na organização hierárquica do conteúdo.

#### **4.5.6 Node.js**

Node.js é um ambiente de tempo de execução JavaScript (runtime) de código aberto e multiplataforma, que permite executar código JavaScript fora de um navegador *web*. Isso significa que, com Node.js, você pode usar JavaScript para criar servidores, ferramentas de linha de comando, aplicações *web* e muito mais (OpenJS Foundation, [s.d.]).

Sua implementação foi crucial para suportar o ecossistema de pacotes npm necessários para o projeto.

#### **4.5.7 NPM**

O npm, ou Node Package Manager, é um gerenciador de pacotes e um registro online para o ambiente Node.js. Ele é essencial para o desenvolvimento Node.js, permitindo a instalação, atualização e gerenciamento de dependências (pacotes) para projetos (Npm, [s.d.]).

Através do package.json, foram definidos *scripts* para execução de tarefas de desenvolvimento e *build*, além do controle de versões das dependências.

#### **4.5.8 Git**

O Git é um sistema de controle de versão distribuído, gratuito e de código aberto, projetado para rastrear alterações em arquivos e facilitar a colaboração em projetos. Ele permite que desenvolvedores acompanhem a evolução de um projeto, registrando as alterações realizadas em seus arquivos ao longo do tempo, o que facilita a reversão para versões anteriores e a integração de código entre diferentes desenvolvedores (Git, [s.d.]).

#### **4.5.9 GitHub**

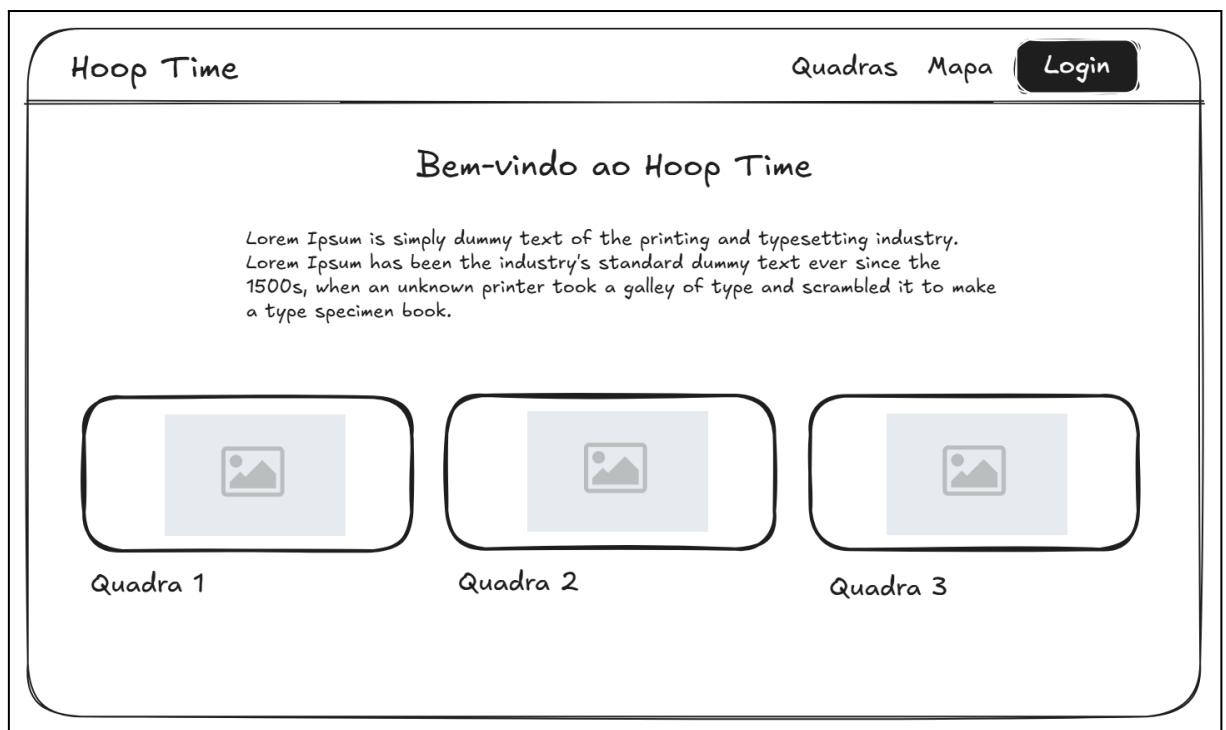
O GitHub é uma plataforma de hospedagem de código-fonte que permite colaboração entre desenvolvedores por meio do controle de versão utilizando o Git. Ele oferece ferramentas para revisão de código, gerenciamento de projetos e integração contínua, sendo amplamente utilizado para o desenvolvimento de software colaborativo e open source. Além disso, o GitHub permite criar repositórios públicos e privados, facilitando o trabalho em equipe e o versionamento de projetos (GitHub, 2025).

#### 4.5.10 Excalidraw

O Excalidraw é uma ferramenta de código aberto para criação de diagramas desenhados à mão de forma intuitiva. Desenvolvida com foco na simplicidade e usabilidade, permite aos usuários criarem representações visuais. Ele funciona diretamente no navegador e possui suporte a compartilhamento em tempo real e exportação em diversos formatos. É uma alternativa prática para criação rápida de esboços visuais, especialmente útil em projetos de software e design de interfaces (Excalidraw, 2025).

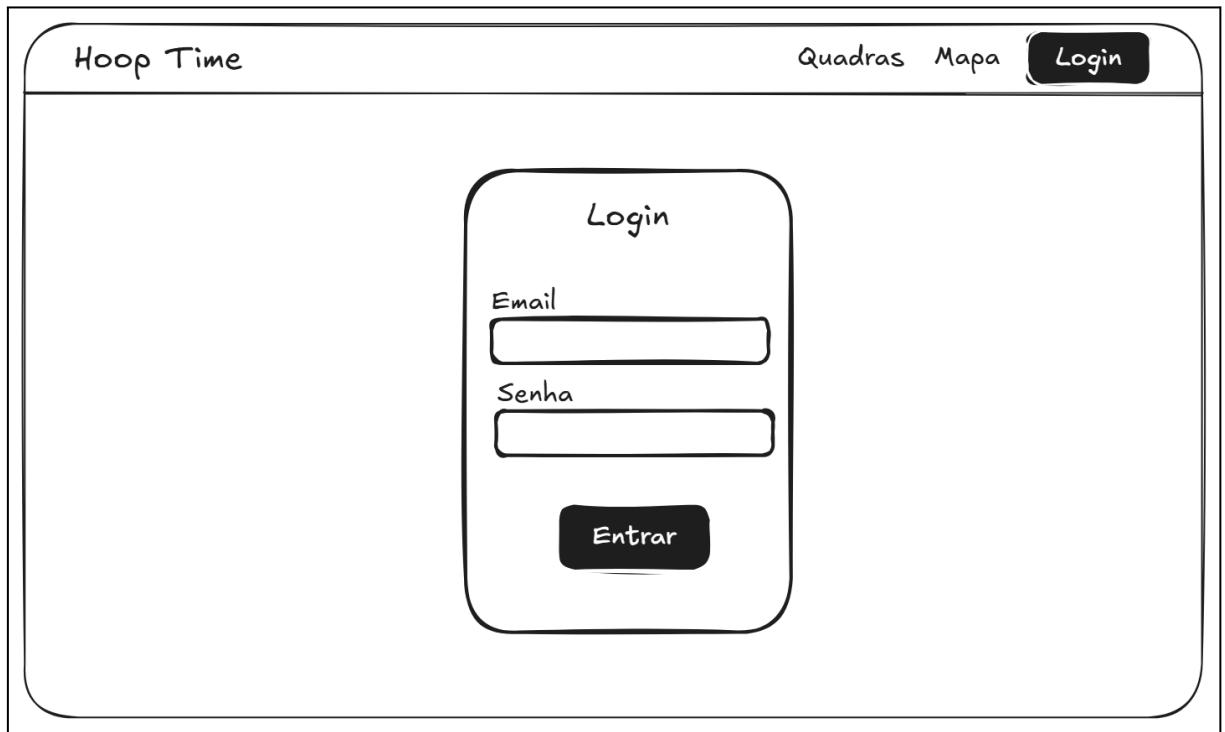
Para a interface deste projeto foram desenvolvidos os protótipos apresentados a seguir nas figuras de 3 a 8.

Figura 3 – Protótipo da página inicial



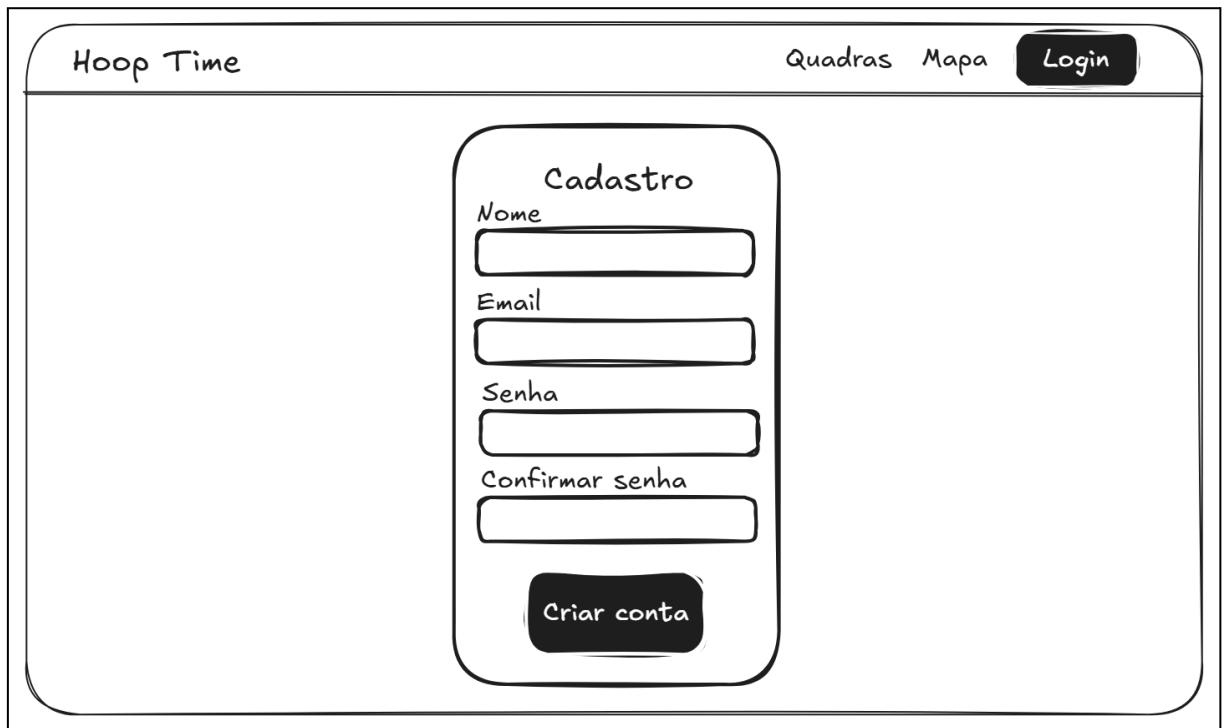
Fonte: Elaborado pelo autor (2025)

Figura 4 – Protótipo da página de login



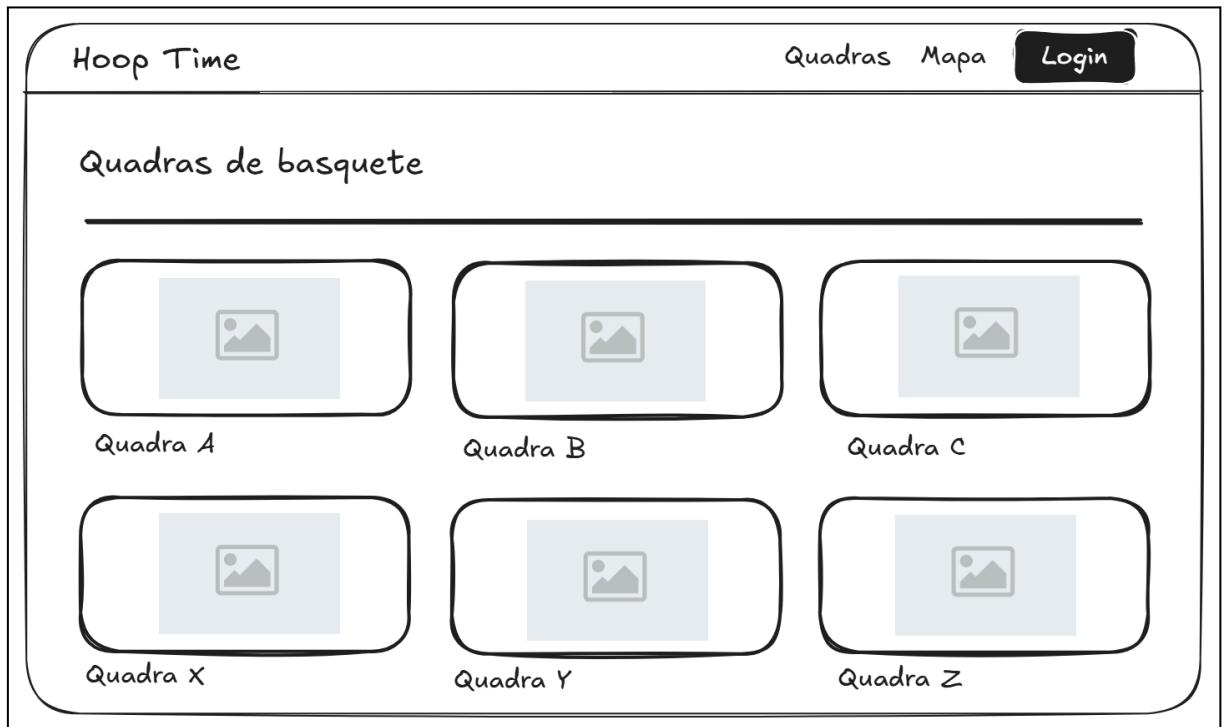
Fonte: Elaborado pelo autor (2025)

Figura 5 – Protótipo da página de registro



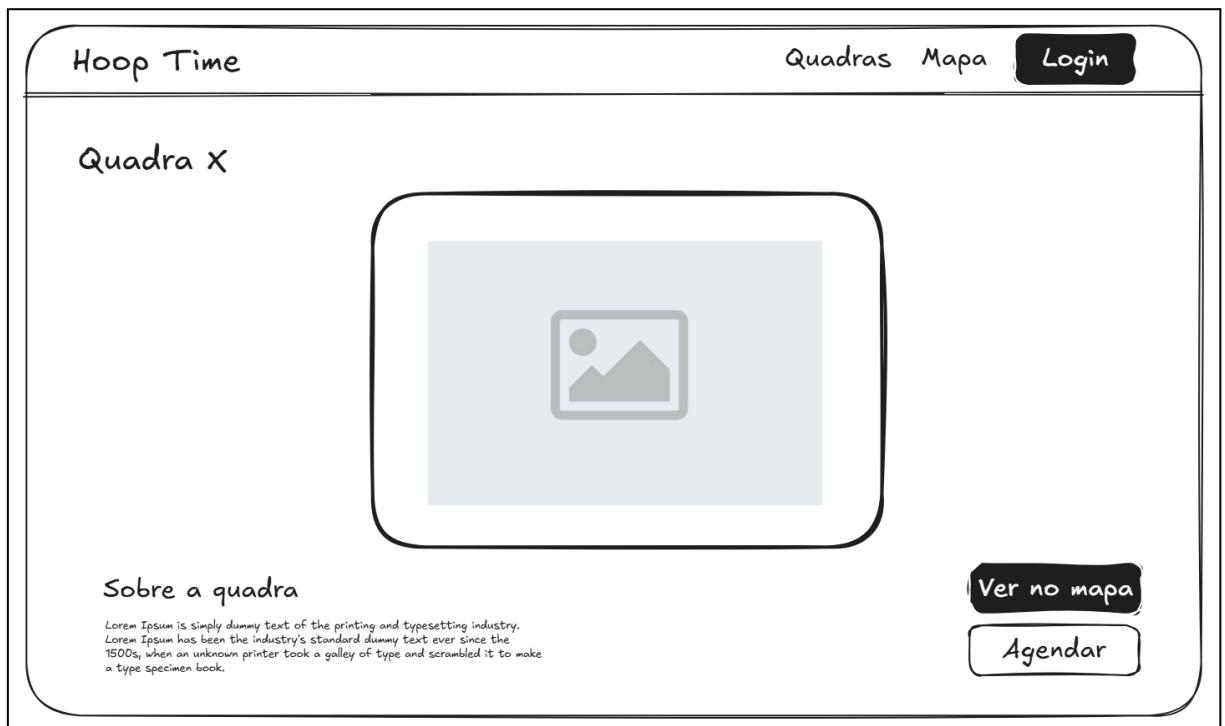
Fonte: Elaborado pelo autor (2025)

Figura 6 – Protótipo da página de lista de quadras



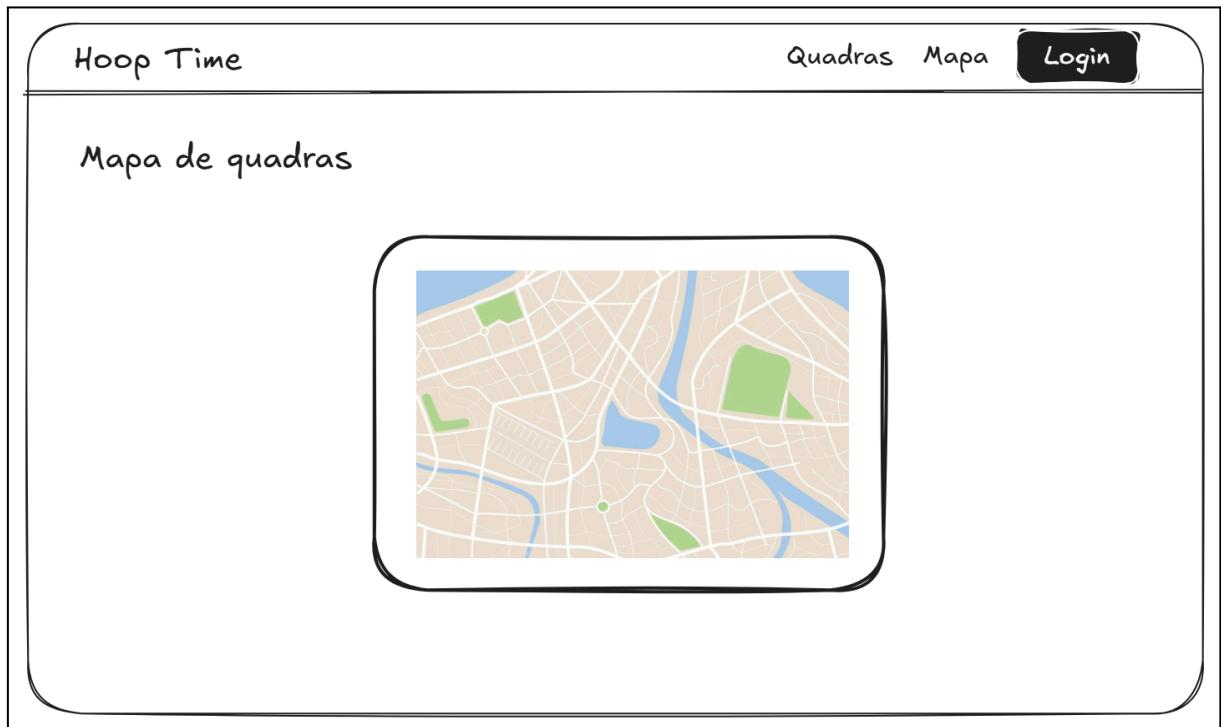
Fonte: Elaborado pelo autor (2025)

Figura 7 – Protótipo da página de quadra



Fonte: Elaborado pelo autor (2025)

Figura 8 – Protótipo da página de mapa de quadras



Fonte: Elaborado pelo autor (2025)

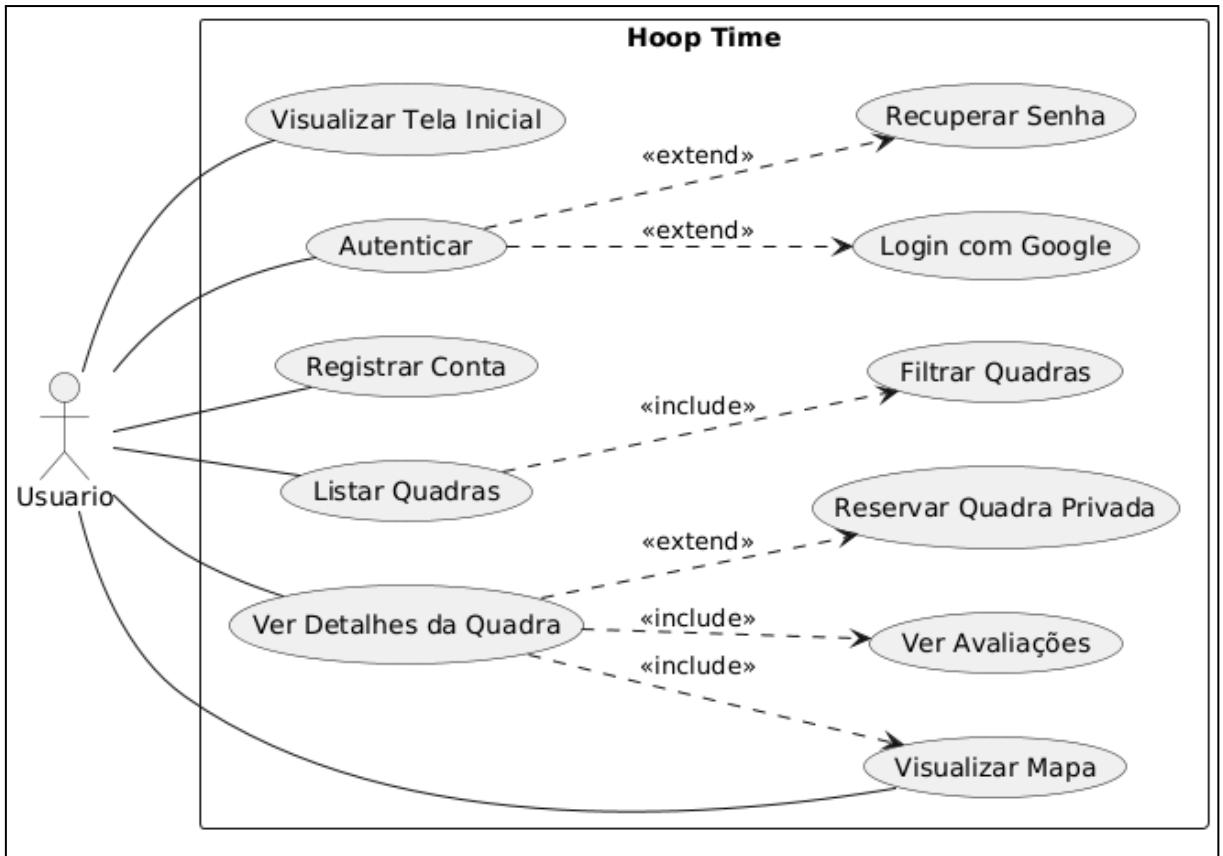
Os protótipos desenvolvidos no Excalidraw serviram como base fundamental para o processo de desenvolvimento da interface, permitindo a validação prévia dos conceitos de usabilidade e organização visual antes da implementação em código. Esta etapa de prototipagem é essencial para definir a disposição dos elementos nas páginas, estabelecer a hierarquia de informações e garantir a consistência visual entre as diferentes telas do sistema. A simplicidade da ferramenta facilitou iterações rápidas no design, contribuindo para um processo de desenvolvimento mais eficiente e estruturado, onde as decisões de interface foram tomadas de forma consciente.

## 5 METODOLOGIA

### 5.1 Diagramas UML

Foi desenvolvido um diagrama de caso de uso, representado na figura 9 a seguir, que ilustra as interações do ator Usuário com o sistema. Esse ator representa qualquer pessoa que acessa a plataforma com o objetivo de localizar e consultar informações sobre as quadras de basquete disponíveis.

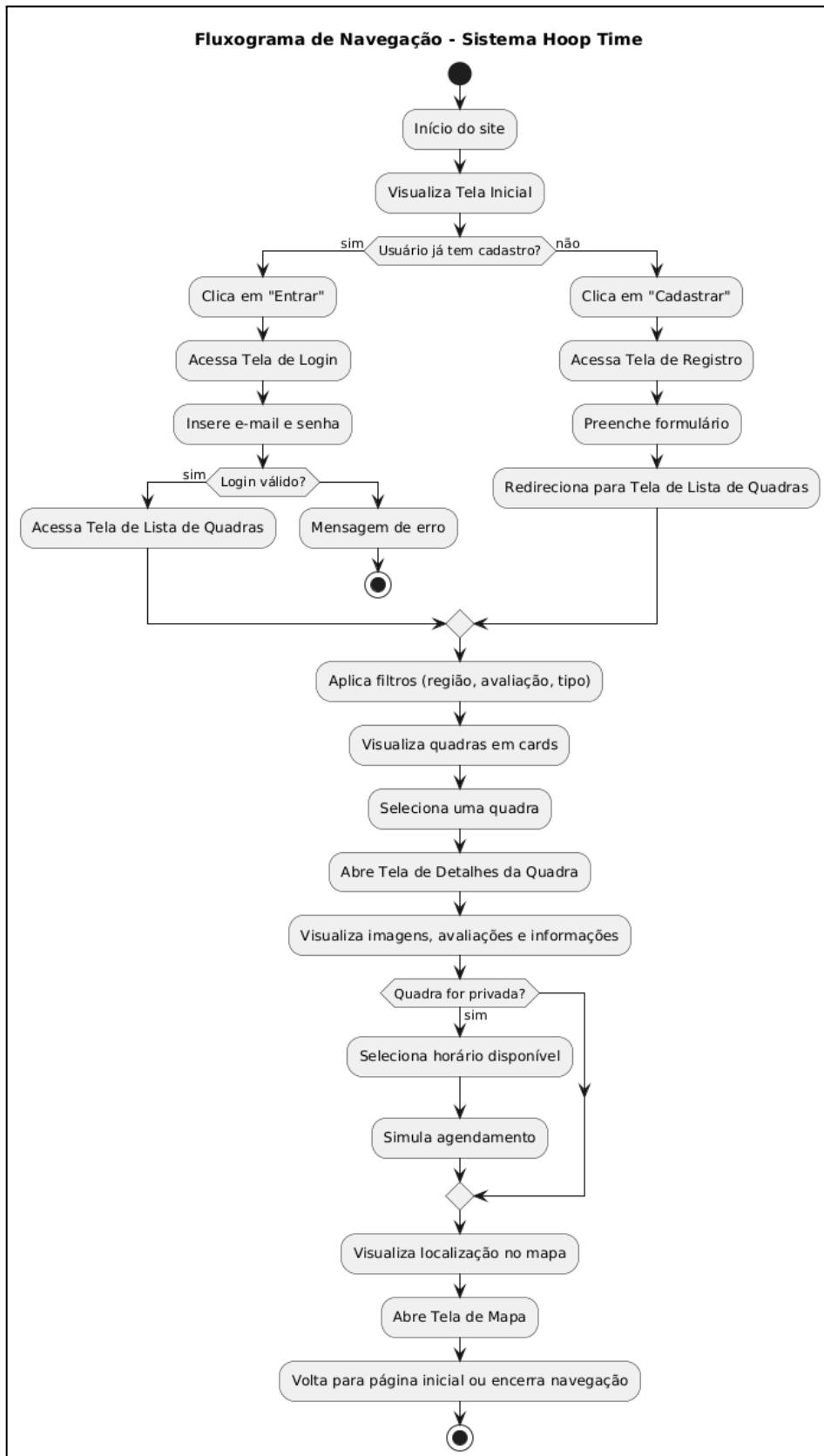
Figura 9 – Diagrama de caso de uso



Fonte: Elaborado pelo autor (2025)

Com o intuito de representar o fluxo de navegação do usuário na plataforma desenvolvida, foi elaborado um diagrama de atividade, representado logo abaixo pela figura 10. Esse diagrama expõe de maneira sequencial as principais ações realizadas pelos usuários, permitindo uma visualização clara do percurso dentro do sistema desde o primeiro acesso até funcionalidades específicas, como o agendamento de horários em quadras privadas.

Figura 10 – Diagrama de Atividade



Fonte: Elaborado pelo autor (2025)

O fluxo inicia-se com o carregamento da página inicial, onde são disponibilizadas opções como *login*, cadastro e acesso direto à visualização das quadras. A partir dessas decisões, o usuário pode seguir diferentes caminhos: caso opte pelo *login* ou cadastro, será direcionado a uma página de autenticação. Após a autenticação bem-sucedida, o sistema libera o acesso às funcionalidades completas.

O usuário pode então explorar a listagem de quadras, utilizar filtros para refinar os resultados e consultar os detalhes de cada espaço esportivo. Também é possível acessar a visualização por mapa, que oferece uma experiência geográfica das quadras cadastradas. No caso de quadras privadas, é oferecida a simulação de agendamento com horários disponíveis.

## 5.2 Requisitos

"Os requisitos são itens indispensáveis para que uma solução consiga se tornar operacional, de forma segura e eficiente para resolver um problema proposto inicialmente" (Casa do Desenvolvedor, 2023).

Os requisitos funcionais e não funcionais foram extraídos a partir da análise do problema proposto, das funcionalidades observadas em soluções similares (como Playfinder e o aplicativo 156+POA) e da definição do escopo do projeto. Como este trabalho teve caráter individual, a validação dos requisitos foi feita com base em revisões constantes durante a prototipagem da interface, considerando a usabilidade, acessibilidade e limitações técnicas de um sistema *front-end* sem integração com *back-end*.

A priorização dos requisitos seguiu uma lógica incremental, sendo priorizadas as funcionalidades essenciais para garantir a navegação, visualização e interação com as quadras cadastradas. Itens que agregam valor, como simulação de agendamento e sistema de avaliação, foram considerados secundários, mas implementados dentro da proposta de prototipagem. Os requisitos não funcionais foram definidos com foco na qualidade da experiência do usuário, responsividade e boas práticas de desenvolvimento.

### 5.2.1 Requisitos funcionais

"Um requisito funcional define uma função de um sistema de software ou seu componente. O requisito funcional representa o que o software faz, em termos de tarefas e serviços. Uma função é descrita como um conjunto de entradas, seu comportamento e as saídas" (Wikipédia, 2023).

A seguir o quadro 1 apresenta os requisitos funcionais pensados e levantados para a aplicação.

Quadro 1 – Requisitos funcionais

Código	Descrição do Requisito
RF01	Exibir lista de quadras disponíveis
RF02	Exibir detalhes da quadra (nome, endereço, tipo, imagem, avaliação)
RF03	Permitir simulação de agendamento de quadras privadas
RF04	Exibir quadras em destaque na página inicial
RF05	Filtrar quadras por tipo, localização, avaliação e nome
RF06	Exibir mapa interativo com localização das quadras
RF07	Permitir registro de usuário com formulário
RF08	Permitir autenticação por e-mail e senha
RF09	Simular <i>login</i> com conta Google
RF10	Exibir mensagens de feedback durante o <i>login</i> e registro
RF11	Redirecionar usuário para página de erro ao acessar rota inexistente

Fonte: Elaborado pelo autor (2025)

### 5.2.2 Requisitos não funcionais

“Requisitos não funcionais são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Estes requisitos dizem respeito a como as funcionalidades serão entregues ao usuário do software.” (Wikipédia, 2019).

No quadro 2, apresentado a seguir, os requisitos não funcionais levantados.

Quadro 2 – Requisitos não funcionais

Código	Descrição do Requisito
RNF01	O sistema deve ser desenvolvido em front-end utilizando React e TypeScript
RNF02	A interface deve ser responsiva e adaptável a diferentes dispositivos
RNF03	O sistema deve utilizar Styled Components para organização e reutilização de estilos
RNF04	O sistema deve apresentar boa usabilidade, com navegação clara e intuitiva
RNF05	O roteamento deve ser realizado utilizando React Router v6
RNF06	O sistema deve ser acessível, com atenção ao contraste e navegação por teclado
RNF07	A aplicação deve simular os dados por meio de listas em JSON
RNF08	O código deve seguir boas práticas de componentização e organização
RNF09	O tempo de carregamento das páginas deve ser rápido, oferecendo boa performance
RNF10	O projeto deve ser versionado com Git em repositório público no GitHub

Fonte: Elaborado pelo autor (2025)

### 5.3 Arquitetura e estrutura do projeto

O Sistema foi desenvolvido seguindo princípios modernos de desenvolvimento front-end, com foco em componentização, reutilização de código e manutenção. A arquitetura do projeto foi estruturada em camadas bem definidas, como apresentado na figura 11 logo a seguir.

Figura 11 – Diretórios e Arquivos Principais

📁 hoop-time	
📁 .git	# Repositório Git para controle de versão
📁 node_modules	# Dependências instaladas pelo npm/yarn
📁 public	# Arquivos públicos acessíveis diretamente pelo navegador
📁 src	# Código fonte da aplicação
📁 api	# Serviços e funções para comunicação com APIs
📁 assets	# Arquivos estáticos (imagens, fontes, etc.)
📁 components	# Componentes React reutilizáveis
📁 pages	# Componentes de página/tela completa
📁 styles	# Estilos globais e configurações de tema
📁 types	# Definições de tipos TypeScript
📄 App.tsx	# Componente principal que define rotas
📄 index.tsx	# Ponto de entrada da aplicação React
📄 styled.d.ts	# Definições de tipos para styled-components
📄 .gitignore	# Arquivos ignorados pelo Git
📄 package-lock.json	# Versões exatas das dependências (npm)
📄 package.json	# Configuração do projeto e dependências
📄 README.md	# Documentação do projeto
📄 tsconfig.json	# Configuração do TypeScript

Fonte: Elaborado pelo autor (2025)

#### 5.3.1 Estrutura do código da página de lista das quadras

O trecho de código abaixo apresenta o componente principal QuadrasList, que gerencia a busca e filtragem das quadras de basquete, utilizando hooks para controlar os estados dos filtros e do carregamento.

Quadro 3 – Trecho de código da página de lista de quadras

```
const QuadrasList: React.FC = () => {
  const [quadras, setQuadras] = useState<Quadra[]>([]);
  const [bairro, setBairro] = useState('');
  const [tipo, setTipo] = useState<string>('');
  const [avaliacao, setAvaliacao] = useState('');
  const [ordenacao, setOrdenacao] = useState<'relevancia' | 'avaliacao' |
  'nome-asc' | 'nome-desc'>('relevancia');
  const [isLoading, setIsLoading] = useState(false);
  const [isBuscaRealizada, setIsBuscaRealizada] = useState(false);
  const fetchQuadras = async () => {
    setIsLoading(true);
    setIsBuscaRealizada(true);
    try {
      const params: BuscaQuadrasParams = {
        bairro,
        tipo,
        avaliacaoMinima: avaliacao ? parseFloat(avaliacao) : undefined,
        ordenacao
      };
      const result = await buscarQuadras(params);
      setQuadras(result);
    } catch (error) {
      console.error('Erro ao buscar quadras:', error);
      setQuadras([]);
    } finally {
      setIsLoading(false);
    }
  };
  const handleBuscar = () => {
    fetchQuadras();
  };
};
```

Fonte: Elaborado pelo autor (2025)

O componente utiliza vários estados do React para controlar:

1. quadras: Armazena a lista de quadras retornadas pela API;
2. bairro, tipo, avaliação e ordenação: Controlam os filtros selecionados pelo usuário;
3. isLoading: Indica quando uma busca está em andamento;
4. isBuscaRealizada: Controla se já foi realizada pelo menos uma busca.

A função fetchQuadras é responsável por:

- Atualizar o estado de carregamento;
- Montar os parâmetros da busca com base nos filtros selecionados;
- Chamar a API buscarQuadras;
- Atualizar o estado com o resultado ou tratar erros;
- Finalizar o estado de carregamento.

O método handleBuscar é provavelmente chamado quando o usuário clica no botão de busca, disparando a função fetchQuadras.

Este código segue boas práticas de React, como:

- Uso de estados para controlar dados dinâmicos;

- Tratamento assíncrono com *try/catch*;
- Gerenciamento de estado de carregamento;
- Tipagem adequada com TypeScript para maior segurança.

Esta parte do sistema permite aos usuários filtrar quadras por bairro, tipo (pública/privada), avaliação mínima e ordenar os resultados conforme suas preferências, formando uma parte essencial da experiência de busca da aplicação.

### **5.3.2 Estrutura do código da página de detalhes da quadra**

O código a seguir implementa duas funções principais do componente QuadraDetail: um useEffect que busca os dados da quadra ao montar o componente ou quando o id muda, e uma função renderStars que converte a avaliação numérica em estrelas.

Quadro 4 – Trecho de código da página de detalhes da quadra

```
useEffect(() => {
  const fetchQuadra = async () => {
    if (!id) return;

    setIsLoading(true);
    try {
      const quadraId = parseInt(id, 10);
      const result = await buscarQuadraPorId(quadraId);
      if (result) {
        setQuadra(result);
        setError(null);
      } else {
        setError('Quadra não encontrada');
      }
    } catch (err) {
      console.error('Erro ao buscar detalhes da quadra:', err);
      setError('Ocorreu um erro ao carregar os detalhes da quadra');
    } finally {
      setIsLoading(false);
    }
  };

  fetchQuadra();
}, [id]);
const renderStars = (rating: number) => {
  const fullStars = Math.floor(rating);
  const hasHalfStar = rating % 1 >= 0.5;
  const stars = [];
  for (let i = 0; i < fullStars; i++) {
    stars.push('★');
  }
}
```

Fonte: Elaborado pelo autor (2025)

Ele mostra duas funções importantes:

1. Um useEffect que busca os dados da quadra quando o componente é montado ou quando o ID da quadra muda:

- O hook recebe o parâmetro id da URL (provavelmente via useParams do React Router);
  - Implementa uma função assíncrona fetchQuadra que:
    - Verifica se há um ID válido;
    - Ativa o estado de carregamento;
    - Converte o ID (que vem como string) para número inteiro;
    - Chama a API buscarQuadraPorId para obter os detalhes da quadra;
    - Atualiza o estado com os dados obtidos ou configura uma mensagem de erro;
    - Desativa o estado de carregamento após a operação.
2. Uma função renderStars que converte a avaliação numérica em uma representação visual de estrelas:
- Calcula o número de estrelas cheias usando Math.floor;
  - Verifica se deve adicionar meia estrela (quando o decimal é maior ou igual a 0,5);
  - Prepara um array para montar a representação visual.

Este código demonstra boas práticas como:

- Uso do hook useEffect para buscar dados quando o componente é montado;
- Tratamento adequado de erros com *try/catch*;
- Gerenciamento de estado de carregamento;
- Controle de efeitos colaterais com *array* de dependências;
- Transformação de dados (converter avaliação numérica em representação visual).

A implementação segue o padrão React de componentes funcionais com *hooks*, garantindo que os detalhes da quadra sejam carregados corretamente e exibidos com uma representação visual de avaliação por estrelas.

### **5.3.3 Estrutura do código da página de login**

O trecho de código abaixo apresenta o componente Login, responsável pela autenticação dos usuários, utilizando hooks para gerenciar estados do formulário e controle de carregamento.

Quadro 5 – Trecho de código da página de *login*

```
const Login: React.FC = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');
  const [isLoading, setIsLoading] = useState(false);
  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    if (!email || !password) {
      setError('Por favor, preencha todos os campos');
      return;
    }
    try {
      setIsLoading(true);
      await new Promise(resolve => setTimeout(resolve, 1000));
      console.log('Login com:', { email, password });
    } catch (err) {
      setError('Ocorreu um erro durante o login. Tente novamente.');
    } finally {
      setIsLoading(false);
    }
  };
};
```

Fonte: Elaborado pelo autor (2025)

O componente utiliza *hooks* do React para gerenciar o estado:

- email e password: Armazenam os dados inseridos pelo usuário no formulário;
- error: Guarda mensagens de erro para exibição ao usuário;
- isLoading: Controla o estado de carregamento durante o processo de autenticação.

A função handleSubmit é executada quando o formulário é enviado e realiza:

1. Prevenção do comportamento padrão do formulário com e.preventDefault();
2. Validação básica dos campos, verificando se email e senha foram preenchidos;
3. Gerenciamento do estado de carregamento (ativando antes da requisição e desativando ao final);
4. Uma simulação de chamada à API com setTimeout de 1 segundo (onde seria implementada a autenticação real);
5. Controle de erros com try/catch;

Características importantes:

- O código está em uma fase de desenvolvimento/prototipagem, pois não implementa realmente a autenticação (apenas simula);
- Existe comentário indicando onde seria implementada a lógica real de autenticação e redirecionamento após *login*;
- Há tratamento adequado para mostrar *feedback* ao usuário (mensagens de erro e estado de carregamento);
- A implementação segue a abordagem de componentes funcionais com hooks do React;

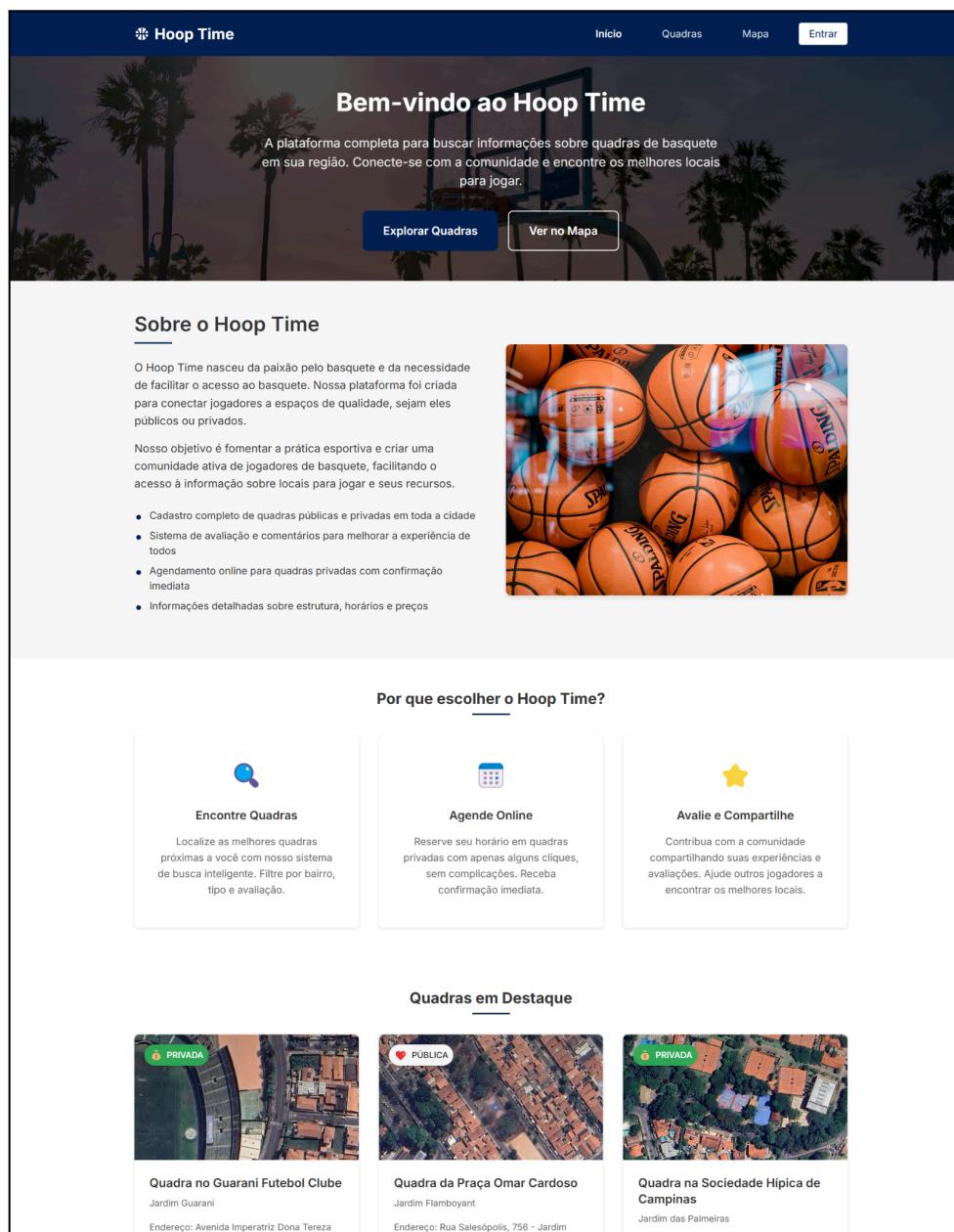
Este é um padrão comum para implementação de formulários de *login* em aplicações React, com estados para controlar os campos do formulário, validação e *feedback* ao usuário sobre o processo de autenticação.

## 6 RESULTADOS

### 6.1 Página Inicial

Ao entrar no site, o usuário é recebido pela página inicial que apresenta uma barra de navegação contendo as opções principais: Home, Quadras, Mapa e Entrar. A página inicial exibe uma imagem em destaque, acompanhada de um texto introdutório. Logo abaixo, são apresentadas as quadras em destaque através de cards. Veja a seguir a figura 12 representando a página inicial.

Figura 12 – Página inicial



Fonte: Elaborado pelo autor (2025)

## 6.2 Página de login

Na página de *login*, representada pela figura 13 abaixo, o usuário encontra uma interface centralizada com campos para inserção de e-mail e senha. A página oferece a possibilidade de integração do *back-end* com recurso de autenticação com conta Google para *login* simplificado, além de *links* para recuperação de senha e cadastro de novos usuários. Mensagens de feedback são exibidas para orientar o usuário durante o processo de autenticação.

Figura 13 – Página de *login*

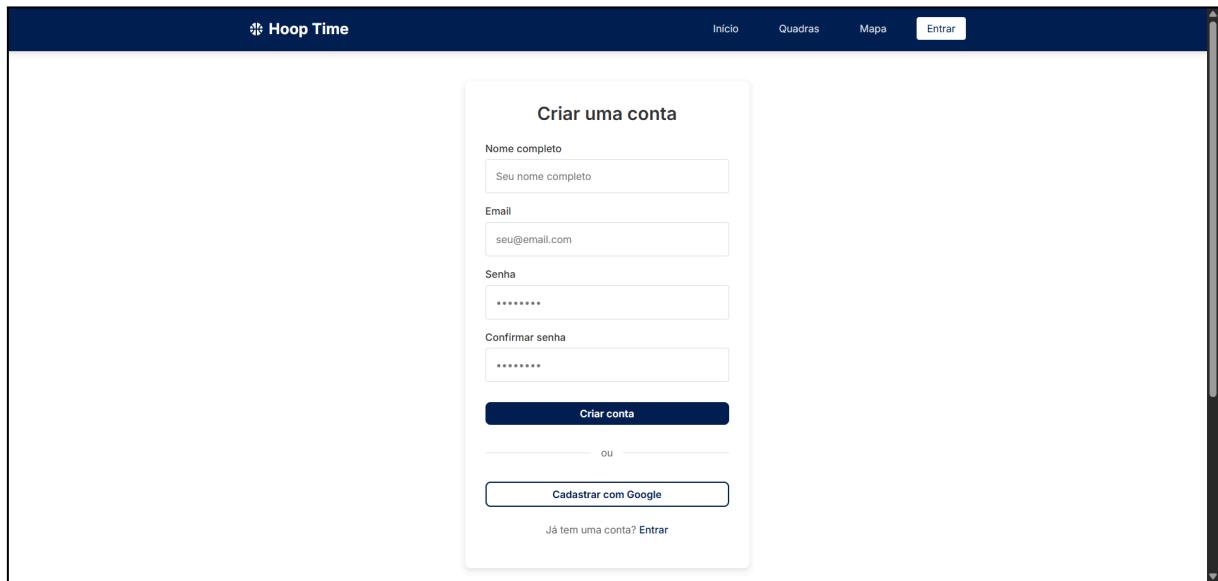
A imagem é uma captura de tela de uma página web. No topo, há uma barra azul com o logo "Hoop Time" e menu com opções "Início", "Quadras", "Mapa" e "Entrar". A seção principal, intitulada "Entre na sua conta", contém campos para "Email" (com o placeholder "seu@email.com") e "Senha" (com placeholder de asteriscos). Há um link "Esqueci minha senha" e um botão "Entrar". Abaixo, uma barra horizontal com o texto "ou" separa o formulário do link "Entrar com Google". Abaixo da barra, há uma mensagem "Ainda não tem uma conta? Cadastre-se". No rodapé, há uma barra cinza com links para "Hoop Time", "Links Rápidos" (submenu com "Início" e "Sobre Nós"), "Recursos" (submenu com "Sobre Nós") e "Contato" (link com "contato@hooptime.com.br").

Fonte: Elaborado pelo autor (2025)

## 6.3 Página de Registro

O processo de registro apresenta um formulário estruturado com campos para informações do usuário. O sistema fornece indicadores de preenchimento obrigatório. Abaixo é apresentada a página de registro pela figura 14.

Figura 14 – Página de registro

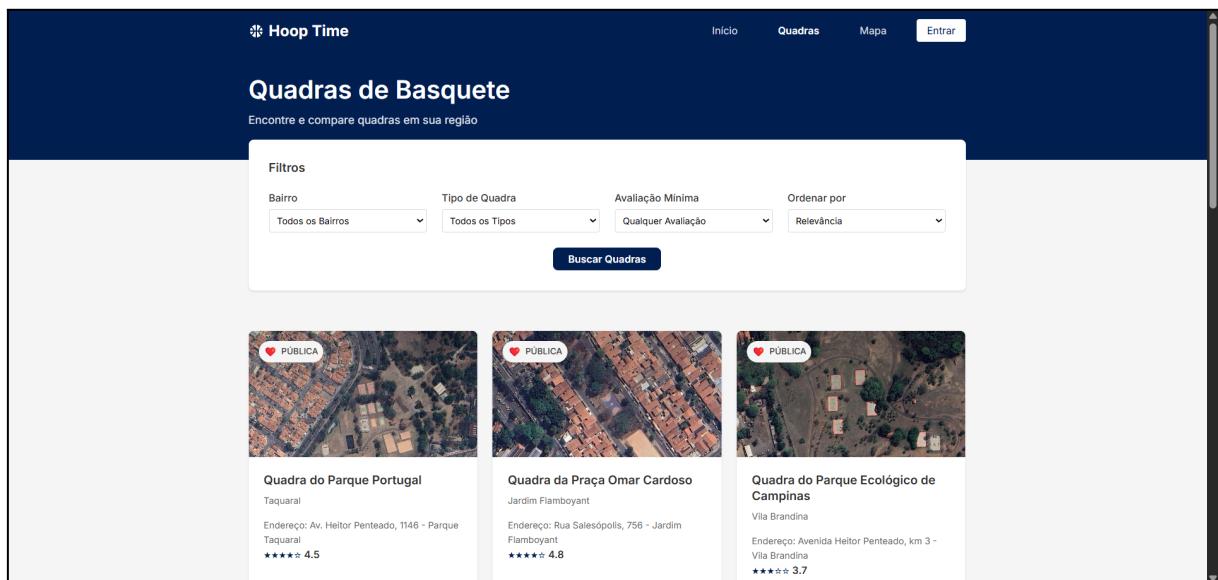


Fonte: Elaborado pelo autor (2025)

#### 6.4 Página de Lista de Quadras

A página de listagem de quadras, exibida pela figura 15 logo abaixo, apresenta um conjunto de cards organizados em *grid*, onde cada *card* exibe uma imagem ampla da região onde se encontra a quadra, nome, localização e avaliação média dos usuários. A interface inclui uma barra de filtros que permite refinar a busca por região, tipo de quadra (pública ou privada), avaliação e disponibilidade.

Figura 15 – Página de lista de quadras



Fonte: Elaborado pelo autor (2025)

## 6.5 Página de Detalhes da Quadra

Na página de detalhes da quadra, mostrada na figura 16, o usuário encontra a possibilidade de visualizar uma imagem mais ampla da área da quadra selecionada. As informações são organizadas em seções que incluem informações acerca da quadra e avaliações de outros usuários. A página também apresenta um botão para acesso ao mapa com a localização exata da quadra e um sistema de reservas integrado para agendamento de horários de quadras privadas.

Figura 16 – Página de detalhes da quadra

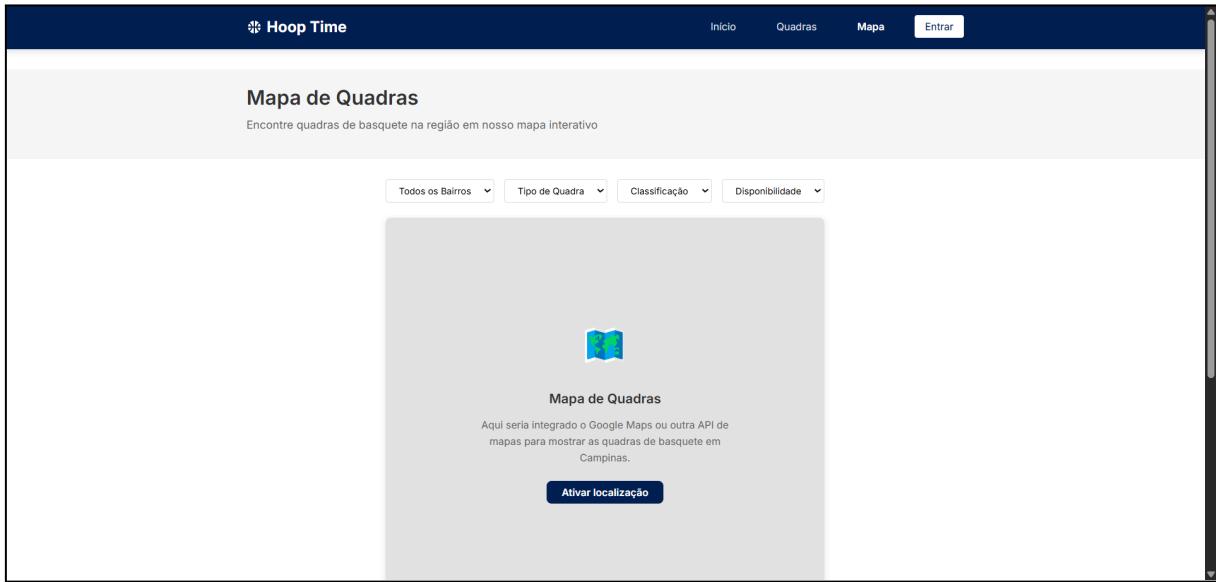


Fonte: Elaborado pelo autor (2025)

## 6.6 Página de Mapa

A visualização em mapa oferece uma perspectiva geográfica das quadras disponíveis. A estrutura da página foi implementada de forma a estar pronta para integrar uma API que forneça dinamicamente os dados das quadras, permitindo assim a exibição automática das informações no mapa conforme forem disponibilizadas no *back-end*. A seguir a página de mapa representada na figura 17.

Figura 17 – Página de mapa

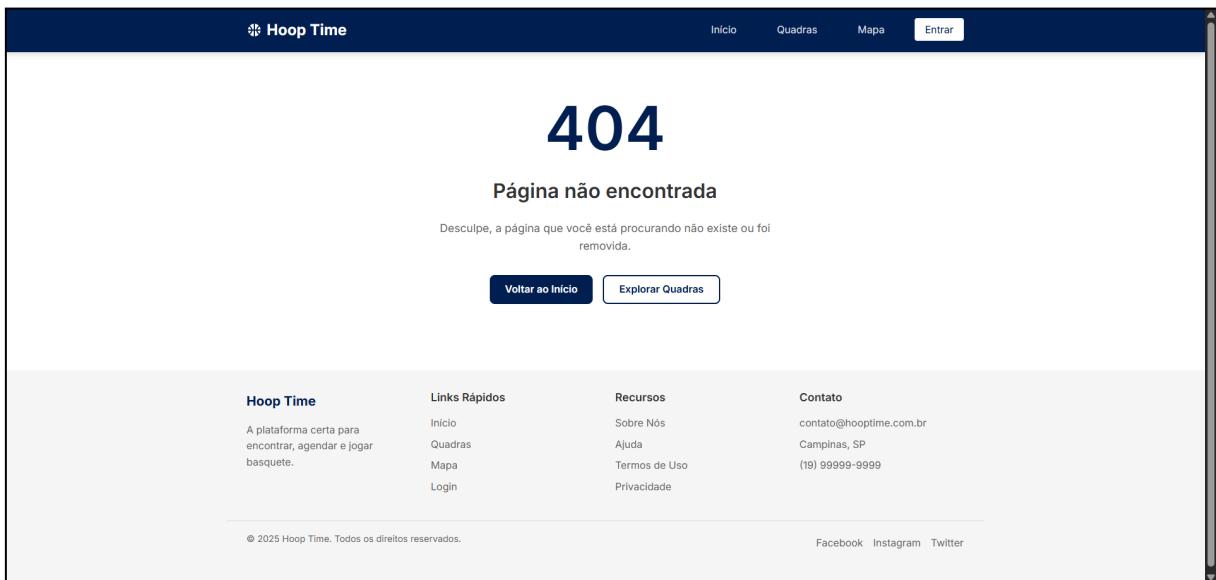


Fonte: Elaborado pelo autor (2025)

## 6.7 Página de Tratamento de Erro

A página de erro, mostrada na figura 18 a seguir, é exibida quando uma página não é encontrada, oferecendo uma mensagem informativa e botões de navegação para retornar às principais áreas do site. A página mantém a estrutura de navegação padrão, permitindo que o usuário continue explorando o sistema facilmente.

Figura 18 – Página de erro



Fonte: Elaborado pelo autor (2025)

O desenvolvimento das páginas apresentadas demonstra a implementação prática dos requisitos funcionais estabelecidos na metodologia, resultando em uma interface web

completa e funcional para o sistema de localização de quadras esportivas. A arquitetura das páginas foi estruturada de forma modular e responsiva, garantindo uma experiência de usuário consistente em diferentes dispositivos. As funcionalidades implementadas atendem às necessidades identificadas durante a pesquisa, proporcionando aos usuários uma ferramenta eficiente para descobrir, avaliar e visualizar quadras esportivas em sua região. A interface desenvolvida consolida os conceitos de usabilidade e design responsivo aplicados ao contexto esportivo, demonstrando como tecnologias modernas de desenvolvimento *front-end* podem ser utilizadas para criar soluções práticas e acessíveis no segmento esportivo.

## 7 CONSIDERAÇÕES FINAIS

Atualmente, a prática esportiva, especialmente o basquete, tem ganhado cada vez mais adeptos no Brasil, impulsionada tanto pela popularização do esporte quanto pela crescente busca por atividades físicas e lazer. Esse interesse crescente cria uma demanda por soluções que facilitem o acesso às estruturas esportivas e promovam a integração entre praticantes. No entanto, ainda existe uma significativa dificuldade em localizar e reservar quadras de basquete de forma prática e eficiente.

O Hoop Time foi desenvolvido com o objetivo de ser uma plataforma que conecta jogadores de basquete às quadras disponíveis em sua região. Por meio dessa aplicação, o usuário pode encontrar quadras próximas, verificar sua disponibilidade e visualizar informações detalhadas sobre cada espaço. A plataforma foi projetada priorizando a usabilidade, permitindo que qualquer pessoa, desde iniciantes até jogadores experientes, possa navegar e utilizar todas as funcionalidades de forma intuitiva.

A interface do sistema oferece recursos que enriquecem a experiência do usuário, como a visualização detalhada das estruturas disponíveis através de *cards* informativos e um layout responsivo que se adapta a diferentes dispositivos. O desenvolvimento focou em criar uma experiência visual agradável e moderna, utilizando React como *framework* principal e implementando conceitos avançados de componentização.

Como trabalhos futuros, destaca-se a necessidade de implementação de um *back-end* robusto com persistência de dados em banco de dados, permitindo o armazenamento de informações de usuários, quadras e reservas. A integração com uma API de mapas também seria fundamental para fornecer funcionalidades de geolocalização e busca por proximidade.

O Hoop Time representa um primeiro passo importante na modernização do acesso às estruturas esportivas, demonstrando o potencial da tecnologia para facilitar a prática esportiva. O projeto estabelece uma base sólida de interface do usuário, sobre a qual podem ser construídas diversas funcionalidades adicionais que enriquecerão ainda mais a experiência dos usuários.

## REFERÊNCIAS

AMAZON WEB SERVICES. Qual é a diferença entre front-end e back-end no desenvolvimento de aplicações? [S.l.]: Amazon Web Services, 2024. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-frontend-and-backend>. Acesso em: 12 out. 2024.

BOOKTEQ. Playfinder. [S.l.]: Bookteq, 2025. Disponível em: <https://www.playfinder.com>. Acesso em: 10 fev. 2025.

CASA DO DESENVOLVEDOR. Requisitos funcionais e não funcionais: o que são e como identificar?. [S.l.]: Casa do Desenvolvedor, 24 fev. 2023. Disponível em: <https://blog.casadodesenvolvedor.com.br/requisitos-funcionais-e-nao-funcionais>. Acesso em: 13 out. 2024.

EXCALIDRAW. Excalidraw developer docs. [S.l.]: Excalidraw, 2023. Disponível em: <https://docs.excalidraw.com>. Acesso em: 30 mar. 2025.

FREVEL, N.; BEIDERBECK, D.; SCHMIDT, S. L. The impact of technology on sports – A prospective study. **Technological Forecasting and Social Change**, [S.l.], v. 182, p. 121838, set. 2022. DOI: <https://doi.org/10.1016/j.techfore.2022.121838>. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0040162522003626?via%3Dihub>. Acesso em: 19 out. 2024.

GAUDIN, B. C. P. O basquete no país do futebol. **Revista de Ciências Sociais**, Fortaleza, v. 38, n. 1, p. 53-58, 2007. Disponível em: <http://www.repositorio.ufc.br/handle/riufc/10181>. Acesso em: 21 abr. 2025.

GIT. Git Documentation. [S.l.]: Git, [s.d.]. Disponível em: <https://git-scm.com/doc>. Acesso em: 24 fev. 2025.

GITHUB. GitHub Docs. [S.l.]: Github, 2025. Disponível em: <https://docs.github.com/pt>. Acesso em: 24 fev. 2025.

MARTINS, D. G.; DOTTO, A. D.; MYSKIW, M. A apropriação de quadras esportivas e o uso de aplicativo para agendamento na nova orla do Guaíba. In: XXIII Congresso Brasileiro de Ciências do Esporte; IX CONICE - CONBRACE, 2023, Fortaleza, CE, Brasil. **Anais** [...]. CONBRACE, 2023. Disponível em: <https://cev.org.br/biblioteca/a-apropriacao-de-quadras-esportivas-e-o-uso-de-aplicativo-para-agendamento-na-nova-orla-do-guaiba>. Acesso em: 24 nov. 2024.

META OPEN SOURCE. Quick Start – React. [S.l.]: Meta Open Source, [s.d.]. Disponível em: <https://react.dev/learn>. Acesso em: 10 nov. 2024.

MICROSOFT. TypeScript Documentation . [S.l.]: Microsoft, 2025. Disponível em: <https://www.typescriptlang.org/docs>. Acesso em: 10 nov. 2024.

MOZILLA FOUNDATION. HTML: Linguagem de Marcação de Hipertexto. [S.l.]: Mozilla Foundation, 2025. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 24 fev. 2025.

NPM. Npm Docs. [S.l.]: Npm, [s.d.]. Disponível em: <https://docs.npmjs.com>. Acesso em: 24 fev. 2025.

OPENJS FOUNDATION. Node.js v24.1.0 documentation. [S.l.]: OpenJS Foudation, [s.d.]. Disponível em: <https://nodejs.org/docs/latest/api>. Acesso em: 24 fev. 2025.

PELLEGRINOTTI, I. L. Atividade física e esporte: a importância no contexto saúde do ser humano. **Revista Brasileira de Atividade Física & Saúde**, Piracicaba, v. 3, n. 1, 1998. Publicado em 15 out. 2012. Disponível em: <https://periodicos.ufpel.edu.br/index.php/RBAFS/article/view/1067>. Acesso em: 19 out. 2024.

PREFEITURA DE PORTO ALEGRE. Quadras do trecho 3 da Orla podem ser agendadas pelo 156+POA. Prefeitura de Porto Alegre, 27 out. 2021. Disponível em: <https://prefeitura.poa.br/smtc/noticias/quadradas-do-trecho-3-da-orla-podem-ser-agendadas-pelo-156poa>. Acesso em: 24 nov. 2024.

REMIX SOFTWARE. React Router. [S.l.]: Remix Software, [s.d.]. Disponível em: <https://reactrouter.com/home>. Acesso em: 24 fev. 2025.

STYLED COMPONENTS. styled-components: Documentation. [S.l.]: Styled Components, [s.d.]. Disponível em: <https://styled-components.com/docs>. Acesso em: 17 fev. 2025.

TAHEIJI, A.; NODA, F.; SALES, F. **Sobe a “laranja”: o crescimento do basquete e sua cultura no Brasil**. [S.l.]: Medium, 28 jun. 2022. Disponível em: <https://medium.com/@laboratoriodejornalismo2022/sobe-a-laranja-o-crescimento-do-basquete-e-sua-cultura-no-brasil-cd2767c83617>. Acesso em: 5 out. 2024.

TECNOFIT. Gerenciamento de quadras esportivas: qual a importância e como fazer?. [S.l.]: Tecnofit, 16 dez. 2024. Disponível em: <https://www.tecnofit.com.br/blog/gerenciamento-de-quadradas-esportivas>. Acesso em: 20 dez. 2024.

WIKIPÉDIA. Requisito funcional. [S.l.]: Wikimedia Foundation, 2023. Disponível em: [https://pt.wikipedia.org/wiki/Requisito\\_funcional](https://pt.wikipedia.org/wiki/Requisito_funcional). Acesso em: 4 maio 2025.

WIKIPÉDIA. Requisito não funcional. [S.l.]: Wikimedia Foundation, 2019. Disponível em: [https://pt.wikipedia.org/wiki/Requisito\\_não\\_funcional](https://pt.wikipedia.org/wiki/Requisito_n%C3%A3o_funcional). Acesso em: 4 maio 2025.