

Sistema de Agendamento de Horários para Padel: Uma Solução Digital para Otimizar Reservas

Anderson Antonio Cagnini - 101508; Ângelo Giroletto - 109699; Eduardo Dutra Carminatti - 104105; Felipe Meneguzzi - 103418; Levi De Sousa Gronenschild - 108540; Rafael de Oliveira - 108408; Renan Poersch - 033056.

Ciência da Computação

Universidade Regional Integrada do Alto Uruguai e das Missões (URI) Erechim – RS –
Brasil

1. Introdução

O padel é um esporte em constante crescimento em diversas cidades, atraindo cada vez mais praticantes e despertando o interesse de clubes e academias em oferecer quadras para aluguel. No entanto, em muitas localidades, a forma de agendar horários para a prática ainda é realizada de maneira pouco organizada, geralmente por meio de mensagens em aplicativos como o WhatsApp. Esse método, apesar de prático em alguns casos, pode gerar falhas no controle de reservas, conflitos de agendamento, dificuldade na gestão de horários disponíveis e até mesmo perda de oportunidades de utilização das quadras.

Diante dessa realidade, surgiu a ideia de desenvolver um sistema digital de agendamento de horários específico para o padel, que permita aos usuários visualizar a disponibilidade em tempo real e efetuar suas reservas de forma rápida, prática e segura. Além de otimizar a organização, a proposta visa reduzir erros, facilitar a comunicação entre administradores e jogadores, e contribuir para a modernização da gestão esportiva na cidade.

O sistema tem como principal objetivo oferecer uma plataforma acessível, intuitiva e eficiente, que substitua o modelo tradicional de agendamento manual, trazendo maior comodidade para os usuários e agilidade para os administradores das quadras. Ao adotar essa solução, espera-se também incentivar ainda mais a prática do padel, tornando o acesso ao esporte mais simples e estruturado.

2. Metodologia

O desenvolvimento do sistema de agendamento de horários para o padel foi conduzido a partir de uma abordagem prática e iterativa, sem a adoção formal de uma metodologia rígida, mas inspirada em princípios ágeis de construção incremental. O grupo optou por trabalhar de maneira colaborativa e flexível, organizando as etapas de acordo com a evolução das necessidades identificadas ao longo do projeto. Esse modelo possibilitou ajustes contínuos, favorecendo tanto a implementação de funcionalidades prioritárias quanto a realização de testes e melhorias em tempo real.

Inicialmente, realizou-se o levantamento de requisitos a partir da observação das dificuldades enfrentadas pela comunidade local no processo de agendamento de quadras, que ocorre de forma manual, por meio de mensagens em aplicativos de comunicação. Essa análise serviu de base para a definição das funcionalidades essenciais do sistema, como a visualização da disponibilidade em tempo real, a reserva de horários e o gerenciamento das quadras. Em seguida, foram elaborados protótipos de baixa fidelidade que auxiliaram na definição da organização das telas e dos fluxos de navegação. A atenção esteve voltada para a simplicidade e a usabilidade, de modo a garantir que qualquer usuário conseguisse realizar o agendamento de maneira intuitiva.

A fase de implementação foi realizada utilizando as tecnologias React.js e JavaScript, que possibilitaram o desenvolvimento de uma aplicação web responsiva, modular e eficiente. O React, por meio de sua arquitetura baseada em componentes e no Virtual DOM, contribuiu para a criação de interfaces dinâmicas e de fácil manutenção, enquanto o JavaScript, com seus recursos de programação assíncrona e manipulação de eventos, viabilizou a integração entre as funcionalidades e a fluidez da aplicação.

Por fim, foram realizados testes práticos com o objetivo de verificar a usabilidade, a responsividade e a confiabilidade do sistema. Essa etapa permitiu corrigir inconsistências, ajustar detalhes de navegação e validar a aplicação em diferentes dispositivos, garantindo maior qualidade no produto final. Dessa forma, o projeto alcançou seu objetivo central de oferecer uma solução digital capaz de modernizar e otimizar o processo de agendamento de quadras de padel na cidade, substituindo o modelo manual por uma plataforma acessível e eficiente.

3. Tecnologias Utilizadas

3.1 React.js

O React.js constitui uma biblioteca JavaScript criada pelo Facebook (Meta) destinada ao desenvolvimento de interfaces de usuário dinâmicas e responsivas. Fundamentado na arquitetura de componentes reutilizáveis e no paradigma do Virtual DOM, o React possibilita a criação de aplicações web interativas com alta performance, através da otimização do processo de atualização da interface e da reconciliação eficiente entre o DOM virtual e o DOM real (React, 2025a; React, 2025b).

A estrutura baseada em estado (state) e propriedades (props) proporciona um gerenciamento de dados simplificado nos componentes, permitindo que a interface responda de forma automática às alterações dos dados. O sistema de hooks, implementado a partir da versão 16.8 do React, capacita os componentes funcionais a utilizarem estado e efeitos colaterais dispensando o uso de classes, resultando em maior simplicidade no desenvolvimento e manutenção das aplicações (React, 2025c).

A seleção do React.js para este projeto fundamenta-se em sua extensa comunidade e ecossistema consolidado, que disponibiliza uma diversificada gama de bibliotecas e ferramentas de apoio, além da performance aprimorada oferecida pelo Virtual DOM e sua reconciliação otimizada. A compatibilidade nativa com TypeScript e a facilidade de integração com tecnologias contemporâneas estabelecem o React como uma alternativa robusta para o desenvolvimento de aplicações em larga escala (React, 2025a; React, 2025d).

3.2 Javascript

O JavaScript constitui uma linguagem de programação interpretada e de alto nível, desenvolvida originalmente pela Netscape em 1995, destinada à criação de aplicações web interativas e dinâmicas. Fundamentado na tipagem dinâmica e no paradigma multi-paradigma, o JavaScript possibilita o desenvolvimento tanto no lado cliente (front-end) quanto no servidor (back-end), através de sua flexibilidade sintática e

capacidade de manipulação do Document Object Model (DOM) em tempo real (MDN, 2025a; ECMAScript, 2025b).

Sua arquitetura baseada em eventos (event-driven) e funções de primeira classe facilita a criação de aplicações responsivas e interativas, permitindo que o código responda automaticamente às ações do usuário e mudanças no estado da aplicação. O sistema de closures e programação assíncrona, implementado através de Promises e async/await, capacita o desenvolvimento de aplicações eficientes que lidam com operações não-bloqueantes, resultando em maior fluidez na experiência do usuário (MDN, 2025c).

A seleção do JavaScript para este projeto fundamenta-se em sua universalidade e ecossistema robusto, que disponibiliza uma vasta gama de frameworks, bibliotecas e ferramentas de desenvolvimento, além da performance aprimorada oferecida pelos motores JavaScript modernos como V8 e SpiderMonkey. Sua compatibilidade nativa com tecnologias web e facilidade de integração com APIs modernas estabelecem o JavaScript como uma alternativa consolidada para o desenvolvimento de aplicações escaláveis e multiplataforma (ECMAScript, 2025a; Node.js, 2025d).

3.3 Figma

O Figma constitui uma plataforma de design colaborativo baseada na web, desenvolvida pela Figma Inc. em 2016, destinada à criação de interfaces digitais e prototipagem interativa. Fundamentado na arquitetura de design vetorial e no paradigma de colaboração em tempo real, o Figma possibilita o desenvolvimento de designs responsivos e sistemas de design escaláveis, através da otimização do fluxo de trabalho colaborativo e da sincronização automática entre membros da equipe (Figma, 2025a; Figma, 2025b).

Sua estrutura baseada em componentes (components) e variantes (variants) proporciona um gerenciamento de design sistematizado, permitindo que as interfaces mantenham consistência visual e respondam automaticamente às alterações nos elementos base. O sistema de Auto Layout e Constrains, implementado para facilitar a

criação de layouts adaptativos, capacita os designers a desenvolverem interfaces responsivas sem a necessidade de múltiplas versões, resultando em maior eficiência no processo de design e manutenção de projetos (Figma, 2025c).

A seleção do Figma para este projeto fundamenta-se em sua capacidade de colaboração simultânea e ecossistema integrado, que disponibiliza uma diversificada gama de plugins e recursos de prototipagem, além da performance aprimorada oferecida pela renderização vetorial na web e sincronização em nuvem. Sua compatibilidade nativa com ferramentas de desenvolvimento e facilidade de handoff para equipes técnicas estabelecem o Figma como uma alternativa robusta para o design de interfaces em projetos de grande escala (Figma, 2025a; Figma, 2025d).

3.4 Node.js

Node.js é um ambiente de execução JavaScript baseado no motor V8 do Chrome, que permite rodar código JavaScript fora do navegador. Ele adota um modelo de entrada e saída não bloqueante, voltado para aplicações escaláveis e de alta performance, especialmente em servidores web e APIs (OPENJS FOUNDATION, 2025).

Com suporte a inúmeros módulos nativos e integração com o gerenciador de pacotes npm, o Node.js se tornou a base para grande parte do desenvolvimento moderno. Ele é amplamente utilizado tanto em back-end quanto em ferramentas de build e automação no front-end (OPENJS FOUNDATION, 2025).

3.5 PostgreSQL

Prisma é uma ferramenta de mapeamento objeto-relacional (ORM) para Node.js e TypeScript que simplifica o acesso a bancos de dados relacionais. Ele oferece um cliente fortemente tipado e migrações automatizadas, aumentando a segurança e reduzindo erros em consultas SQL (PRISMA DATA, 2025).

Com integração nativa em aplicações Next.js, Prisma permite consultas diretas em Server Components e APIs, otimizando tanto o fluxo de dados quanto a

performance. Essa abordagem facilita o desenvolvimento full-stack e garante maior consistência entre as camadas da aplicação (PRISMA DATA, 2025).

4. Ui/Ux Design e Desenvolvimento

O design da interface do sistema de agendamento de horários para o padel foi elaborado com foco na atratividade visual e na facilidade de uso. Para isso, foi utilizada uma paleta de cores que combina verde no fundo com detalhes em azul, branco e preto, criando um ambiente visualmente chamativo e agradável, sem comprometer a clareza das informações apresentadas. Essa escolha de cores também remete à identidade esportiva do padel, transmitindo energia e dinamismo.

O processo de prototipação foi realizado utilizando ferramentas como Figma e Photoshop, permitindo a criação de modelos de interface e elementos visuais que orientaram a implementação do sistema. Esses protótipos serviram como base para o desenvolvimento das telas, garantindo consistência visual e fluidez na navegação entre as diferentes funcionalidades do sistema.

A experiência do usuário (UX) foi cuidadosamente planejada para atender tanto usuários que acessam o sistema via computador quanto aqueles que utilizam dispositivos móveis, garantindo total responsividade. O processo de agendamento foi estruturado para ser intuitivo, permitindo que os usuários realizem reservas com poucos cliques, mas mantendo funcionalidades mais detalhadas para aqueles que desejam gerenciar horários e quadras de forma completa. Além disso, foram adotados princípios de acessibilidade, como botões de tamanho adequado, letras legíveis e contraste de cores, assegurando que a plataforma seja inclusiva e de fácil uso para pessoas de diferentes idades e perfis.

A implementação das ideias de UI e UX foi realizada utilizando React.js e JavaScript, com a criação de componentes personalizados que seguiram os padrões estabelecidos nos protótipos. Cada elemento visual e funcional foi desenvolvido manualmente, garantindo flexibilidade para ajustes futuros e integração perfeita entre a estética do sistema e sua usabilidade. Dessa forma, o projeto alcançou um equilíbrio entre um design atrativo e uma experiência de usuário prática, eficiente e agradável,

consolidando uma plataforma que facilita o agendamento de quadras de padel de maneira moderna e organizada.

5. Figuras

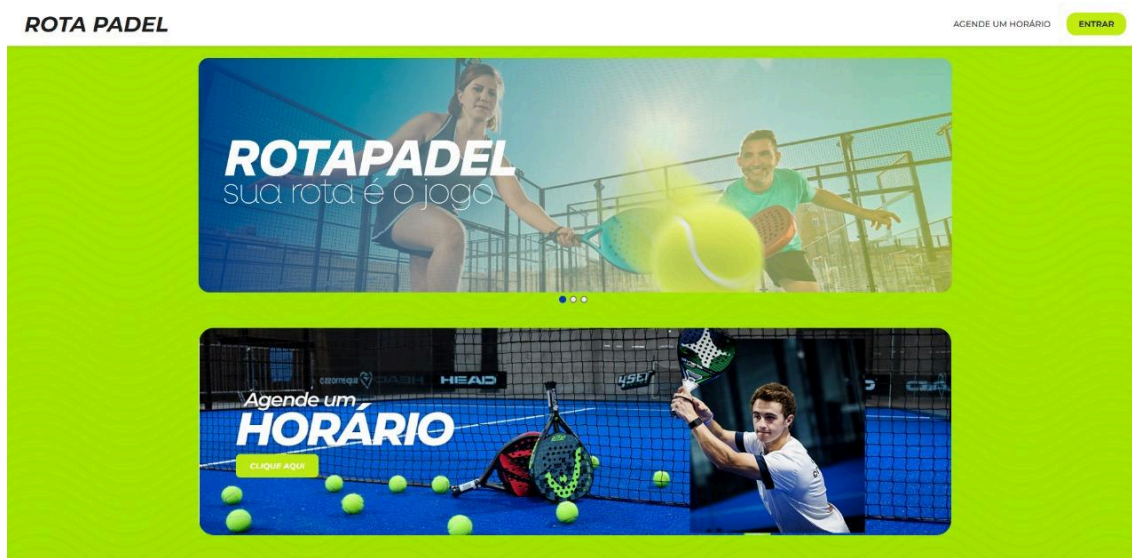


Figura 1 - Tela de início do site.

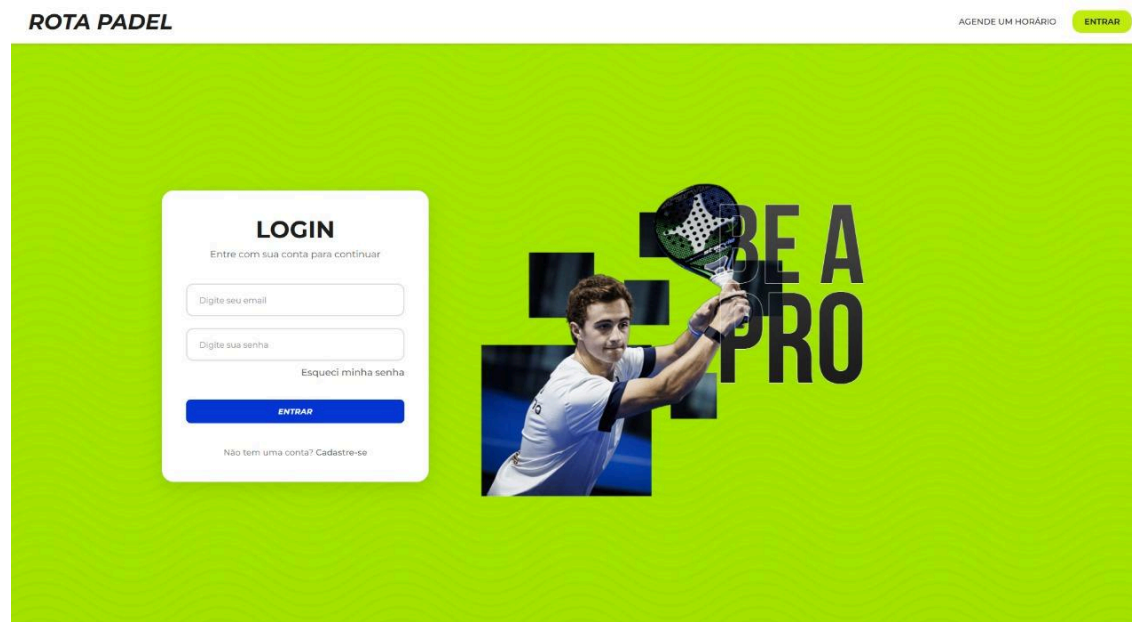


Figura 2 - Tela de login.

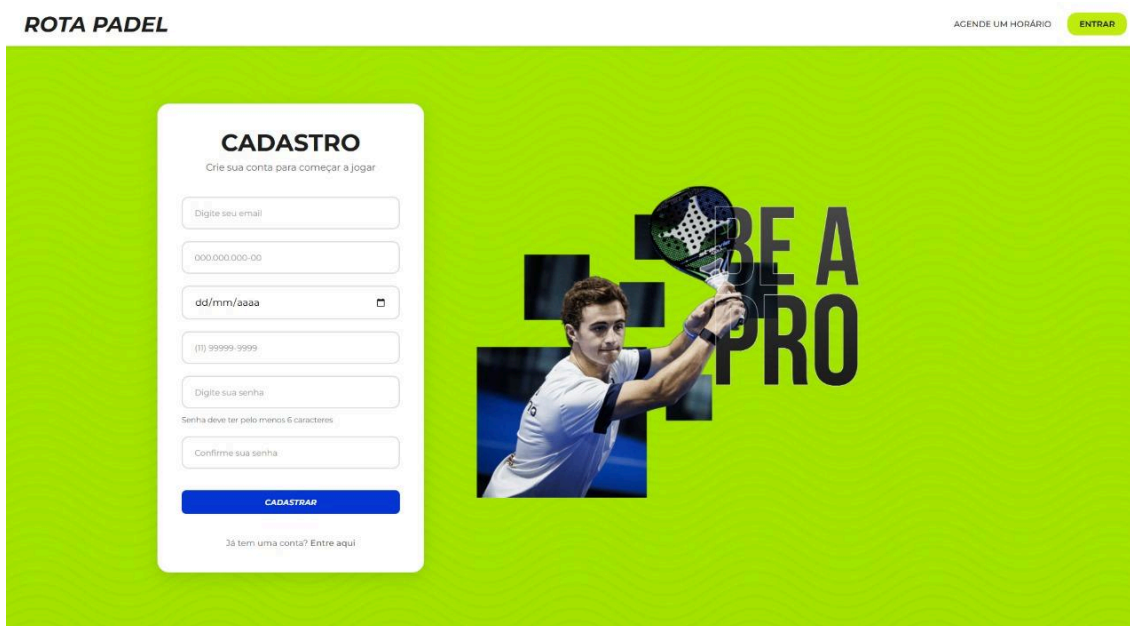


Figura 3 - Tela de cadastro.

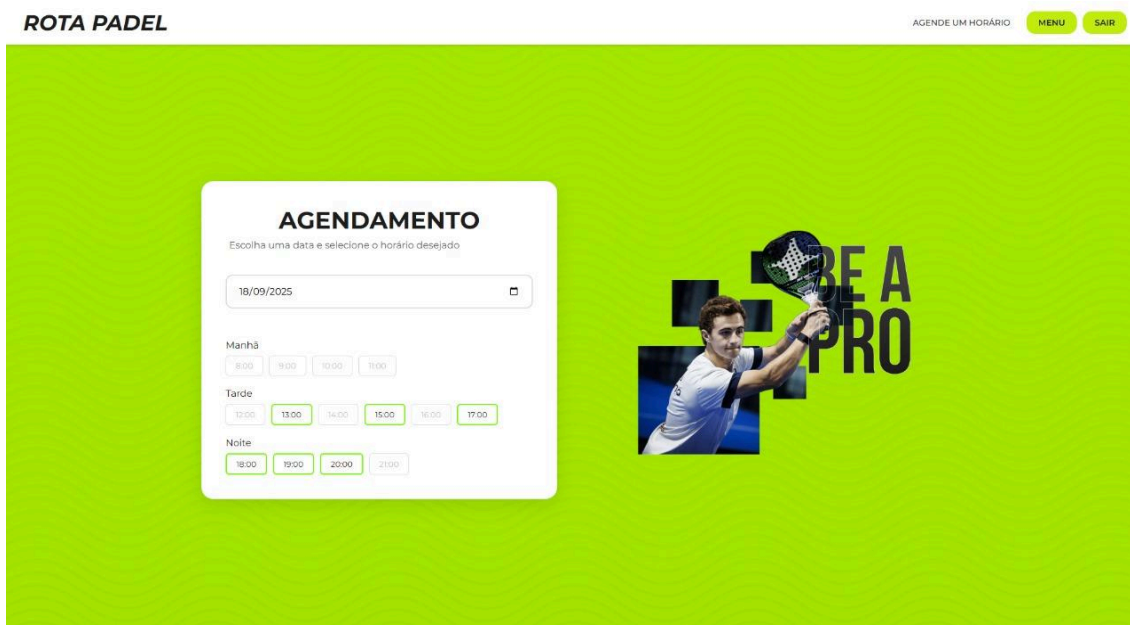


Figura 4 - Tela de agendamentos.

6. Conclusão

O desenvolvimento do sistema digital de agendamento de horários para quadras de padel representa um avanço significativo na modernização da gestão esportiva local. Ao substituir métodos tradicionais e manuais, como o agendamento via WhatsApp, por uma plataforma digital intuitiva e eficiente, conseguimos resolver problemas recorrentes como conflitos de horários, falta de controle e baixa acessibilidade.

A escolha de tecnologias como React.js, JavaScript, Figma, Node.js e PostgreSQL permitiu a criação de uma aplicação web responsiva, segura e de fácil manutenção. A interface foi projetada com foco na experiência do usuário (UX), utilizando uma paleta de cores vibrante e contrastante, aliada a botões de fácil leitura e navegação simplificada. Ferramentas como Figma e Photoshop foram essenciais na prototipação e no design visual do sistema, garantindo consistência e atratividade.

A implementação de funcionalidades como visualização em tempo real da disponibilidade das quadras, agendamento automatizado e notificações push contribuiu para a otimização do processo de reserva, aumentando a comodidade para os usuários e a eficiência para os administradores. Além disso, a plataforma promoveu maior transparência e comunicação entre as partes envolvidas, fortalecendo a confiança e a satisfação dos usuários.

Em termos de impacto, o sistema não só facilitou o acesso ao esporte, tornando-o mais organizado e acessível, mas também abriu oportunidades para a expansão e profissionalização da prática do padel na cidade. A modernização da gestão esportiva local reflete-se na melhoria da experiência dos jogadores, na otimização dos recursos das quadras e no incentivo ao crescimento do esporte na comunidade.

Este projeto demonstra como a tecnologia pode ser uma aliada poderosa na transformação digital de práticas esportivas locais, promovendo eficiência, acessibilidade e engajamento. Acredita-se que iniciativas como esta possam servir de modelo para outras localidades que enfrentam desafios semelhantes na gestão de espaços esportivos.

7. Referências

REACT. React – A JavaScript library for building user interfaces. Facebook (Meta), 2025a. Disponível em: <https://react.dev/>. Acesso em: 17 set. 2025.

REACT. FAQ: React Internals. Facebook (Meta), 2025b. Disponível em: <https://legacy.reactjs.org/docs/faq-internals.html>. Acesso em: 17 set. 2025.

REACT. Introducing Hooks. Facebook (Meta), 2025c. Disponível em: <https://legacy.reactjs.org/docs/hooks-overview.html>. Acesso em: 17 set. 2025.

REACT. Using TypeScript with React. Facebook (Meta), 2025d. Disponível em: <https://react.dev/learn/typescript>. Acesso em: 17 set. 2025.

VERCEL. Next.js – Documentation. Disponível em: <https://nextjs.org/docs>. Acesso em: 17 set. 2025.

VERCEL. Progressive Web Apps with Next.js – Guide. Disponível em: <https://nextjs.org/docs/app/guides/progressive-web-apps>. Acesso em: 17 set. 2025.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL – Documentation. Disponível em: <https://www.postgresql.org/docs>. Acesso em: 17 set. 2025.