



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

IGOR DANTAS LUCENA

**EVOLUÇÃO DO SISTEMA DE AGENDAMENTO DE HORÁRIOS
DO COMPLEXO ESPORTIVO DA UFCG - SAHCE-UFCG**

CAMPINA GRANDE - PB

2023

IGOR DANTAS LUCENA

**EVOLUÇÃO DO SISTEMA DE AGENDAMENTO DE HORÁRIOS
DO COMPLEXO ESPORTIVO DA UFCG - SAHCE-UFCG**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientadora : Professora Dra. Melina Mongiovi Brito Lira.

CAMPINA GRANDE - PB

2023

IGOR DANTAS LUCENA

**EVOLUÇÃO DO SISTEMA DE AGENDAMENTO DE HORÁRIOS
DO COMPLEXO ESPORTIVO DA UFCG - SAHCE-UFCG**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

Melina Mongiovi Brito Lira

Orientador – UASC/CEEI/UFCG

Robert Kalley Cavalcanti de Menezes

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 28 de JUNHO de 2023.

CAMPINA GRANDE - PB

RESUMO

O complexo esportivo da Universidade Federal de Campina Grande (UFCG) atualmente oferece opções de agendamento consideradas mais tradicionais. Os usuários têm a possibilidade de fazer reservas pessoalmente na secretaria do complexo localizada no campus, ou por meio de um aplicativo de mensagens. No entanto, esses métodos são considerados pouco eficientes. Para aprimorar o processo tanto para as comunidades envolvidas quanto para os administradores dos espaços, torna-se necessário o desenvolvimento de uma aplicação online. Uma versão inicial dessa aplicação foi criada no início deste ano, porém ainda não estava disponível para uso e já se fez necessário adicionar novas funcionalidades. Este trabalho documenta o desenvolvimento de uma Progressive Web App (PWA), com ênfase nas funcionalidades essenciais que serão implementadas, destacando-se: recuperação de senha do usuário, controle de reserva, atualização dos perfis, notificações por e-mail e bloqueio de usuários. Essas funcionalidades são consideradas prioritárias e fundamentais para o pleno funcionamento e experiência do usuário no sistema.

EVOLUTION OF THE SCHEDULING SYSTEM FOR THE SPORTS COMPLEX OF UFCG - SAHCE-UFCG

ABSTRACT

The sports complex of the Federal University of Campina Grande (UFCG) currently offers scheduling options that are considered more traditional. Users have the possibility to make reservations in person at the complex's office located on campus, or through a messaging app. However, these methods are considered inefficient. To improve the process for both the communities involved and the space administrators, it is necessary to develop an online application. An initial version of this application was created earlier this year, but it was not yet available for use and there was already a need to add new functionalities. This work documents the development of a Progressive Web App (PWA), with emphasis on the essential functionalities that will be implemented, including: user password recovery, reservation control, profile updates, email notifications, and user blocking. These functionalities are considered prioritized and essential for the smooth functioning and user experience of the system.

Evolução do Sistema de Agendamento de Horários do Complexo Esportivo da UFCG - SAHCE-UFCG

Igor Dantas Lucena
igor.lucena@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande - PB

Melina Mongiovi Brito Lira
melina@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande - PB

RESUMO

O complexo esportivo da Universidade Federal de Campina Grande (UFCG) atualmente oferece opções de agendamento consideradas mais tradicionais. Os usuários têm a possibilidade de fazer reservas pessoalmente na secretaria do complexo localizada no campus, ou por meio de um aplicativo de mensagens. No entanto, esses métodos são considerados pouco eficientes. Para aprimorar o processo tanto para as comunidades envolvidas quanto para os administradores dos espaços, torna-se necessário o desenvolvimento de uma aplicação online. Uma versão inicial dessa aplicação foi criada no início deste ano, porém ainda não estava disponível para uso e já se fez necessário adicionar novas funcionalidades. Este trabalho documenta o desenvolvimento de uma *Progressive Web App* (PWA), com ênfase nas funcionalidades essenciais que serão implementadas, destacando-se: recuperação de senha do usuário, controle de reserva, atualização dos perfis, notificações por e-mail e bloqueio de usuários. Essas funcionalidades são consideradas prioritárias e fundamentais para o pleno funcionamento e experiência do usuário no sistema.

Palavras-chave

Desenvolvimento de software. UFCG. Complexo esportivo. PWA.

Backend: https://github.com/Igordl/SAHCE_UFCG_BACK

Frontend: <https://github.com/Igordl/sahce-ufcg>

1. INTRODUÇÃO

O complexo esportivo é parte integrante da Universidade Federal de Campina Grande (UFCG) e se destina à prática de atividades físicas de diversos tipos. Ele é composto por três quadras de areia de diferentes tamanhos, dois campos gramados e uma quadra de tênis. Tanto a comunidade externa quanto a interna da instituição têm permissão para utilizar esses espaços. Atualmente é possível realizar reserva de horário, porém esses agendamentos só podem ser feitos presencialmente, na secretaria do complexo ou via *whatsapp* (aplicativo de mensagens). É viável utilizar qualquer um dos espaços sem a necessidade de agendamento prévio; no entanto, é importante que o usuário esteja disposto a ceder o local caso esteja reservado e o usuário que efetuou a reserva compareça ao local.

Considerando o tamanho da comunidade e a necessidade de aproveitar as tecnologias disponíveis, é evidente a importância de oferecer um sistema de agendamento mais prático para esses espaços.

É essencial levar em conta que, apesar da existência de vários aplicativos disponíveis nesse segmento, é necessário

desenvolver um específico para atender às necessidades do caso em questão. Essa situação ocorre devido às especificidades relacionadas aos distintos perfis de usuários envolvidos. Esses perfis incluem os administradores, que são os funcionários da secretaria do complexo esportivo, os usuários da comunidade interna, representados pelos alunos e funcionários da instituição, e a comunidade externa, composta por pessoas sem vínculo direto com a universidade. É necessário estabelecer uma distinção entre os perfis da comunidade interna e externa, uma vez que os usuários com vínculo à instituição devem ter prioridade ao agendar um horário.

Com o objetivo de simplificar esses agendamentos, foi desenvolvida uma aplicação no formato de *Progressive Web App* (PWA), que pode ser considerada uma evolução híbrida entre páginas web tradicionais e um aplicativo móvel [1]. Este trabalho tem como objetivo documentar o processo de desenvolvimento dessa aplicação, com foco na facilitação dos agendamentos e na melhoria da experiência dos usuários do complexo.

2. SOLUÇÃO

O SAHCE-UFCG (Sistema de Agendamento de Horários do Complexo Esportivo da UFCG) é uma plataforma inicialmente desenvolvida por outro aluno da instituição e entregue no início deste ano. Devido à necessidade de incluir novas funcionalidades, uma nova versão da aplicação foi desenvolvida.

2.1 Funcionalidades

2.2.1 Recuperação de senha

Por meio do botão disponível na tela de login, é possível cadastrar uma nova senha. Ao clicar neste botão, um token será enviado para o e-mail do usuário. O usuário deve então preencher esse token na tela de recuperação e cadastrar a nova senha.

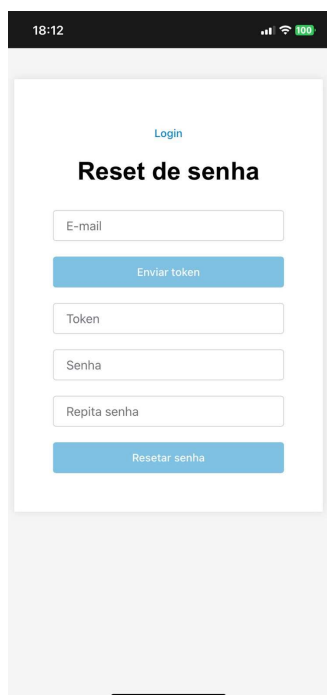


Figura 1 -Página de recuperação de senha.

2.2.2 Controle de reserva

Caso o usuário já possua uma reserva ativa ou já esteja na lista de reserva de outro usuário, não será possível realizar um novo agendamento pois no back-end a regra de listagem de espaços para reserva apresenta somente locais em que ele possa realizar a reserva. Essa funcionalidade é fundamental para evitar a monopolização dos espaços para um mesmo grupo.

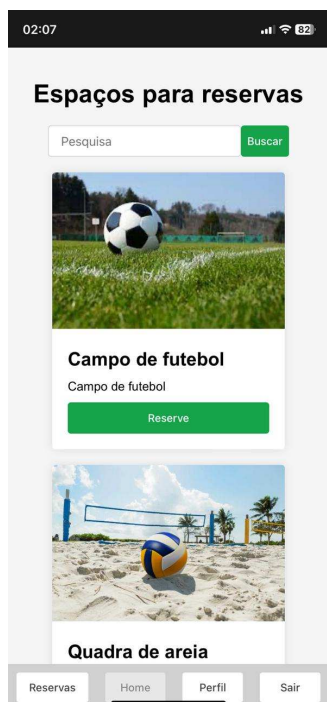


Figura 2 - Dashboard do usuário.

2.2.3 Atualização dos perfis

A cada semestre, a aplicação requer que os usuários atualizem seus cadastros, permitindo assim o controle dos perfis. Por exemplo, se um aluno se formou, ao atualizar seu cadastro, seu perfil será alterado de "comunidade interna" para "comunidade externa". Isso garante a atualização adequada dos perfis de acordo com as mudanças na situação dos usuários.

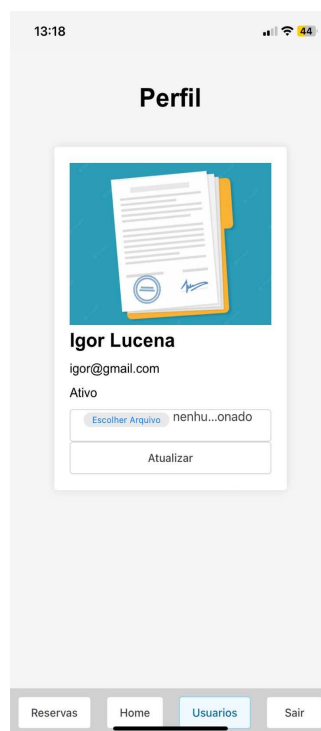


Figura 3 - Página de perfil do usuário.

2.2.4 Notificação por email

Quando um horário é cancelado devido à inadequação do espaço para uso, o usuário receberá uma notificação por e-mail. Isso permite que o usuário seja informado sobre o cancelamento e possa tomar as medidas necessárias em relação ao seu agendamento.

2.2.5 Bloqueio de usuários

Se um usuário violar as regras do complexo esportivo, um perfil de administrador terá a capacidade de bloquear o acesso desse usuário, impedindo-o de fazer novos agendamentos. Da mesma forma, é possível desbloquear um usuário caso a situação seja resolvida ou sejam aplicadas medidas disciplinares apropriadas.

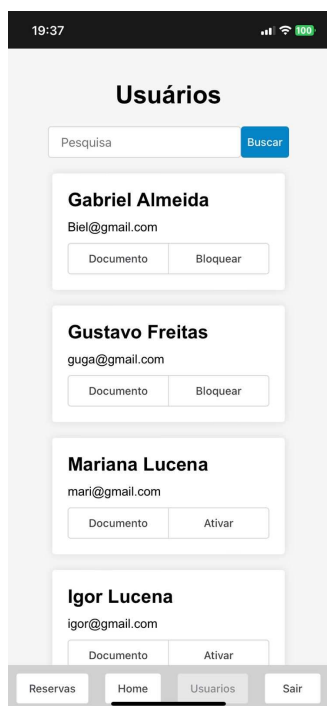


Figura 4 - Dashboard do administrador.

2.2.6 Aprovação de cadastro

Cabe aos perfis de administradores do sistema a responsabilidade de aprovar o cadastro de novos usuários, o que inclui a verificação da matrícula do usuário em questão para garantir que ele realmente pertence à instituição. Essa etapa de aprovação é importante para assegurar a validade e a autenticidade dos cadastros no sistema.

2.2.7 Priorização de reserva

O objetivo dessa funcionalidade é priorizar os usuários pertencentes à comunidade interna no agendamento de horários. Alguns espaços específicos, como a quadra de beach tênis, serão disponibilizados para reserva pelos perfis da comunidade externa somente a partir da quinta-feira. Isso garante que os usuários internos tenham prioridade no agendamento dos espaços durante os primeiros dias da semana.

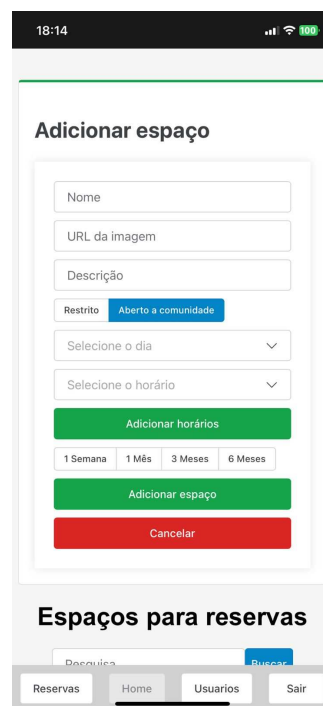


Figura 5 - Dashboard de adição de espaços do administrador.

2.2.8 Remarcação de horário

É necessário respeitar a restrição de não permitir que um mesmo usuário tenha dois horários ativos ao mesmo tempo. Para agendar um novo horário, o usuário deve primeiro cancelar qualquer agendamento ativo ou aguardar o término do horário previamente marcado. Essa medida visa evitar sobreposições e garantir que cada usuário tenha apenas um horário ativo por vez.

2.2 Arquitetura

A arquitetura Cliente-Servidor é um dos modelos mais amplamente utilizados na construção de sistemas web. Nessa arquitetura, o cliente é responsável por solicitar informações ou serviços, enquanto o servidor é encarregado de fornecer essas informações ou serviços solicitados.

A comunicação entre o cliente e o servidor ocorre por meio de um protocolo de comunicação, como o HTTP (Hypertext Transfer Protocol) [4], que permite a troca de mensagens entre as partes. O Cliente envia uma requisição ao Servidor, contendo os parâmetros necessários para sua solicitação, como ação desejada e dados relevantes.

O Servidor, por sua vez, recebe a requisição do Cliente e processa as informações recebidas. Isso pode envolver acesso a bancos de dados, execução de lógica de negócios, processamento de algoritmos, entre outras tarefas. Após o processamento, o Servidor envia uma resposta ao Cliente, contendo os resultados da solicitação.

Essa arquitetura é caracterizada pela divisão clara de responsabilidades entre o Cliente e o Servidor. O Cliente é responsável pela interface com o usuário e pela apresentação dos dados recebidos do Servidor. Já o Servidor é responsável por lidar com a lógica de negócios, processar as requisições e fornecer as respostas adequadas.

2.3 Back-end

O backend desempenha um papel crucial nas interações com o banco de dados, assumindo responsabilidade pela inclusão,

remoção e atualização dos dados, bem como pela seleção e tratamento das informações a serem enviadas ao cliente. Para facilitar a comunicação entre essas entidades, o servidor foi construído seguindo o padrão REST API [4], reconhecido por seu design moderno e simplificado.

O backend passou por uma reestruturação, destacando-se os módulos de controller e domínio. Os controllers compõem um conjunto de módulos que contêm as funções responsáveis por lidar com as chamadas feitas à API. Nesse contexto, o controller assume a responsabilidade de tratar os dados presentes nas requisições, delegar qualquer processamento ou consulta necessária ao serviço correspondente e, em seguida, tratar os dados recebidos e retornar as informações ao cliente. No contexto do domínio, é incumbida a tarefa de definir os endpoints da API, estabelecendo as rotas e recursos disponíveis para acesso externo. Nessa definição, são especificados os métodos HTTP adequados para cada endpoint, como GET, POST, PUT ou DELETE, que determinam as ações a serem executadas sobre os dados. Outro aspecto relevante é a associação entre os endpoints e seus respectivos controllers. O domínio mapeia cada rota para o controller adequado, que será responsável por processar a requisição, realizar as operações necessárias no banco de dados ou em outras camadas do sistema, e retornar as informações correspondentes ao cliente.

2.4 Front-end

O desenvolvimento do frontend foi focado na acessibilidade e na usabilidade para os três tipos de usuários. Utilizou-se a biblioteca JavaScript React [6] para criar as interfaces de usuário. A escolha dessa tecnologia baseou-se na sua capacidade de encapsular componentes de maneira intuitiva e próxima ao JavaScript.

Foi utilizado o scaffolding do PWABuilder, que é um conjunto de arquivos e estrutura de diretórios inicial fornecido pelo PWABuilder para facilitar o desenvolvimento de Progressive Web Apps (PWAs). Ele inclui uma base de código predefinida, recursos essenciais e uma estrutura organizacional para iniciar o desenvolvimento de um PWA de forma mais rápida e eficiente [7].

A biblioteca de estilo *Shoelace*[10] utilizado, trata-se de uma biblioteca de componentes de interface de usuário (UI) prontos para uso, desenvolvida em HTML, CSS e TypeScript. Essa biblioteca oferece uma coleção de componentes estilizados e responsivos, como botões, menus, formulários, cartões e muito mais, que podem ser facilmente integrados em projetos web. O Shoelace visa fornecer componentes de alta qualidade, personalizáveis e de fácil utilização, ajudando os desenvolvedores a criar interfaces atraentes e funcionais de maneira rápida e eficiente.

Existem dois principais benefícios ao utilizar uma PWA. O primeiro é a capacidade de ser multiplataforma de forma simples, permitindo que seja instalada em diversos dispositivos como uma aplicação nativa. Isso proporciona uma experiência consistente para os usuários, independentemente do sistema operacional que estão utilizando. O segundo benefício diz respeito ao uso de memória do dispositivo. As PWA tendem a ocupar um espaço menor em comparação aos aplicativos nativos, o que é vantajoso para os usuários que têm restrições de armazenamento em seus dispositivos. Além disso, as PWA não exigem a distribuição através das lojas de aplicativos, como a App Store, Play Store ou Microsoft Store. Isso resulta em uma redução de custos, uma vez que não é necessário pagar taxas ou passar pelo processo de aprovação das lojas. Em vez disso, o custo está relacionado apenas à aquisição de um domínio e hospedagem, o que é consideravelmente mais acessível em comparação à publicação nas lojas de aplicativos mencionadas.

2.4.1 Instalando a PWA

2.4.1.1 Requisitos:

- Dispositivo móvel ou computador com acesso à internet.
- Navegador web compatível (ex.: Google Chrome, Mozilla Firefox, Safari).

2.4.1.2 Acesso à PWA:

- Digite o URL fornecido pela PWA no navegador e acesse o site.

2.4.1.3 Instalação em dispositivos móveis (Android):

- Toque no ícone de menu do browser e toque em adicionar para tela inicial.

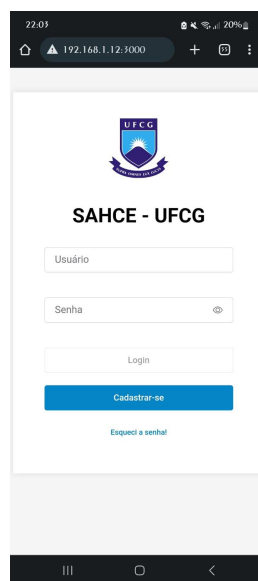


Figura 6 - Tela inicial da aplicação em dispositivo Android.

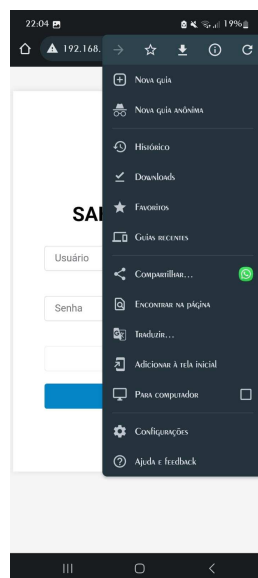


Figura 7 - Menu do browser do dispositivo Android.

- Confirme o nome que deseja atribuir a aplicação e toque em adicionar.



Figura 8 - Tela de edição e confirmação de nome no dispositivo Android.

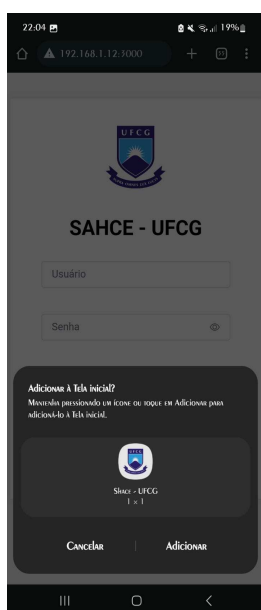


Figura 9 - Tela de confirmação de ícone no dispositivo Android.

- A PWA será instalada automaticamente.
- O atalho estará disponível na tela inicial. Ao tocar no ícone a aplicação abre como um aplicativo nativo do sistema

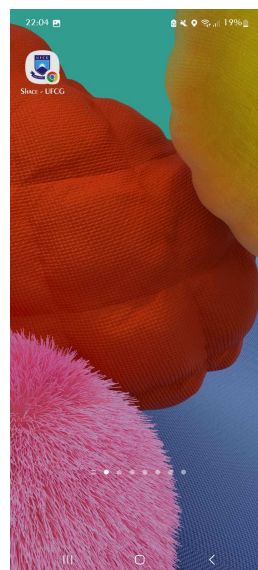


Figura 10 - Tela inicial do dispositivo Android.

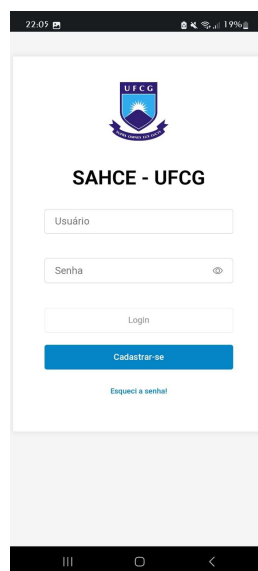


Figura 11 - Tela inicial da aplicação após instalada no dispositivo Android.

2.4.1.4 Instalação em dispositivos móveis (iOS):

- Toque no ícone de compartilhamento e selecione "Adicionar à Tela de Início".



Figura 12 - Tela inicial da aplicação web após instalada no dispositivo iOS.

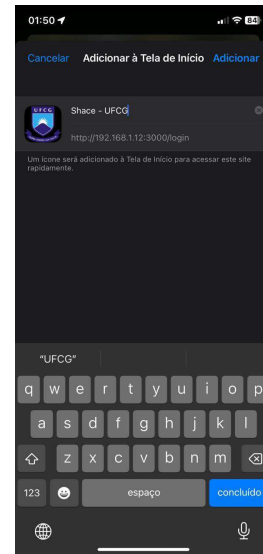


Figura 14 - Tela de edição e confirmação no dispositivo iOS.

- A PWA será adicionada à tela inicial.



Figura 13 - Tela de compartilhamento do dispositivo iOS.

- Personalize o nome e confirme a adição.



Figura 15 - Tela inicial do dispositivo iOS.



Figura 16 - Tela inicial da aplicação web após instalada no dispositivo Android.

2.4.1.5 Instalação em computadores (Google Chrome):

- Clique no ícone de instalação na barra da URL do browser ou clique no ícone de menu (três pontos) e selecione "Instalar [nome do aplicativo]".

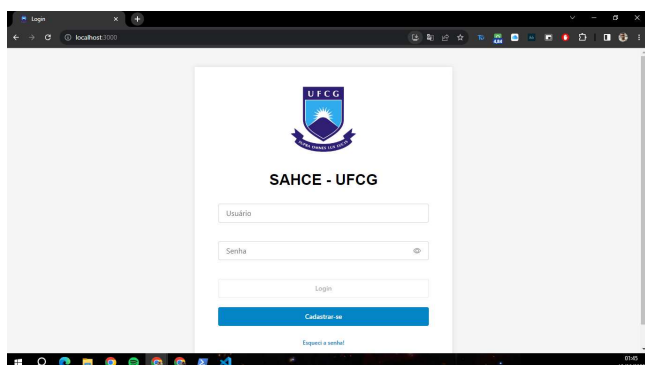


Figura 17 - Tela inicial da aplicação web no desktop.

- Confirme a instalação.

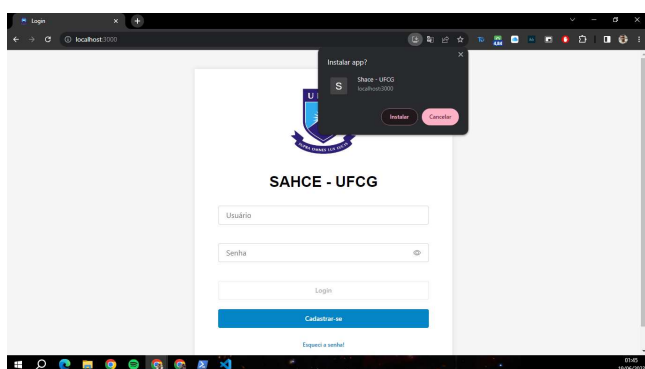


Figura 18 - Tela de confirmação de instalação no desktop.

- A PWA será baixada e instalada automaticamente.
- O atalho estará disponível na área de trabalho ou na barra de tarefas.

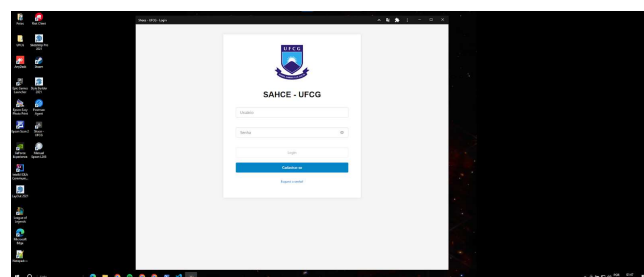


Figura 19 - Tela da aplicação após instalada no desktop.

Com a pré-configuração do Vite, uma ferramenta de construção de aplicativos da web é extremamente rápida e leve. Ela foi projetada para desenvolvimento moderno de front-end e fornece uma experiência de desenvolvimento altamente produtiva [6], e a utilização do LitElement, biblioteca JavaScript que permite a criação de componentes web reutilizáveis e baseados em padrões modernos da Web, como Web Components. O Vite é flexível o suficiente para suportar diferentes estruturas e bibliotecas.

3. METODOLOGIA

Inicialmente, foi necessário identificar os novos requisitos essenciais para aprimorar a aplicação existente. Embora a aplicação existisse, ainda não estava pronta para uso. Por meio de uma conversa entre o autor e os funcionários da secretaria do complexo esportivo, foi possível listar as principais necessidades.

Com os requisitos claramente estabelecidos, a ferramenta escolhida para priorizar as funcionalidades foi o Trello. O Trello é uma plataforma de gerenciamento de projetos online que utiliza um sistema visual de quadros, listas e cartões para organizar tarefas e monitorar o progresso delas. Dessa forma, foi possível criar um backlog, que é uma lista priorizada de funcionalidades, requisitos, melhorias e correções que devem ser desenvolvidas no produto [2], e iniciar o desenvolvimento.

4. EXPERIÊNCIA

O conhecimento de diversas tecnologias, conceitos e habilidades adquiridas ao longo da graduação foi fundamental para o desenvolvimento deste trabalho. Algumas disciplinas foram especialmente relevantes nesse processo, incluindo Programação II, Laboratório de Programação II, Banco de Dados I, Engenharia de Software e Metodologia Científica.

4.1 Desenvolvimento

No início, foram identificadas as funcionalidades essenciais do sistema e incluídas no backlog. Em seguida, cada funcionalidade recebeu uma prioridade (baixa, média ou alta) e o backlog foi organizado de acordo, priorizando as funcionalidades de maior importância. Dentro desse contexto, as principais funcionalidades foram: recuperação de senha, controle de reservas, atualização dos perfis, notificação por e-mail e bloqueio de usuários.

As tecnologias a serem utilizadas foram definidas e foram criados repositórios no GitHub [3] para o backend e o frontend das aplicações. Assim, o processo de desenvolvimento teve início, seguindo a abordagem de primeiro realizar cada tarefa no backend e depois no frontend.

4.2 Desafios

A conclusão deste trabalho enfrentou dois principais desafios. O primeiro deles foi a implementação do frontend, que envolveu o aprendizado de uma nova tecnologia. Foi necessário dedicar tempo para adquirir o conhecimento necessário. O segundo desafio foi a refatoração do backend, visando simplificar sua estrutura. Ambos os desafios impactaram o cronograma do projeto em relação ao tempo.

5. CONCLUSÃO

Como fruto deste trabalho, desenvolvemos uma aplicação web no formato PWA (Progressive Web Application), com todas as funcionalidades já existentes na primeira versão do sistema, além de agregar novas funções.

O desenvolvimento dessa aplicação como PWA proporcionou uma experiência multiplataforma para essa nova versão, expandindo seu alcance além do sistema Android presente na versão anterior. Assim, os usuários podem desfrutar plenamente do sistema em dispositivos iOS, desktop e Android.

A instalação da aplicação é simplificada, e oferece aos usuários a conveniência de economizar espaço de memória em seus dispositivos. Além disso, os custos de publicação da aplicação são notavelmente reduzidos, visto que é necessário apenas implantar a aplicação web em um servidor próprio da instituição e registrar um domínio no DNS (Domain Name System).

Dessa forma, unimos funcionalidade e acessibilidade em uma solução moderna e eficiente, agregando valor tanto para os usuários quanto para a instituição.

6. REFERÊNCIAS

- [1] FERNANDES, Diego. Programador full-stack, front-end e mobile. Co-fundador e CTO na Rocketseat. (2018)
- [2] SUTHERLAND, J. (2010). The Scrum Guide. Scrum.org. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>. Acesso em: 20 de maio de 2023.
- [3] GITHUB. [n. d.]. Github. Retrieved Maio 20, 2023 from <https://github.com/>
- [4] MDN Web Docs. [n. d.]. Uma visão geral do HTTP. Retrieved Maio 25, 2023 from
- [5] Disponível em: <https://vitejs.dev/>. Acesso em: 30 de maio de 2023.
- [6] Facebook. [n. d.]. React: uma biblioteca JavaScript para criar interfaces de usuário. Retrieved Maio 25, 2023 from <https://pt-br.reactjs.org/>
- [7] PWAbuilder. Site oficial do PWAbuilder. Disponível em: <https://www.pwabuilder.com/scaffold>. Acesso em: 25 de maio de 2023
- [8] Disponível em: <https://vitejs.dev/>. Acesso em: 30 de maio de 2023.
- [9] Red Hat. 2020. O que é API REST? Retrieved Maio 25, 2023 from <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>
- [10] Shoelace. Site oficial do Shoelace. Disponível em: <https://shoelace.style/>. Acesso em: 30 de maio de 2023.