

Functional High-fidelity Prototype of a Student Meal Scheduling and Cafeteria Assistance System

Protótipo Funcional de Alta Fidelidade de um Sistema de Agendamento de Refeições para Estudantes e Assistência à Cantina

Gregório Cândido dos Santos Valadares de Almeida¹ , Monck Charles Nunes de Albuquerque¹ , Nataly de Souza Cunha¹ 

¹Instituto Federal de Educação, Ciência e Tecnologia da Bahia – Campus Seabra
Seabra, Bahia – Brazil

{gregcandido555, monckcna, natalycunha005}@gmail.com

Abstract. *At the Federal Institute of Bahia - Campus Seabra, many students request school meals via a form on Google Forms, the platform on which the Cafeteria depends to monitor requests, experiencing difficulty with manual processes and non-targeted functionalities. Thus, the objective of this work was the creation of a prototype to assist these demands and implement autonomy of the school. To do this, these steps were taken: define the app's requirements and elements; technical training; building and testing, and professional validation. It was built a functional and intuitive high-fidelity system, ensuring a more comprehensive user experience and turning into reality a new solution for the Cafeteria's meal requests.*

Keywords. *Progressive-web-app; Management system; Information system; Meal booking software.*

Resumo. *No Instituto Federal da Bahia - Campus Seabra, vários alunos solicitam a refeição escolar por um formulário do Google Forms, plataforma da qual o refeitório depende para monitorar as inscrições, apontando dificuldade nos processos manuais e as funcionalidades não direcionadas. Assim, o objetivo deste trabalho foi a criação de um protótipo para gerir essas demandas e estabelecer maior autonomia para esta instituição. Foram feitos: definição de requisitos e elementos do app, capacitação técnica, construção, testagem e validação profissional. Obteve-se um sistema de alta fidelidade funcional e de design intuitivo, garantindo uma experiência de usuário fluida e tornando realidade uma nova solução para a gerenciamento de pedidos do Refeitório.*

Palavras-Chave. *Progressive-web-app; Sistema de gerenciamento; Sistema de informação; Software de agendamento de refeição.*

1. Introdução

A inovação tecnológica, em muitos casos, envolve a criação de novas ferramentas e soluções. No entanto, seu significado vai além da simples construção de novos mercados e invenções — também se refere à elaboração de novas formas de atender tanto a mercados já estabelecidos e maduros, quanto a mercados inéditos [Tidd and Bessant 2015].

Além disso, inovações tecnológicas desempenham um papel fundamental na facilitação de gerenciamento dos processos na comunidade estudantil, pois têm a capacidade de oferecer uma gama de ferramentas e soluções que tornam as atividades estudantis mais eficientes, acessíveis e engajadoras. Através de aplicativos, plataformas *online* e recursos digitais, estudantes podem acessar informações relevantes e colaborar de forma remota. Outrossim, a criação de aplicações permite o auxílio de tarefas rotineiras e a automatização de processos, liberando tempo e recursos para maior foco em outras demandas [Aragão et al. 2020, Teodosio et al. 2021].

Nas instituições de ensino técnico brasileiras, parte dos estudantes comparecem de forma integral, seja por motivos de atividades acadêmicas, transporte escolar, estudos individuais ou outras razões. No IFBA Campus Seabra, há um Refeitório Estudantil que oferece almoço diário gratuito, sendo administrado por uma nutricionista.

Para garantir a refeição do dia, cada aluno deve preencher um formulário *online* de confirmação para o almoço, criado pela própria instituição através da plataforma *Google Forms*¹. Nele, há o espaço de preenchimento básico de identificação e o cardápio semanal. Segundo registros deste formulário diário, a média de estudantes que costumam almoçar é de 120 a 150 por dia. Para entender esse quantitativo, os servidores responsáveis precisam, manualmente, fechar o formulário, fazer a contagem e gerar uma lista impressa com todos os alunos cadastrados, previamente ao horário do almoço, de forma a utilizá-la como método de autenticação para recebimento da refeição. Ademais, o *Google Forms* não performa como uma ferramenta direcionada para as demandas processuais do refeitório, carecendo de funções específicas como filtros automatizados, confirmação de presença e exportação de relatório estilizado; além disso, dado que a instituição depende especialmente da existência e integridade dessa plataforma, nos casos do *Forms* apresentar momentos de mau funcionamento, manutenção ou deixar de existir, haveria total interferência na plena organização diária das refeições.

Assim, o objetivo deste trabalho foi criar um protótipo de alta fidelidade funcional, prático e disponível para todos os alunos e servidores por meio da rede local. O aplicativo será utilizado para as solicitações de interesse à refeição estudantil, além de fornecer um espaço de administração destes registros para a gestão do refeitório, de forma a garantir eficiência nos processos e autossuficiência para a instituição.

Para isso, juntamente ao refeitório do IFBA Campus Seabra, foi realizado uma investigação inicial dos processos de preenchimento de formulário e administração destes registros, o que norteou o levantamento de requisitos, casos de uso e elementos do aplicativo. Após isso, foi feita uma capacitação teórica sobre aplicativos de natureza *progressive-web-app* (PWA), *framework Laravel* e os fundamentos da programação,

¹Google Forms: <https://www.google.com/intl/pt-BR/forms/about/>

banco de dados e sistemas — especialmente as operações CRUD (*Create, Read, Update, Delete*). Então, fez-se a construção do protótipo e, de forma concomitante, a sua testagem, seguindo-se para a validação profissional do sistema pela nutricionista e o servidor auxiliar do refeitório.

2. Fundamentação teórica

Nesta seção serão apresentados os principais conceitos necessários para a compreensão da temática e a realização do protótipo.

2.1. Alimentação Escolar

A alimentação escolar refere-se ao fornecimento de refeições/lanches nutritivos e balanceados em ambientes educacionais, de forma a garantir que, apesar de fatores psicológicos, sociais e/ou culturais, os estudantes recebam uma nutrição adequada para auxiliar no seu crescimento físico e cognitivo, bem como garantir melhor desempenho acadêmico durante o dia a dia escolar [Marcondes 1972, Pedraza et al. 2018].

Os programas de alimentação escolar podem ser implementados através de iniciativas governamentais ou educacionais, e visam fornecer refeições saudáveis e equilibradas, contendo nutrientes fundamentais para a fase de desenvolvimento dos estudantes no período letivo [Pedraza et al. 2018].

2.2. Aplicativo/app

Um aplicativo — também denominado de aplicação ou *app* — consiste em um programa de computador desenvolvido com, normalmente, mais de uma linguagem, de forma a oferecer um conjunto de funcionalidades. Possui o objetivo de cumprir uma ou mais necessidades gerais ou específicas de usuários prospectados [Nonnenmacher 2012].

Os *apps* são programas extremamente disseminados e demandados no meio comercial, com o objetivo de integrarem um conjunto de serviços para atender necessidades de clientes de uma empresa, dos funcionários de uma corporação, entre vários casos. Além disso, podem ser desenvolvidos para funcionarem em diversas plataformas, como *desktops*, *smartphones*, *tablets*, entre outras [Nonnenmacher 2012, Pressman and Maxim 2021].

Dentre os vários tipos de *software* — tais como o básico, de tempo real, científico, entre outros —, uma aplicação comercial também pode ser entendida como um sistema de informação, já que contempla o interesse de empresas em manter controle sobre as informações que elas produzem e consomem [Wazlawick 2019].

2.3. Engenharia de Software

A Engenharia de *Software* contempla a abrangência de métodos, práticas e processos relacionados à criação e manutenção de sistemas de *software* de alta qualidade. Ela envolve a aplicação sistemática de abordagens científicas e tecnológicas para o desenvolvimento de aplicações, incluindo comunicação, análise de requisitos, modelagem, projeto, codificação, teste, implantação e a manutenção de sistemas [Pressman and Maxim 2021].

2.3.1. Requisitos de *Software*

Os requisitos de *software* são descrições formais e informais das funcionalidades e restrições que um sistema de *software* deve atender. Eles representam as necessidades, expectativas e objetivos dos *stakeholders* (partes interessadas) que vão utilizar ou interagir com o sistema. Os requisitos podem ser divididos em funcionais, que descrevem as funcionalidades que o sistema deve executar, e não funcionais, que descrevem as restrições que o sistema deve obedecer, como: requisitos de desempenho, segurança, usabilidade, entre outros [Martins 2001, Sommerville and Sawyer 1997].

Requisitos são conceitos fundamentais para a Engenharia de *Software*, devendo ser considerados com atenção e rigor ao longo do processo de desenvolvimento e revisão de uma aplicação, pois, o sucesso de um projeto depende também da análise cuidadosa destes fatores. Deve-se levar em conta as necessidades dos clientes, as restrições técnicas e operacionais que o aplicativo deve atender, e as características do ambiente em que o sistema será utilizado, buscando sempre aprimorar a qualidade dos sistemas e garantindo que eles atendam aos requisitos e expectativas dos usuários — sendo, assim, confiáveis, eficientes, seguros e fáceis de manter [Pressman and Maxim 2021, Sommerville and Sawyer 1997].

2.4. Prototipação

Um protótipo consiste em uma maneira eficiente de representar um conceito através da retenção de suas propriedades mais importantes, ou seja, trata-se de uma prévia mais leve e simplificada de um produto final, mantendo-se suas características essenciais e relevantes, e proporcionando aprimoramento e agilidade nos processos de desenvolvimento de *software* [Floyd 1984].

A prototipagem é o processo de criação de um sistema funcional já nos primeiros estágios de desenvolvimento de um produto, comumente com as finalidades de: facilitação de avaliação de sua aparência, funcionalidade e usabilidade antes da produção robusta; levantamento e/ou validação de requisitos de *software*, tornando possível uma flexibilidade de alterações conforme as mudanças definidas por clientes — diminuindo os custos, atrasos de entrega e reestruturação do trabalho; treinamento de usuário; testes de sistema, entre outros objetivos [Floyd 1984].

Nessa etapa, resgata-se os requisitos e desejos de um projeto final e transforma-se em uma prévia palpável, possibilitando a avaliação de componentes e métodos selecionados pelos desenvolvedores. Pode ser feita em diferentes níveis de fidelidade, desde prévias de baixa fidelidade — que geralmente são esboços ou modelos simples feitos à mão — até modelos de alta fidelidade — versões mais completas e precisas do produto final [Wessel et al. 2022].

De acordo com a apresentação de slides de [Paulista 2005], o processo de desenvolvimento de um protótipo encontra-se demonstrado nas etapas da Figura 1:

Para ser realizada uma prototipação de *software*, existem diferentes abordagens:

- Modelo em cascata: uma fase deve ser terminada para a outra começar, composta pelos passos: requisitos e análise - projeto de sistemas e de *software* -

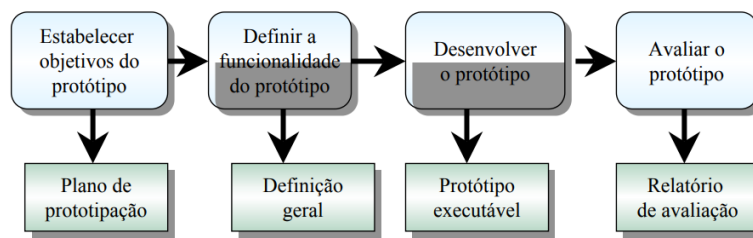


Figura 1. Processo de desenvolvimento de protótipo
[Paulista 2005]

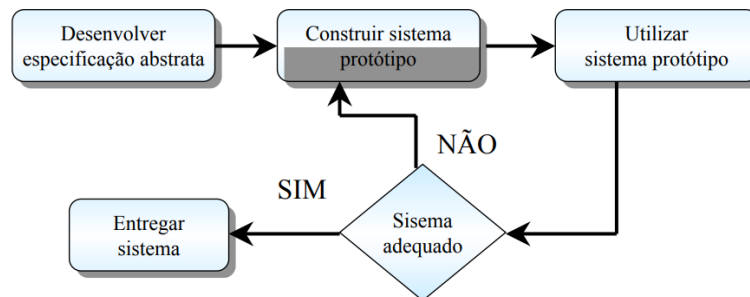


Figura 2. Prototipação evolucionária
[Paulista 2005]

implementação - testes - operação e manutenção;

- Modelo incremental: atividades são intercaladas para possibilitar *feedback* rápido ao cliente, composta pelos passos: definição de escopo - especificação, desenvolvimento e validação em paralelo ao desenvolvimento das versões inicial, intermediárias e final;
- Modelo espiral: resgata elementos dos modelos Incremental e de Cascata, composta pelos passos: *loop* entre definição de requisitos, projeto, implementação e testes, nessa ordem;
- Modelo evolucionário: implementação inicial, obtendo um retorno por parte dos usuários e fazendo seu aprimoramento através de várias versões do protótipo [Pressman and Maxim 2021, Sommerville and Sawyer 1997].

2.4.1. Modelo Evolucionário

Especificamente no modelo evolucionário (Figura 2), em vez das atividades de especificação, desenvolvimento e validação serem feitas separadamente, todas essas etapas são feitas simultaneamente, contando-se com um rápido *feedback* por meio dessas atividades, assim gerando várias versões de protótipo até o alcance da validação e criação da versão final [Pressman and Maxim 2021].

Entre as vantagens do modelo evolucionário, estão: atendimento rápido às necessidades urgentes de clientes, além de sua especificação poder ser desenvolvida gradativamente, conforme a compreensão melhor dos clientes e seus problemas [Pressman and Maxim 2021].

Entre os riscos da abordagem, encontram-se: inviabilidade da produção de documentos que reflitam cada versão do sistema, já que sempre estão se modificando, além de ser possível a não compreensão da natureza da abordagem e a insatisfação com o protótipo ao decorrer do processo [Kalinowski et al. 2023].

2.5. Modelagem conceitual de banco de dados

Trata-se da primeira etapa na elaboração de um banco de dados a ser associado a um *software*, permitindo consultas e atualizações dos registros. Nesse passo, ocorre um levantamento e análise dos requisitos de dados do banco, bem como a sua replicação visual, permitindo um planejamento mais palpável da estrutura do banco de dados. Para a representação aplicada neste trabalho, é utilizado o Modelo Lógico-Relacional, um modelo de dados realístico em relação ao funcionamento e organização das informações [Elmasri et al. 2005].

2.5.1. Modelo Lógico-Relacional de Banco de Dados e Diagramas Relacionais

Dentro de um projeto de banco de dados, existem diferentes etapas de planejamento com respectivas complexidades e características, como por exemplo: modelos Conceitual, Lógico e Físico [Heuser 2009, Ramakrishnan and Gehrke 2003].

Especificamente sobre o Modelo Lógico, representado pelo Diagrama Lógico-Relacional, consiste em uma representação abstrata e estruturada dos dados, que descreve a organização e os relacionamentos entre as entidades (tabelas) que compõem o banco de dados. Ele define as regras e restrições que determinam como os dados serão integrados. O modelo lógico é independente do sistema de gerenciamento de banco de dados (SGBD) utilizado e é projetado para ser compreendido por usuários e desenvolvedores, facilitando o planejamento, a implementação e a manutenção do banco de dados [Heuser 2009, Ramakrishnan and Gehrke 2003].

Dessa forma, o Modelo Lógico é uma maneira estruturada e completa de representar a parte conceitual de um banco de dados, definindo especificamente suas tabelas, atributos, tipos e armazenamentos, bem como seus relacionamentos e cardinalidades [Heuser 2009, Ramakrishnan and Gehrke 2003].

2.6. Tecnologias e linguagens utilizadas

Nesta subseção, serão expostos os recursos digitais e linguagens utilizados para a programação do protótipo de sistema.

2.6.1. Figma

O Figma é uma ferramenta bastante implementada em processos de desenvolvimento *web* e de sistemas, especificamente sobre os passos de planejamento de *design*, experiência e interface de usuário, bem como prototipação. Suas produções podem ser disponibilizados através de código em CSS, possibilitando serem diretamente replicados em uma programação [Figma 2023, Putra et al. 2021].

2.6.2. Framework

No contexto da Engenharia de *Software* e da programação, os *frameworks* (ou bibliotecas) tratam-se de estruturas de classes que constituem implementações pré-estabelecidas e que podem originar diferentes artefatos de *software*. Na prática, tratam-se de reuniões de códigos para cada funcionalidade ou pacote, com cada um de seus arquivos oferecendo um *script* para utilização [Silva 2000].

Os *frameworks* possuem vários propósitos, como: aprimoramento de qualidade, flexibilidade e velocidade de codificação por meio da reutilização de métodos — implicando na redução drástica da reescrita de código-fonte; estimulação de padronização na codificação de projetos, através da utilização protocolada de classes e funções; além de, por fim, haver o oferecimento de diversos recursos de uso comum, como: validações, formulários, integrações com bancos de dados e outros recursos prontos ou semi-prontos [Gabardo 2017].

2.6.3. HTML

O HTML (*HyperText Markup Language*) — ou linguagem de marcação de hipertexto, em português —, em definição, trata-se de uma linguagem de marcação utilizada para estruturar o conteúdo de páginas *web*, tais como textos, formulários, documentos que podem conter *links* para outros documentos, imagens, vídeos e outros tipos de mídia. Ela também possui capacidade para implementação do CSS, que é uma linguagem de estilização da apresentação do conteúdo da página em questão de *design* e *layout* [Silva 2019].

Além disso, o HTML também é eficiente para a acessibilidade, pois permite que o conteúdo seja traduzido em texto, podendo, dessa forma, ser interpretado mais facilmente por pessoas com deficiências visuais ou que utilizam tecnologias assistivas [MacDonald 2011].

2.6.4. JavaScript

JavaScript é uma linguagem de programação de alto nível, com natureza interpretada e orientada a objetos, amplamente utilizada para o desenvolvimento de aplicações *web* interativas. Essa linguagem faz parte da “tríade de tecnologias que todo desenvolvedor *web* deve conhecer”, assumindo o papel de definir o comportamento das funcionalidades presentes nos códigos [Flanagan 2004].

2.6.5. *Laravel*

O *framework Laravel* trata-se de uma biblioteca de código aberto para aplicações web modernas e escaláveis, disponibilizando uma vasta gama de componentes para desenvolvimento, possuindo funcionalidades pré-disponibilizadas para programação *front-end*, *back-end*, conexão com banco de dados, autenticação de usuários, entre inúmeras funções. Sua linguagem principal é o PHP, entretanto, possui suporte para várias outras linguagens como HTML, JavaScript, SQL, entre outras [Laravel 2023].

O *framework* segue o padrão arquitetural MVC (*Model-View-Controller*), o que ajuda a separar a lógica de negócios, a interface do usuário e a manipulação de dados em componentes distintos. Atualmente, o Laravel se encontra na sua décima primeira versão, possuindo uma documentação *online* completa e diversificada para orientação e possibilidades de seu uso. Suas estruturas de código são completas e visualmente ergonômicas, havendo separação de diferentes partes e funções por cores [Laravel 2023].

2.6.6. *My SQL*

O MySQL é um popular sistema de gerenciamento de banco de dados SQL de código aberto que é desenvolvido, distribuído e suportado pela Oracle Corporation. A parte SQL significa “*Structured Query Language*” (Linguagem de Consulta Estruturada), que é a linguagem padronizada mais comum usada para acessar bancos de dados [Christudas 2019].

A organização das estruturas do banco de dados ocorre em arquivos físicos otimizados para garantir um desempenho rápido. O modelo lógico, composto por elementos como bancos de dados, tabelas, visualizações, registros e colunas, proporciona um ambiente programático flexível e adaptável. No geral, esse sistema gerencia e permite adicionar, acessar e processar os dados armazenados com alto desempenho, confiabilidade e facilidade de uso [Christudas 2019].

2.6.7. PHP

O PHP é uma linguagem de programação com a natureza de aprimoramento e facilitação da conexão de páginas com o banco de dados cedido pelo servidor, comumente agindo, inclusive, como linguagem de “cola”. Com essa linguagem, é possível criar *scripts* por meio do HTML, havendo compatibilidade com os servidores *web* mais importantes (especialmente o *Apache*), e pode já estar incluso nesses *softwares* de servidores de código aberto, tais como o *Laragon*² e o *Xampp* [Converse and Park 2003].

²*Laragon*: <https://laragon.org/download/index.html>

2.6.8. PWA

Um *Progressive-Web-App* (PWA) é uma aplicação de natureza *web*, contendo sua versão a ser hospedada na *Internet* e que utiliza tecnologias modernas para ser baixada em dispositivos como computadores ou portáteis, de forma a proporcionar uma experiência de usuário semelhante a de uma aplicação nativa de dispositivos móveis — incluindo, por exemplo, recursos como notificações *push*, acesso *offline*, tela inicial personalizada, logotipo, entre outros. Além disso, pode ser atualizada automaticamente, sem a necessidade de uma atualização manual do usuário [Tandel and Jamadar 2018].

Os PWA's são desenvolvidos com tecnologias *web* padrão, como HTML, CSS e *JavaScript*, sendo projetadas para serem responsivas, rápidas e seguras. Esse tipo de aplicativo utiliza um modelo de *cache* para permitir que ele seja acessado *offline* e ofereça uma experiência de usuário contínua, mesmo em conexões de internet instáveis [Oliveira et al. 2022].

Os PWAs são amplamente utilizados em diversas áreas, como comércio eletrônico, mídia, jogos e serviços financeiros. Oferecem uma maneira eficiente e escalável para empresas fornecerem serviços aos usuários em geral [Hume 2017].

3. Metodologia

O presente trabalho consiste de uma pesquisa aplicada, visto que tem como natureza o levantamento de conhecimentos científicos para a aplicação direta na resolução tangível de problemas e desenvolver novas tecnologias, produtos ou serviços. Trata-se do tipo de pesquisa que busca desenvolver e aprimorar protótipos ou produtos com potencial de implementação na sociedade [Gomes and Gomes 2019].

Nesse sentido, foram feitos os seguintes passos para investigação do problema e desenvolvimento do protótipo:

1. **Levantamento interno de requisitos e escolha de tecnologias:** nessa etapa inicial, foram estudados os processos de inscrição e administração do formulário, acompanhando a rotina e obtendo relatos dos profissionais sobre os métodos atuais relacionados no campus. Sobre isso, foram levantadas as necessidades, requisitos e casos de uso, os quais orientaram a delimitação das prioridades e elementos do desenvolvimento do aplicativo;
2. **Capacitação teórica:** estudo sobre as temáticas que embasaram o desenvolvimento do aplicativo, tais como o *framework Laravel*, PWA's, os fundamentos do desenvolvimento de *software* ligado a banco de dados, sistemas de informação e operações CRUD;
3. **Instalação dos softwares e pacotes necessários:** *download* e habilitação do *Laravel*³ e seus recursos que possibilitaram, por exemplo, a implementação do app em forma de PWA; do emulador *Xampp*⁴ para a habilitação dos computadores em servidores locais, relacionando-os com seu banco de dados *My SQL*; e do editor de código-fonte *Visual Studio Code*⁵ para digitação, fácil acesso e organização do

³*Laravel*: <https://getcomposer.org/>

⁴*Xampp*: <https://www.apachefriends.org/>

⁵*Visual Studio Code*: <https://code.visualstudio.com/>

código e dos recursos da biblioteca;

4. **Desenvolvimento do aplicativo e testagem:** para a construção do protótipo, utilizou-se a linguagem de marcação HTML, as linguagens de programação PHP e *JavaScript*⁶, a plataforma *Figma*⁷ para planejamento de sua estilização, que foi realizada pela linguagem CSS, bem como a utilização do web editor *Canva*⁸ para criação de ilustrações visuais. A testagem do protótipo e seus variados ajustes acabaram por ser concomitantes ao seu desenvolvimento, como usual em relação ao modelo de prototipação evolucionário;
5. **Validação profissional, manutenção e testagem:** após o alcance de uma versão robusta do protótipo, foi realizada uma validação profissional com a nutricionista e um servidor do Campus, com a intenção de avaliação do protótipo para, assim, serem planejados ajustes relevantes. Após isso, os testes da manutenção também foram sendo realizados.

4. Desenvolvimento

Conforme investigação do problema através do acompanhamento dos processos digitais e manuais de inscrição para a refeição estudantil e gestão destes registros, bem como consultas com a nutricionista do campus, foram elencados os Requisitos de *Software*. Na Tabela 1, encontram-se todos os Requisitos Funcionais, ou necessidades do programa. Na sequência, a Tabela 2 apresenta os Requisitos Não Funcionais, que são as restrições [Martins 2001].

Também foram definidos os casos de uso da aplicação. A Tabela 3 apresenta os casos totais e generalizados sobre o uso do *app*. Já a Tabela 4 especifica os acessos dos casos de uso por tipo de usuário:

Obteve-se como resultados as seguintes páginas do aplicativo: formulário de registro de presença na refeição escolar, a ser utilizada pelos alunos; informações da equipe desenvolvedora; exposição do cardápio semanal; página de *login*; página de registro de usuário; menu de funcionalidades a ser utilizado pelos administradores autenticados, contendo: relatório de registros totais com filtros, tutorial de cadastro de novo usuário, além de uma página de atualização manual do cardápio semanal.

Essas páginas são conectadas a um banco de dados do sistema local, contendo três tabelas: *discentes* — registro dos estudantes —, *users* — cadastro e autenticação de usuários — e *cardápios* — atualização do cardápio diário.

À medida que o aplicativo se tornou mais robusto, fez-se a adição de recursos e configurações pensadas nas nuances da realidade apresentada, como o caso da transformação de seu *website* em versão para aplicativo PWA, possibilitando-se que na página inicial houvesse um ícone para *download*. Pensando-se nessa funcionalidade de replicação para dispositivos móveis, fez-se necessário que as páginas fossem estilizadas de forma responsiva, de modo que seu *layout* se encaixasse adequadamente em dispositivos móveis de diferentes resoluções. Além disso, visando a ergonomia visual, tanto para a

⁶*JavaScript*: <https://www.java.com/pt-BR/>

⁷*Figma*: <https://www.figma.com/>

⁸*Canva*: <https://www.canva.com/>

Tabela 1. Requisitos Funcionais do aplicativo

Identificador	Nome	Descrição
RF01	Fornecer formulário de almoço	O sistema deve disponibilizar um formulário para que os discentes se inscrevam para o almoço utilizando nome, <i>e-mail</i> , turma, data e justificativa
RF02	Fazer <i>login</i> de administradores	O sistema deve ser capaz de realizar <i>login</i> de administradores
RF03	Página para visualização de registros do almoço	O sistema deve disponibilizar uma página que lista todos os registros diários de confirmação no almoço
RF04	Cadastro/atualização do cardápio	O sistema deve disponibilizar uma página para que os administradores cadastrem ou atualizem o cardápio semanal
RF05	Disponibilizar cardápio	O sistema deve disponibilizar uma página de visualização do cardápio semanal
RF06	Filtrar registros do almoço	O sistema deve ser capaz de filtrar os registros de confirmação para o almoço
RF07	Cadastrar novos usuários	O sistema deve disponibilizar uma página de novo usuário e ser capaz de cadastrar novos administradores

Tabela 2. Requisitos Não Funcionais do aplicativo

Identificador	Nome	Descrição
RNF01	<i>Design</i> do sistema	O sistema deve possuir um <i>design</i> simples e intuitivo, de acordo com a identidade da instituição
RNF02	Acesso de rede	O sistema deve funcionar em qualquer dispositivo com <i>Internet</i>
RNF03	Responsividade	O sistema deve ter conteúdo responsivo, de modo a ser plenamente utilizado em qualquer computador de mesa ou dispositivo móvel

leitura da listagem quanto para a facilitação do processo de impressão do relatório, foi-se decidido a simplificação máxima de sua estilização, com o mínimo de elementos visuais possíveis.

Tabela 3. Casos de uso do aplicativo

Identificador	Nome
CU01	Inscriver no formulário
CU02	Visualizar cardápio
CU03	Cadastrar cardápio
CU04	Visualizar registros
CU05	Filtrar registros
CU06	Atualizar cardápio
CU07	Cadastrar administrador

Tabela 4. Casos de uso por tipo de usuário

Tipo de usuário	Nome
Discente	Inscriver no formulário
Discente e administrador	Visualizar cardápio
Administrador	Cadastrar cardápio
	Visualizar registros
	Filtrar registros
	Atualizar cardápio
	Cadastrar administrador

4.1. Banco de Dados

Nesta seção, é apresentado o Diagrama de Modelo Lógico Relacional (Figura 3), criado com a ferramenta do *MySQL Workbench*⁹ para ilustrar estrutural e graficamente o banco de dados elaborado e utilizado neste trabalho, em especial as tabelas de própria autoria da equipe: cardápios, discentes e *users*.

À esquerda, tem-se a tabela *cardápios*, que possui chave primária *id* e os atributos de entrada de texto de segunda-feira a sexta-feira, destinados a guardar as informações de refeições de cada dia letivo. Há inseridos também os atributos de data de criação e atualização do registro. Esta tabela está relacionada com a de *users* com a seguinte cardinalidade: muitos *users* visualizam e atualizam um cardápio.

À direita, tem-se a tabela *users*, que também possui chave primária *id* e os atributos de identificação pessoal *name*, *usuario*, *e-mail*, *password* e *token* de autenticação. A tabela *users* está relacionada com a tabela *discentes* com a seguintes cardinalidade: muitos *users* pesquisam muitos *discentes*.

No centro, tem-se a tabela *discentes*, que possui chave primária *id* e os atributos de identificação pessoal nome, matrícula, justificativa, *data_permanencia*, e dois atributos de data de criação e atualização do registro. *Discentes* está relacionada com a tabela *cardápios* com a seguinte cardinalidade: muitos *discentes* visualizam um cardápio; assim como se liga à tabela de *users*: muitos discentes são administrados por muitos *users*. Devido à natureza do Diagrama de Modelo Lógico-Relacional, quando se estabelece uma relação

⁹*MySQL Workbench*: <https://www.mysql.com/products/workbench>

de tabelas de tipo *muitos para muitos*, há o surgimento de uma tabela extra contendo as chaves estrangeiras das tabelas envolvidas.

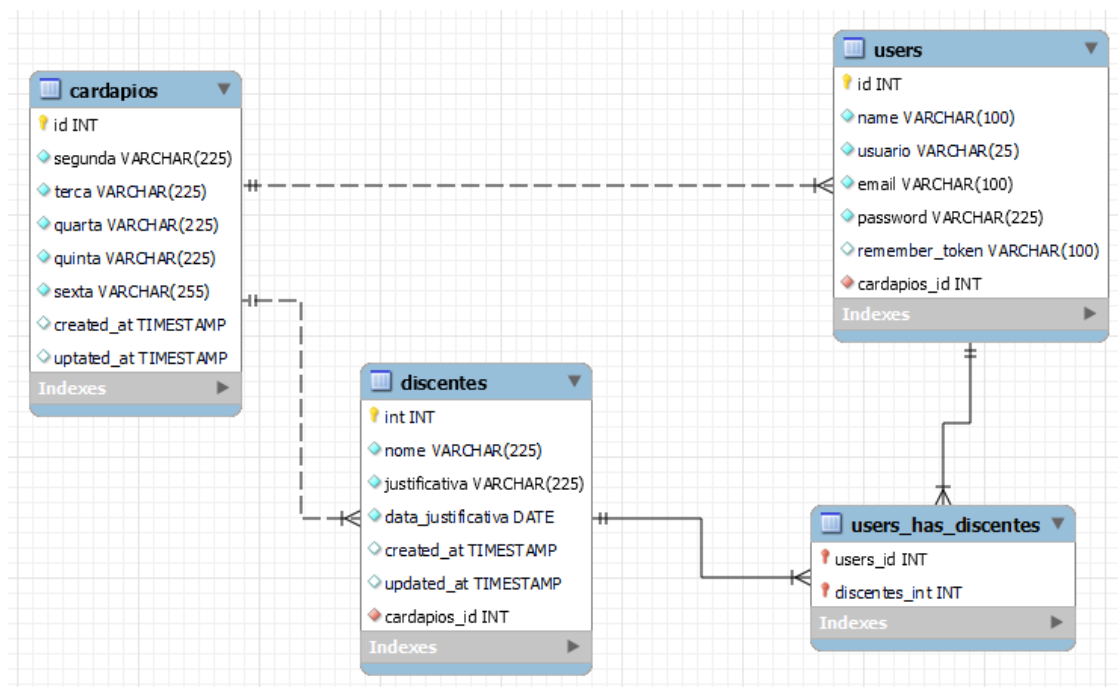


Figura 3. Visualização do DLR do banco de dados do protótipo

4.2. Páginas do Protótipo

Na página inicial (*Home*) há o formulário de indicação de interesse pela refeição estudantil (Figura 4a) e, também, uma barra de navegação que leva à página de formulário, dos desenvolvedores, do cardápio e à de login. O formulário solicita identificação de nome, *e-mail*, matrícula, turma, justificativa de permanência e data, com um botão para confirmar. Este formulário possui codificação que impede mais de uma inscrição no mesmo dia, através da verificação do número de matrícula. No plano de fundo desta página, há uma composição de arte feita com a plataforma *Canva*¹⁰ com vários alimentos flutuantes, de forma a dinamizar a identidade visual do aplicativo de refeitório estudantil. Na Figura 4b, há a visualização de sua versão responsiva para mobile.

¹⁰*Canva*: <https://www.canva.com>

The image shows two versions of a web form titled 'Indicação de alimentação no Campus' from the Instituto Federal da Bahia - Campus Seabra. The desktop version (a) features a horizontal layout with a green header, a navigation bar with 'FORMULÁRIO', 'CARDÁPIO', 'SOBRE', and 'LOGIN', and a form with fields for 'Nome completo', 'E-mail', 'Turma', 'Justificativa de Permanência', and 'Data'. The mobile version (b) shows a vertical layout with the same fields and a 'Confirmar' button. Both versions include a footer with copyright information for IFBA.

Figura 4. Página inicial - Formulário. Versões (a) *desktop* / (b) *mobile*

A partir da página de Formulário (Figura 4) é possível visualizar próximo a sua URL um ícone de *download* — cuja aparência difere-se em cada navegador de acesso — do *website* em forma de *software* para computador, assim como se caracteriza a natureza de um PWA. Esse mesmo tipo de replicação pode acontecer em dispositivos móveis. A Figura 5 demonstra como é este processo:

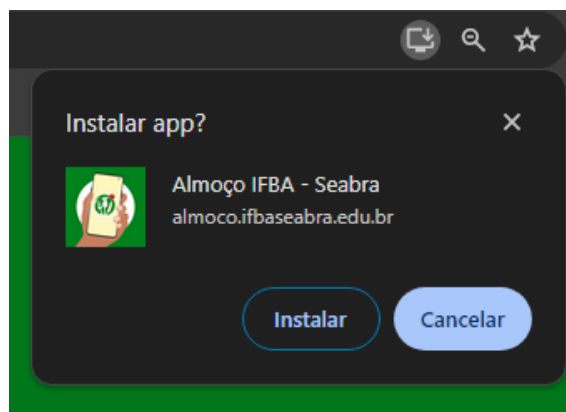


Figura 5. Download do PWA do *website*

Assim que um estudante se cadastra no formulário, recebe imediatamente um *e-mail* automático com uma mensagem de confirmação de inscrição para o almoço na data referente, como demonstrado na Figura 6.

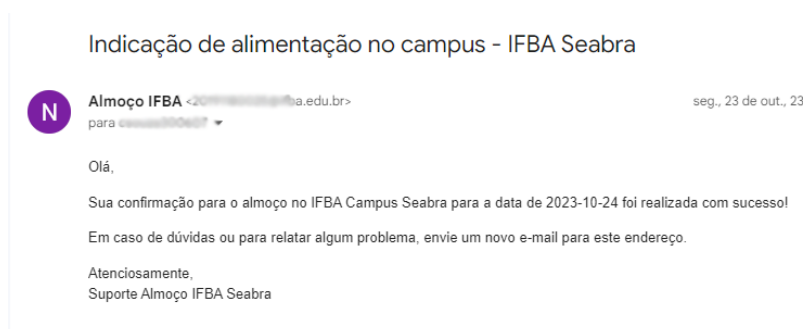


Figura 6. Confirmação por e-mail.

A aba de cardápio semanal na Figura (7a) é acessada pela barra de navegação, e expõe para o público geral qual será a refeição de cada dia da semana, através de um quadro de design e organização simplificados e podendo ser atualizado pela administração a qualquer momento.

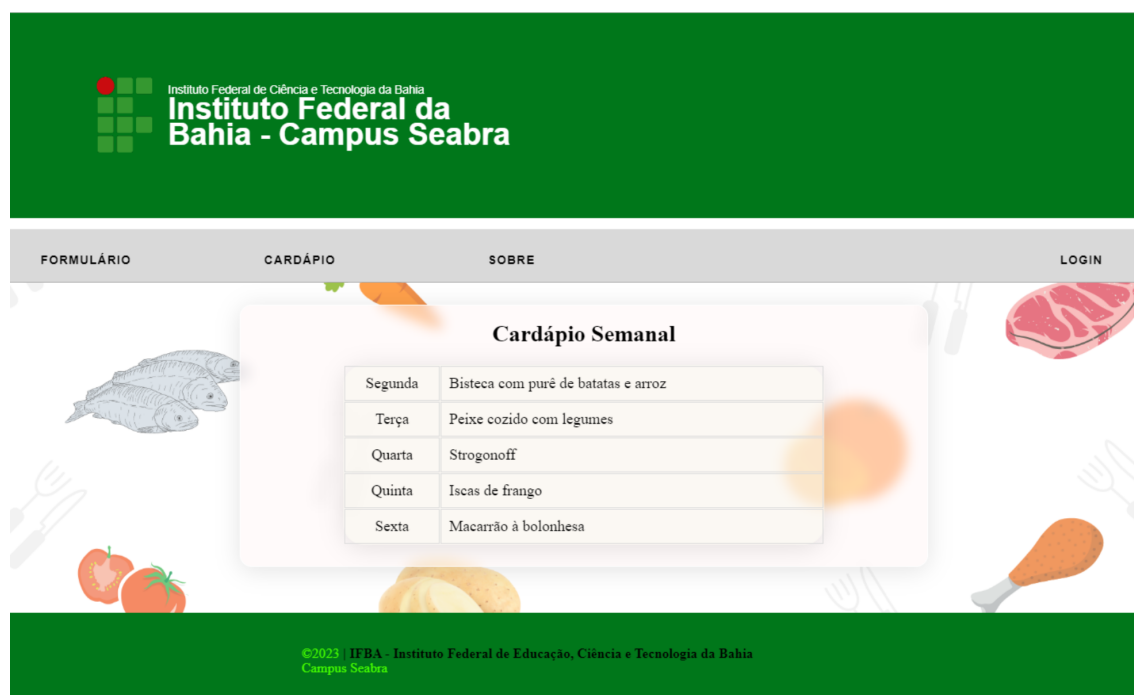


Figura 7. Aba Cardápio.

A página de *login* (Figura 8) tem como funcionalidade o encaminhamento ao menu de administração para pessoas pré-registradas. Nela há o formulário para efetuar o login, e também é permitido a alteração de senha.

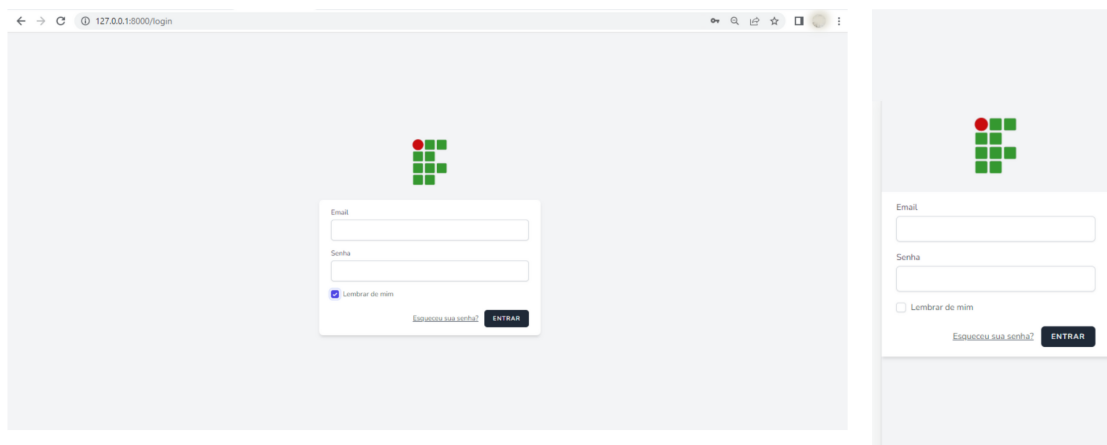


Figura 8. Página de Login. Versão *desktop* / *mobile*

Esta página de *login* também realiza com sucesso a autenticação de *login* dos usuários pré-registrados – os administradores do refeitório. Quando os dados inseridos nos campos não correspondem a um login correto, aparece uma mensagem de erro “*Os dados informados não correspondem aos nossos registros*” em vermelho, exposta na Figura 9:

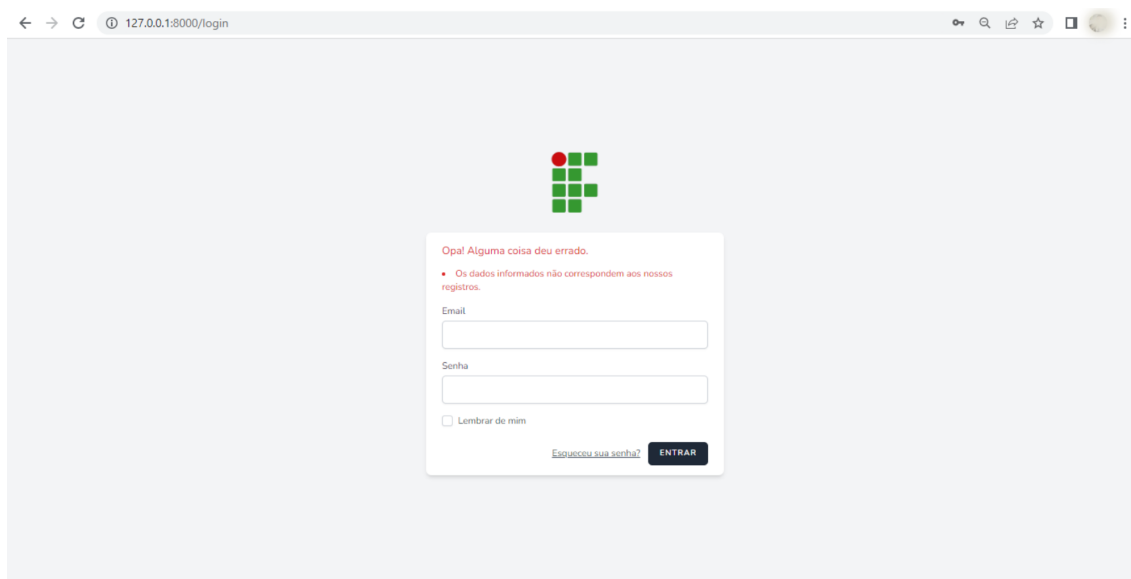


Figura 9. Dados incorretos de *login*

Os responsáveis pelo Refeitório Estudantil podem acessar, através da página de *login*, a página de administrador, por onde poderão gerir os dados. Nesta página, encontra-se dois botões que levam ao relatório diário e o relatório geral das inscrições de alimentação do dia — contendo funcionalidades de filtros de pesquisa por nome, data, turma, justificativa, matrícula e de verificação se o estudante almoçou ou não —, além de

um outro para a página de atualização do cardápio semanal. Abaixo desses campos, em vermelho, há um *hiperlink* para uma página de cadastro de novos administradores.

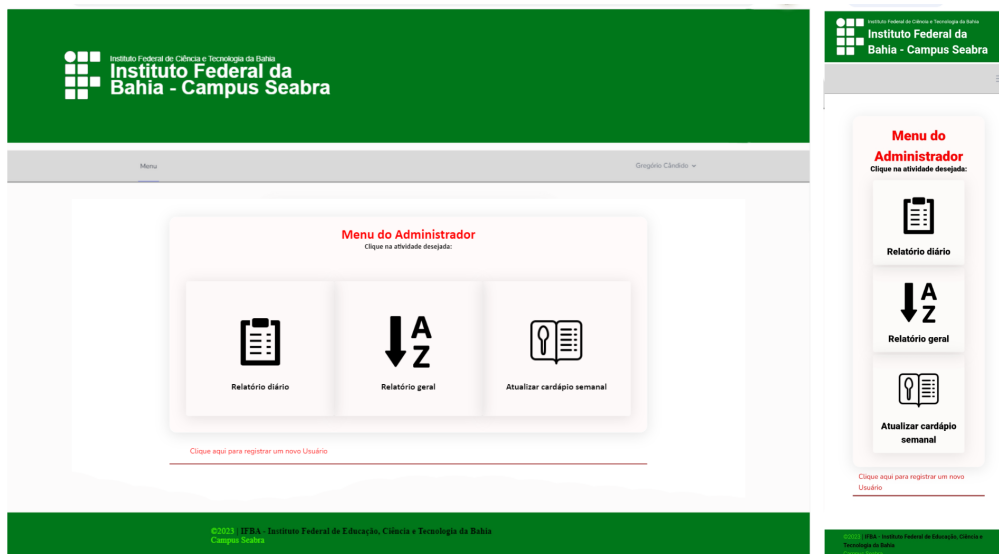


Figura 10. Menu do Adminstrador. Versões (a) *desktop* / (b) *mobile*

Quando acessado, o relatório diário já apresenta os registros de alimentação do dia atual, sempre em ordem alfabética, como representado na Figura 11. Nessa página, através da coluna “Permanência”, conforme a chegada dos estudantes na fila do almoço, o administrador poderá marcar ou não a presença de cada aluno pelos campos de *checkbox* (seleção), podendo salvar a conferência pelo botão de Confirmar no final da página.

Permanência	Nome	E-mail	Turma	Justificativa
<input type="checkbox"/>	Adriano Augusto Mendes	adriano.mendes@ifba.edu.br	3.18.1V	Transporte
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	2.28.1V	Contraturno
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	3.18.1M	Transporte
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	4.28.1M	Contraturno
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	4.18.1M	Transporte
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	2.28.1M	Atendimento física
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	3.18.1V	Contraturno
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	3.28.1V	Contraturno
<input type="checkbox"/>	Alana Rêgina	alana.regina@ifba.edu.br	1.28.1M	Transporte

Figura 11. Relatório diário - Versão *desktop*.

Por fim, dentro do relatório geral (Figura 12) há uma tabela com a listagem dos dados totais de inscrições, sendo possível filtrar registros de forma específica, seja pelo nome, data, matrícula, e-mail, justificativa ou presença no almoço, sendo possível utilizar vários filtros ao mesmo tempo, de forma a fazer pesquisas mais precisas.

RELATÓRIO GERAL
Permanência para o almoço

Buscar discente:

Data: Nome: Matrícula: E-mail:

Turma: Justificativa: Almoçou:

Total de inscrições: 99

Ordem	Nome	Matricula	E-mail	Turma	Justificativa	Data e Almoço hora
120				4.18.1M	Transporte	2023-10-16 sim
121				4.18.1M	atendimento	2023-10-16 não
122				4.18.1M	Transporte	2023-10-17
123				4.18.1M	Atendimento de matemática	2023-10-17

Figura 12. Relatório geral - Versão mobile.

5. Validação Profissional

Foi realizada uma reunião com a nutricionista e o servidor auxiliar do Refeitório Estudantil para uma avaliação profissional do aplicativo. Durante a reunião, o protótipo funcional foi apresentado e ambos profissionais testaram o sistema, utilizando-o para realizar as mesmas tarefas que executavam manualmente no dia a dia. Eles apontaram que, com o aplicativo, essas tarefas foram realizadas de maneira mais direta e automatizada, resolvendo os problemas iniciais enfrentados no processo manual. Além disso, relataram uma experiência de usuário fluida, o que tornou a navegação e utilização das ferramentas mais intuitivas e eficientes.

O retorno foi bastante positivo, com ambos mostrando entusiasmo em explorar melhor os recursos oferecidos pela aplicação. Isso gerou uma discussão produtiva sobre possibilidades de melhorias, tanto para implementação imediata quanto para trabalhos futuros.

Entre os pontos mais destacados na avaliação profissional, está a natureza PWA (Progressive Web App) do aplicativo, possibilitando seu uso em dispositivos móveis pelos alunos e administradores, com total responsividade. Essa característica garante praticidade aos responsáveis pelo refeitório, que podem gerenciar os registros diretamente pelo celular, precisando apenas de uma conexão com a internet.

Outro aspecto prestigiado foi a funcionalidade de filtros múltiplos no relatório diário. A possibilidade de filtrar registros de forma mais direta e completa, similar ao que fariam no *Google Forms*, foi considerada uma grande vantagem, contribuindo para maior

agilidade e precisão no gerenciamento diário.

6. Relatórios de acessibilidade e PWA

Pensando-se no fato de que há a presença de estudantes com deficiências visuais na instituição, foi realizado um teste de acessibilidade para se obter referência de melhorias para a plena utilização do aplicativo por estes discentes.

As páginas analisadas foram as de principal utilização pelos discentes: “Formulário Almoço”, “Desenvolvedores” e “Cardápio”. O teste foi realizado pela ferramenta *online Lighthouse*¹¹ do *Google DevTools*, a qual processa páginas *web* de forma a apresentar diversos tipos de diagnóstico, como de acessibilidade, de desempenho, de replicação em Progressive-Web-App, entre outros.

Em quesito de acessibilidade, as pontuações das três páginas supracitadas ficaram entre 70 e 74 de 100, tanto em *desktop* quanto em dispositivos móveis. Dessa forma, apesar do resultado se apresentar acima da média, compreende-se que há uma considerável necessidade de melhoria dos recursos de acessibilidade oferecidos pelo aplicativo.

Há diversas carências de acessibilidade expostas pelo relatório destas páginas: a falta de atributos *alt*, a falta de etiquetas associadas nos elementos no formulário de interesse à refeição, a necessidade de maior contraste entre elementos visuais das páginas, como ilustrado em um dos relatórios exposto na Figura 13.



Figura 13. Relatório de acessibilidade do *Lighthouse*.

¹¹*Lighthouse*: <https://developer.chrome.com/docs/lighthouse/overview/>

A partir dos resultados dos diagnósticos, pôde-se obter consciência da situação de desempenho do protótipo em quesito de acessibilidade visual, compreendendo-se que são necessárias novas implementações de programação e melhorias de aspectos visuais, de maneira a aprimorar a experiência do sistema para os estudantes que poderiam utilizá-lo.

Em quesito de replicação do aplicativo em forma de PWA, através do relatório do *Lighthouse*, observou-se que o sistema integrou essa natureza digital com sucesso, podendo ser baixado e utilizado em dispositivos *desktop* e móveis. A Figura 14 lista os detalhes deste diagnóstico.



Figura 14. Relatório de replicação em PWA do *Lighthouse*.

7. Conclusões

O propósito central deste projeto se voltou na resolução da problemática de que, no contexto de gestão das solicitações do almoço estudantil, a administração do refeitório do IFBA Campus Seabra possuía dificuldade com o aspecto manual e a falta de eficiência nos processos de gestão dos registros, urgindo na necessidade de desenvolvimento de uma ferramenta que auxiliasse nessas atividades. Assim, obteve-se como resultado um protótipo funcional de aplicativo que oferece aos alunos a praticidade de confirmar sua presença no almoço de forma *online*, bem como uma visualização conveniente do cardápio semanal. Além disso, oferece funcionalidades rápidas de monitoramento das inscrições – com a utilização da contagem de alunos e dos filtros múltiplos – e de organização do refeitório – com o cadastro do cardápio semanal e o registro de administradores.

A construção do sistema foi norteada pela investigação inicial realizada pela equipe sobre os processos já estabelecidos na escola em relação à inscrição e gerenciamento através do *Google Forms*, bem como o acompanhamento dessas atividades e a consulta com os profissionais encarregados. Intercalou-se o desenvolvimento do aplicativo — utilizando a biblioteca *Laravel*, as linguagens PHP, HTML, CSS e *JavaScript*, as *plataformas* Figma e Canva — e a realização de diversos testes simultâneos. Nessa trajetória, diversas dificuldades foram enfrentadas, necessitando maior aprofundamento técnico, pesquisa extracurricular e aproveitamento de recursos digitais para a resolução

dos problemas. Entretanto, mesmo diante a tamanha dificuldade, a equipe alcançou a finalização de um trabalho concreto, funcional e pragmático.

Por fim, conseguiu-se desenvolver um protótipo de alta fidelidade funcional, alcançando-se os objetivos iniciais do trabalho, desde o entendimento do problema à validação profissional. Com isso, encontra-se cada vez mais próxima a possibilidade de implementação da ferramenta em campo, visando estabelecer a independência da instituição em relação a serviços externos, além da eficiência frente a processos rotineiros de gestão do refeitório. Dessa forma, observa-se o quão necessário e importante são as ações de pesquisa aplicada para a solução de problemas e otimização de processos no cenário estudantil.

Referências

- [Aragão et al. 2020] Aragão, A., Machado, L., Moreno, N., Viana, D., Silva, F., Sousa, T., Rivero, L., Teles, A., da Conceição, A. F., and Costa, I. (2020). Evoluindo uma aplicação para cidades inteligentes através de avaliação de métricas de qualidade e usabilidade.
- [Christudas 2019] Christudas, B. (2019). *MySQL*.
- [Converse and Park 2003] Converse, T. and Park, J. (2003). *PHP: A Bíblia*. Alta Books.
- [Elmasri et al. 2005] Elmasri, R., Navathe, S. B., Pinheiro, M. G., et al. (2005). *Sistemas de Banco de Dados*. Pearson Prentice Hall, 5 edition.
- [Figma 2023] Figma (2023). Figma. Disponível em: <https://www.figma.com/>.
- [Flanagan 2004] Flanagan, D. (2004). *JavaScript: O Guia Definitivo*. Bookman Editora.
- [Floyd 1984] Floyd, C. (1984). A systematic look at prototyping. *ACM Computing Surveys*, 16(2):149–206.
- [Gabardo 2017] Gabardo, A. C. (2017). *Laravel para Ninjas*. Novatec.
- [Gomes and Gomes 2019] Gomes, A. S. and Gomes, C. R. A. (2019). Classificação dos tipos de pesquisa em informática na educação. *Revista Brasileira de Informática na Educação*, 27(2):45–56.
- [Heuser 2009] Heuser, C. A. (2009). *Projeto de Banco de Dados*. Pearson Prentice Hall.
- [Hume 2017] Hume, D. (2017). *Progressive Web Apps*. Simon & Schuster.
- [Kalinowski et al. 2023] Kalinowski, M., Escovedo, T., Villamizar, H., and Lopes, H. (2023). *Engenharia de Software para Ciência de Dados: Um Guia de Boas Práticas com Ênfase na Construção de Sistemas de Machine Learning em Python*. Novatec.
- [Laravel 2023] Laravel (2023). Documentação. Disponível em: <https://laravel.com/docs/11.x>.
- [MacDonald 2011] MacDonald, M. (2011). *HTML5: The Missing Manual*. O'Reilly Media.
- [Marcondes 1972] Marcondes, R. S. (1972). *Educação em Saúde na Escola*. Saraiva.
- [Martins 2001] Martins, L. E. G. (2001). *Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade*. Atlas.

- [Nonnenmacher 2012] Nonnenmacher, R. F. (2012). Estudo do comportamento do consumidor de aplicativos móveis. *Revista Brasileira de Marketing*, 11(4):90–110.
- [Oliveira et al. 2022] Oliveira, F. C. S., Freitas, H. R., Ramos, J. L. C., and Diniz, M. R. (2022). Um web app com elementos de u-learning e gamificação voltado ao contexto da pedagogia da alternância. *Revista Brasileira de Educação a Distância*, 16(3):301–317.
- [Paulista 2005] Paulista, U. E. (2005). Prototipação de software. Disponível em: http://disciplinas.lia.ufc.br/es062/arquivos/Captulo_8-Prototipacao.pdf.
- [Pedraza et al. 2018] Pedraza, D. F., Melo, N. L. S. d., Silva, F. A., and Araujo, E. M. N. (2018). Avaliação do programa nacional de alimentação escolar: Revisão da literatura. *Revista Brasileira de Nutrição*, 16(1):22–39.
- [Pressman and Maxim 2021] Pressman, R. S. and Maxim, B. R. (2021). *Engenharia de Software-9*. McGraw-Hill Education.
- [Putra et al. 2021] Putra, Z. F. F., Ajie, H., and Safitri, I. A. (2021). Designing a user interface and user experience from piring makanku application by using figma application for teens. *Journal of Digital Design*, 12(4):78–85.
- [Ramakrishnan and Gehrke 2003] Ramakrishnan, R. and Gehrke, J. (2003). *Database Management Systems*. McGraw-Hill Education, 3 edition.
- [Silva 2019] Silva, M. S. (2019). *HTML5: A Linguagem de Marcação que Revolucionou a Web*. O'Reilly Media.
- [Silva 2000] Silva, R. P. e. (2000). Suporte ao desenvolvimento e uso de frameworks e componentes. *Revista Brasileira de Computação*, 22(1):45–60.
- [Sommerville and Sawyer 1997] Sommerville, I. and Sawyer, P. (1997). Viewpoints: Principles, problems and a practical approach to requirements engineering. *ACM Computing Surveys*, 29(1):53–72.
- [Tandel and Jamadar 2018] Tandel, S. and Jamadar, A. (2018). Impact of progressive web apps on web app development. *International Journal of Web Engineering and Technology*, 13(2):125–137.
- [Teodosio et al. 2021] Teodosio, S. d. S., Parente, D. P., and Alves, F. R. V. (2021). As tecnologias da informação e comunicação como possibilidade para a gestão da informação no Âmbito da política de assistência estudantil. *Revista Brasileira de Tecnologia da Informação*, 9(3):101–113.
- [Tidd and Bessant 2015] Tidd, J. and Bessant, J. (2015). *Gestão da Inovação-5*. Pearson.
- [Wazlawick 2019] Wazlawick, R. (2019). *Engenharia de Software: Conceitos e Práticas*. Novatec.
- [Wessel et al. 2022] Wessel, M., Thies, F., and Benlian, A. (2022). The role of prototype fidelity in technology crowdfunding. *Journal of Business Research*, 45(1):92–104.