



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

Minha Bandeja: Uma plataforma web para gerenciamento de filas do Restaurante
Universitário da UNIRIO

Caio Rodrigues Dias de Jesus
Walter Saldanha Pereira Filho

Orientador
Paulo Sérgio Medeiros dos Santos

RIO DE JANEIRO, RJ – BRASIL

Minha Bandeja: Uma plataforma web para gerenciamento de filas do Restaurante
Universitário da UNIRIO

Caio Rodrigues Dias de Jesus

Walter Saldanha Pereira Filho

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovado por:

Prof. Paulo Sérgio Medeiros dos Santos, DSc. (UNIRIO)

Prof. Reinaldo Viana Alvares, Dsc. (UNIRIO)

Prof. Victor Vidigal Ribeiro, Dsc. (Granbery)

RIO DE JANEIRO, RJ – BRASIL

FEVEREIRO DE 2023

Agradecimentos

Para começar, gostaríamos de agradecer ao nosso professor Paulo Sérgio, nosso orientador, por toda a ajuda durante o desenvolvimento deste trabalho no sentido de alinhar as ideias dispersas e nos guiar a alinhá-las de forma a viabilizar a confecção deste trabalho.

Em seguida, não poderíamos deixar de mencionar todos os professores da UNIRIO que com certeza influenciaram, mesmo que indiretamente, na produção deste trabalho, com todo o conhecimento oferecido durante nosso período de graduação na instituição. Este trabalho reúne conteúdo presente em diversas disciplinas (incluindo optativas) e é maravilhoso ver o que somos capazes de desenvolver após os anos dedicados na instituição.

Por último, à instituição como um todo, o BSI, por cultivar um ambiente de ensino e aprendizado, promovendo troca constante entre todos os discentes, de onde ideias como a desenvolvida neste trabalho puderam encontrar espaço para serem desenvolvidas.

Agora, ao separar as narrativas, eu, Caio, quero primeiramente agradecer a Deus, que me dá sustentação e paz, acima de todas as coisas. Ainda, gostaria de deixar o meu sincero agradecimento a todos que nos últimos anos dedicaram algum tempo e esforço a mim, seja ajudando com alguma dificuldade durante o curso ou fora dele. Como não é possível listar todos, gostaria de mencionar alguns cujo impacto se mostrou imensurável: meus pais, Bartolomeu e Rose, por sempre me motivarem a perseguir a educação e me ensinarem a valorizar a mesma; aos meus irmãos Igor e Felipe, por serem grandes inspirações de vida no que se diz respeito ao esforço, consistência e dedicação ao perseguir nossos objetivos; à minha esposa Greice, que esteve sempre ao meu lado durante os diversos períodos de adversidade, sempre mantendo meus pés no chão e reduzindo minha carga de vida para que eu pudesse continuar o foco nos meus estudos; ao meu filho Bernardo, por ser a minha força para continuar seguindo em frente, sonhando e trabalhando pelo futuro dele; ao meu amigo João Paulo, por ter sido meu grande parceiro e professor nas matérias que mais tive dificuldade; e finalmente, mas não menos importante, ao Walter, amigo com quem compartilho este trabalho, por

ter transformado dias difíceis em dias divertidos, cultivando lembranças que levarei para a vida toda do meu período de discente.

E, antes de encerrar, eu, Walter, preciso agradecer imensamente a Deus, que me sustentou por toda essa jornada, por períodos de alto e baixo, por trazer calma diante de toda a preocupação; à minha família, principalmente meus pais, Walter e Eloísa, que sempre, sempre valorizaram a educação na minha vida e na dos meus irmãos, que se abstiveram de tantas coisas para favorecer a perseguição na busca pela educação, que investiram, e não foi pouco, mas com a certeza do resultado; aos meus irmãos, Pedro Henrique e Sophia Mariah, que também abriram mão de muita coisa, muitas noites me esperando terminar os estudos para dormir, que sempre me deram a motivação e o suporte necessário para focar nas coisas importantes; à minha parceira, Victória Regina, sem essa mulher na minha vida eu não seria tão feliz, ela me motiva, me auxilia, me direciona, ela sempre esteve ao meu lado (por vezes caindo de sono), durante todos esses anos na conclusão da minha graduação, me ajudando a ter certeza das minhas escolhas; aos meus amigos, não vou citar todos, pois poderia cometer alguma injustiça; a toda a equipe que fez ou faz parte do Sistema Integrado de Gestão Acadêmica da UFRJ, minha primeira referência profissional, que hoje me deixa colher muitos frutos; a todas as caronas, onde fiz grandes amizades; e deixando por último, pois eu o agradeço sempre que posso, meu amigo Caio, onde pude compartilhar durante toda a minha graduação, além de uma amizade verdadeira, difícil de encontrar, momentos únicos e que serão inesquecíveis em minha vida, mesmo com tantas dificuldades e desafios, nossa cooperação culminou na conclusão deste trabalho.

RESUMO

Os Restaurantes Universitários são peças importantes da comunidade acadêmica, pois fornecem alimentação de qualidade a um custo acessível, mantendo normas nutricionais e de higiene. Para serem bem aproveitados, as dificuldades de acesso ao mesmo devem ser reduzidas ao máximo, sendo as filas um dos maiores problemas presentes no dia a dia dos usuários no caso da UNIRIO.

Este trabalho idealiza e desenvolve uma primeira versão da plataforma Minha Bandeja para atenuar este problema e trazer aumento na qualidade de vida dos usuários do "bandejão", diminuindo o tempo gasto para fazer uma refeição.

O Minha Bandeja é uma aplicação WEB para gerenciamento de filas, contando com um módulo de cadastro e autenticação, além do módulo principal do controle de filas, utilizando-se tecnologias de *front-end* e *back-end*, explorando também banco de dados relacionais e não-relacionais.

O presente trabalho é uma versão preliminar do sistema, e dentro do escopo deste é possível realizar a administração básica de filas de espera, assim como o ingresso nestas por parte dos usuários registrados.

A partir desta versão preliminar, a plataforma pode ser continuada como um projeto dentro da UNIRIO como forma de digitalizar ainda mais o controle do acesso e uso do Restaurante. O projeto pode ser aproveitado também para promover um ambiente de aprendizagem e conhecimento aos alunos que fizerem parte do projeto.

Palavras-chave: Desenvolvimento de Software, Gestão de Fila, Sistemas WEB, Restaurante Universitário, REST API.

ABSTRACT

University Restaurants are important parts of the academic community: they provide quality food at an affordable cost, maintaining nutritional and hygiene standards. To be well used, difficulties in accessing it must be reduced to the maximum, with queues being one of the biggest problems present in the daily lives of users in the case of UNIRIO.

This work idealizes and develops a first version of the *Minha Bandeja* platform to alleviate this problem and bring an increase in the quality of life of "*bandejão*" users, reducing the time spent to have a meal.

Minha Bandeja is a WEB application for managing queues, with a registration and authentication module, in addition to the main queue control module, using front-end and back-end technologies, also exploring relational and non-relational databases.

The present work is a preliminary version of the system, and within its scope it is possible to carry out the basic administration of queues, as well as the entry into these by registered users.

From this preliminary version, the platform can be continued as a project within UNIRIO as a way to further digitize access control and use of the Restaurant. The project can also be used to promote a learning and knowledge environment for students who were part of the project.

Keywords: Software Development, Queue Management, WEB Systems, University Restaurant, REST API.

SIGLAS

UNIRIO - Universidade Federal do Rio de Janeiro

CCET - Centro de Ciências Exatas e Tecnologia

CLA - Centro de Letras e Artes

UFRJ - Universidade Federal do Rio de Janeiro

CT - Centro de Tecnologia

RU - Restaurante Universitário

CPF - Cadastro de Pessoas Físicas

SIAPE - Sistema Integrado de Administração de Pessoal (normalmente utiliza-se Matrícula SIAPE)

PRAE - Pró-Reitoria de Assuntos Estudantis

API - *Application Programming Interface*

AWS - *Amazon Web Services*

BSD - *Berkeley Software Distribution*

GCP - *Google Cloud Platform*

JS - *JavaScript*

JSON - *JavaScript Object Notation*

JWT - *JSON Web Tokens*

HTML - *Hypertext Transfer Protocol*

TTL - *Time to Live*

WWW - *World Wide Web*

MVP - *Minimum viable product*

Índice

| | |
|---|-----------|
| 1. Introdução | 13 |
| 1.1. Motivação | 13 |
| 1.2. Problema | 14 |
| 1.3. Proposta de solução | 15 |
| 1.4. Objetivos | 15 |
| 1.5. Organização do Texto | 15 |
| 2. Fundamentação | 17 |
| 2.1. Restaurante Universitário | 17 |
| 2.2. Sistema de Filas | 18 |
| 2.3. Trabalhos Relacionados | 19 |
| 3. Desenvolvimento da Plataforma | 20 |
| 3.1. Atores do sistema | 20 |
| 3.1.1. Usuário do RU | 20 |
| 3.1.2. Usuário Administrador | 20 |
| 3.1.3. Sistema | 20 |
| 3.2. Elicitação de requisitos | 21 |
| 3.2.1. Requisitos Funcionais | 21 |
| 3.2.2. Requisitos Não-funcionais | 22 |
| 3.2.3. Regras de Negócio | 22 |
| 3.3. Seleção das tecnologias | 24 |
| 3.3.1. JavaScript | 24 |
| 3.3.2. Node.js | 24 |
| 3.3.3. TypeScript | 25 |
| 3.3.4. Postgres | 26 |
| 3.3.5. Redis | 26 |
| 3.3.6. NestJS | 27 |
| 3.3.7. GIT | 28 |
| 3.3.8. Sonar Cloud | 29 |
| 3.3.9. VueJS | 29 |
| 3.4. Atividades de qualidade | 31 |
| 3.5. Arquitetura Proposta | 31 |
| 3.5.1. Aplicativo Web | 32 |
| 3.5.2. Node Rest API | 32 |
| 3.5.3. Banco de Dados relacional | 33 |
| 3.5.4. Banco de Dados não relacional | 33 |
| 3.5.5. Worker Orquestrador de Filas | 34 |
| 4. Apresentação da plataforma | 35 |
| 4.1. Implementação das Filas | 35 |

| | |
|---|-----------|
| 4.2. Funcionalidades para ambos os perfis | 36 |
| 4.2.1. Registro de novo usuário | 36 |
| 4.2.2. Acesso à plataforma | 37 |
| 4.2.3. Visualizar as filas | 38 |
| 4.2.4. Entrar na fila | 38 |
| 4.2.5. Sair da fila | 39 |
| 4.2.6. Sair da plataforma | 39 |
| 4.3. Funcionalidades exclusivas de usuário administrador | 40 |
| 4.3.1. Visualizar as filas | 40 |
| 4.3.2. Visualizar usuários na fila | 40 |
| 4.3.3. Alterar configurações de filas | 41 |
| 4.3.4. Ver novas configurações de fila agendadas | 42 |
| 4.3.5. Excluir configuração de fila agendada | 43 |
| 5. Uso hipotético do sistema | 44 |
| 5.1. Configurações iniciais | 44 |
| 5.2. Utilização no dia a dia pelos alunos e servidores | 44 |
| 5.3. Utilização no dia a dia pelos administradores do sistema | 46 |
| 6. Conclusão | 47 |
| 6.1. Contribuições | 47 |
| 6.2. Limitações | 47 |
| 6.3. Trabalhos futuros | 49 |
| Referências Bibliográficas | 50 |

Índice de Figuras

| | |
|---|----|
| Figura 1. Fila do Restaurante Universitário | 14 |
| Figura 2. RU da UNIRIO | 17 |
| Figura 3. Logotipo JavaScript | 24 |
| Figura 4. Logotipo Node.js | 24 |
| Figura 5. Logotipo TypeScript | 25 |
| Figura 6. Logotipo Postgres | 26 |
| Figura 7. Logotipo Redis | 26 |
| Figura 8. Logotipo NestJS | 27 |
| Figura 9. Logotipo Git | 28 |
| Figura 10. Logotipo SonarCloud | 29 |
| Figura 11. Logotipo VueJS | 29 |
| Figura 12. Formulário de cadastro | 36 |
| Figura 13. Formulário de acesso | 37 |
| Figura 14. Visualização das filas do dia | 38 |
| Figura 15. Visualização de filas com foco na fila qual o usuário solicitou entrada | 38 |
| Figura 16. Menu com botão de sair da plataforma | 39 |
| Figura 17. Visualização de filas do administrador | 40 |
| Figura 18. Visualização de uma fila por um administrador | 40 |
| Figura 19. Formulário de configuração de filas | 42 |
| Figura 20. Visualização de configurações agendadas de configuração de filas | 42 |

Índice de ilustrações

| | |
|---|----|
| Ilustração 1. Representação esquemática de um sistema com fila | 18 |
| Ilustração 2. Diagrama da Arquitetura Proposta | 31 |
| Ilustração 3. Diagrama de Entidade-Relacionamento | 33 |
| Ilustração 4. Diagrama de Representação | 34 |
| Ilustração 5. Representação de filas do sistema | 35 |

Índice de tabelas

| | |
|--|----|
| Tabela 1. Requisitos Funcionais | 21 |
| Tabela 2. Requisitos não-funcionais | 22 |
| Tabela 3. Regras de negócio | 22 |

1. Introdução

Quando um estudante é aprovado no vestibular e realiza o objetivo de ingressar em uma universidade pública, por vezes este sonho vem repleto de diversos obstáculos atrelados. Muitos destes desafios são de cunho intelectual, como as disciplinas desafiadoras e as inúmeras provas e listas de exercícios oferecidas pelos professores. Entretanto, alguns passam despercebidos, como a alimentação e transporte.

O restaurante universitário, às vezes, é visto apenas como uma alternativa à alimentação. Contudo, para inúmeros alunos que ingressam na instituição, principalmente os que vêm de outros estados, é um pré-requisito para a permanência do estudante. Ainda mais no Centro de Ciências Exatas e Tecnologia da Universidade Federal do Estado do Rio de Janeiro (CCET - UNIRIO), onde as opções de alimentação próximas são escassas e muito dispendiosas.

Com as inúmeras possibilidades abertas a serem desenvolvidas com o conhecimento de Engenharia de Software obtido durante a graduação, optou-se por pensar em um problema que pudesse gerar um tipo de retorno à instituição.

1.1. Motivação

As ideias apresentadas neste trabalho são baseadas em uma experiência vivida por seus autores na Universidade Federal do Rio de Janeiro (UFRJ), que contava com um sistema de agendamento de refeições¹ no restaurante universitário do Centro de Tecnologia (CT).

Refletindo sobre a experiência atualmente vivida pelos alunos e servidores da UNIRIO no acesso ao restaurante universitário da instituição, pode ser viável a implementação de um sistema de informação que pudesse auxiliar todos de forma a organizar de maneira prática e acessível o acesso a este, economizando o tempo de todos os envolvidos. Atualmente, filas enormes são vistas fisicamente, tornando a experiência como um todo pouco satisfatória.

1.2. Problema

O Restaurante Universitário da UNIRIO favorece inúmeros alunos da instituição com alimentação adequada. Entretanto, nem sempre a forma como isso é organizado possibilita uma interação ótima entre os alunos e suas aulas. Muitas vezes, uma aula termina às 12h e a outra começa 13h, outras, no turno da noite, nem sequer possuem intervalo. Dessa forma, o aluno tem apenas 1h de almoço, e mais da metade (quando não todo esse tempo) é investido na fila do restaurante.

Abaixo, na Figura 1, pode ser visualizado as grandes filas formadas nos horários de refeição.

Figura 1. Fila do Restaurante Universitário



Fonte: FOURSQUARE, 2019

Por meio de um sistema de apoio como o Minha Bandeja, as necessidades dos alunos serão endereçadas, ajudando-os na organização do tempo de alimentação e no tempo de estudo.

Quando uma aula termina às 12h e a outra começa às 13h, por exemplo, os alunos podem entrar previamente na fila, sem ter a preocupação de como o tamanho da fila pode impactar seu horário de chegada na próxima aula.

1.3. Proposta de solução

De forma a utilizar todo o conhecimento de sistemas de informação adquirido durante a formação acadêmica na UNIRIO, a proposta de solução construída pelos autores desta monografia é um sistema WEB cuja responsabilidade é organizar os horários das refeições dos alunos da instituição no restaurante universitário, de forma a reduzir o caos criado por filas muito grandes e pouco otimizadas. Além disso, buscando reduzir a “ansiedade” gerada pelos alunos na utilização do restaurante em relação aos horários das atividades acadêmicas, pois muitas vezes as janelas entre uma aula ou atividade e a próxima são curtas, e o pouco tempo para a refeição ainda é encurtado pelas filas mencionadas.

1.4. Objetivos

Compreendendo as restrições organizacionais, legais e de segurança da informação envolvidos na implantação de um novo sistema online em uma instituição pública, o objetivo deste trabalho é o de construir um sistema WEB cujas funcionalidades incluem: agendamento prévio de horários no restaurante universitário por alunos, professores e funcionários da universidade, e que seja capaz de ser disponibilizado utilizando quaisquer tipos de infraestrutura:

- **Tradicional**, ou seja, gerida pelos próprios funcionários da UNIRIO e de uso exclusivo da instituição, também chamado de *on-premise*;
- **Em nuvem**, ou seja, gerida por uma das ferramentas conhecidas pelo mercado, como Amazon Web Services (AWS), Azure, Google Cloud (GCP), dentre outras.

1.5. Organização do Texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Neste capítulo, apresentamos a fundamentação da nossa proposta e explanação sobre as técnicas e abordagens utilizadas;
- Capítulo III: Neste capítulo, abordamos com detalhes a construção da plataforma WEB que foi proposta como solução para o problema abordado neste trabalho;

- Capítulo IV: Neste capítulo, abordamos a apresentação da plataforma, descrevendo detalhadamente as funcionalidades do sistema, sua agregação de valor e o principal objetivo;
- Capítulo V: Neste capítulo, abordamos o uso hipotético do sistema, detalhando passo a passo a visão dos atores.
- Capítulo VI: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

2. Fundamentação

Buscando compreender os conceitos por trás do problema apresentado, nota-se a importância e a apropriação de alguns conhecimentos relevantes para a formalização deste trabalho.

2.1. Restaurante Universitário

Os Restaurantes Universitários (RUs), ou "bandejão", como são popularmente conhecidos entre os alunos, são restaurantes que existem dentro das universidades públicas e tem como finalidade a oferta de refeições ao corpo discente, docente e técnico administrativo da Universidade, por meio de uma alimentação de qualidade e em quantidade adequada.

Na Figura 2, a seguir, podemos observar a disposição das bancadas para servir as refeições.

Figura 2. RU da UNIRIO



Fonte: UNIRIO, 2019

O Restaurante Universitário da UNIRIO foi inaugurado no dia 26 de agosto de 2016 e aberto ao público interno 4 dias depois², sendo localizado em um prédio dedicado à instalação de um refeitório, ao lado do Centro de Letras e Artes no campus da Avenida Pasteur no número 458. O espaço dispõe de 340 lugares, distribuídos em 2 pavimentos (térreo e mezanino), contando com duas bancadas para o fornecimento dos alimentos. Atualmente, a média de refeições servidas é de aproximadamente 1.000 refeições ao dia³, distribuídas entre almoço e jantar, informado no portal do próprio Restaurante Escola⁴. Ademais, o cardápio é divulgado no portal a cada início de semana e no perfil do Instagram.

2.2. Sistema de Filas

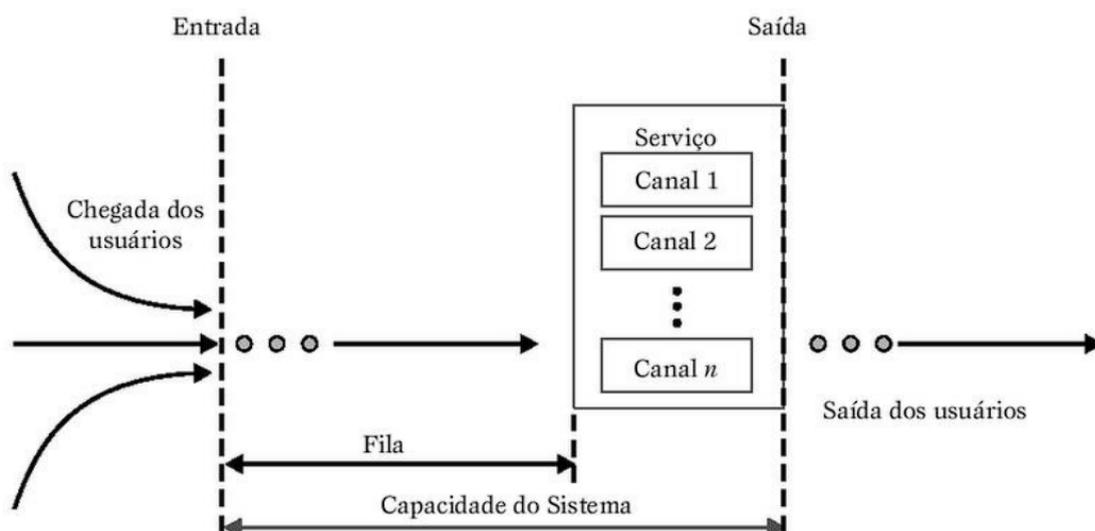
Os sistemas de filas geralmente são associados a melhoria de desempenho de um sistema (computacional ou não), buscando otimizar a utilização de recursos de serviço e agilizar o atendimento. Falando especificamente de software, o pioneiro nesse estudo foi A. K. Erlang, que, no começo do século XX como engenheiro de uma companhia dinamarquesa de telefones, estudou o problema da congestão de linhas. Os modelos de filas nos permitem encontrar um equilíbrio entre a falta ou excesso de capacidade, mitigando os efeitos não desejados no custo e tempo de espera ^{5,6}.

Utilizando-se dessa teoria, é possível identificar as causas-raiz que originam as filas. A principal meta é mitigar ou resolver os efeitos não desejados dos problemas de espera, no caso do Restaurante Universitário e dos alunos poderá ser a redução de tempo. Com a aplicação dessa metodologia, são fornecidas ferramentas para identificar e avaliar variáveis como tempo de espera e o tamanho da fila, por exemplo.

Entre as principais características de um sistema de filas, pode-se citar o processo de chegada de usuários, processo de atendimento, canais ou postos de serviço/atendimento, capacidade do sistema e disciplina de atendimento, tal como foram descritos por Tomé *et al.* (2019)⁷.

A Ilustração 1 apresenta de forma detalhada um esquema de sistemas utilizando filas.

Ilustração 1. Representação esquemática de um sistema com fila



Fonte: FOGLIATTI, MATTOS⁸ apud TOMÉ (2007)

Os ambientes consistem em pontos de entrada que os usuários alcançam. Em seguida, há um espaço destinado a formar uma única fila e vários postos de atendimento

paralelos. Numa altura em que todos os canais de atendimento estão ocupados, os clientes aguardam na fila. Essa espera ocorre devido ao fluxo do sistema quando a demanda é maior que a capacidade disponível. Assim que um deles ficar vago, o próximo usuário da fila é chamado. Ao término do serviço, o usuário será acionado em um local destinado para saída do sistema.

Estes conceitos podem ser implementados amplamente em serviços utilizados no dia a dia, para qualquer representação virtual de um recurso físico escasso que possa ser oferecido digitalmente.

2.3. Trabalhos Relacionados

Como mencionado na seção de Motivação, este trabalho teve como base um trabalho de conclusão de curso realizado por um discente da UFRJ¹.

Nesse projeto, o discente descreve de forma detalhada a forma como solucionou o problema de grandes filas na sua instituição. O estudo relacionado possui várias similaridades com esta monografia, por exemplo, utiliza-se também de uma aplicação WEB para fazer a interface com o usuário, contando com um arquitetura cliente-servidor com APIs e do protocolo HTTP⁹, conjuntamente com o modelo REST¹⁰.

Há também diferenças entre os trabalhos, como por exemplo a solução conceitual dada para o problema: aqui, estamos apresentando uma solução utilizando filas virtuais, isto é, o usuário entra em uma fila numa faixa horária, diferentemente do trabalho relacionado, que usa marcação de horários exatos. O modelo de filas foi adotado por conta da experiência cotidiana dos autores com o modelo de marcação de horário, que precisa ter um fluxo de entrada preciso, sem atrasos. Caso contrário, se forma uma fila física na entrada do restaurante universitário, frustrando a experiência e usabilidade do sistema.

3. Desenvolvimento da Plataforma

O desenvolvimento da plataforma se iniciou com uma reunião de planejamento, onde foi realizada uma sessão mais criativa sobre quais funcionalidades poderiam ser adicionadas ao sistema.

Numa segunda reunião, foi feita uma priorização das funcionalidades que seriam capazes de apresentar mais valor ao objetivo proposto neste trabalho. A partir desta, foi possível reconhecer os atores do sistema e realizar a elicitação dos requisitos que serão apresentados abaixo.

Além disso, houve uma reunião para realizar a seleção das tecnologias que os autores acreditavam ser mais relevantes e impactantes.

3.1. Atores do sistema

3.1.1. Usuário do RU

Trata-se do usuário que entrará em uma fila aguardando seu momento de refeição. Podendo ser um aluno, docente ou servidor da instituição. O usuário é reconhecido por seu CPF ou passaporte, quando estrangeiro.

3.1.2. Usuário Administrador

Trata-se do usuário que trabalha no restaurante universitário, e que, portanto, precisa de alguns poderes, tais como fazer o provisionamento das vagas de um determinado horário, adicionar ou remover um usuário da fila.

3.1.3. Sistema

Trata-se do sistema propriamente dito que possui algumas responsabilidades invisíveis aos usuários, como o controle de filas, remanejamento, abertura de novos horários de filas, etc.

3.2. Elicitação de requisitos

3.2.1. Requisitos Funcionais

A Tabela 1 lista e descreve os requisitos funcionais presentes no sistema.

Tabela 1. Requisitos Funcionais

| Requisito | Descrição | Motivação |
|-----------|---|---|
| RF1 | O usuário do RU pode solicitar a entrada na fila. | Permitir que o usuário possa entrar na fila de um horário de sua escolha. |
| RF2 | O usuário do RU pode listar os horários disponíveis. | Permitir que o usuário possa visualizar os horários disponíveis. |
| RF3 | O usuário do RU pode solicitar a retirada da fila. | O usuário pode desistir de uma fila (para entrar em outra, por exemplo). |
| RF4 | O usuário do RU pode se autenticar na plataforma. | O usuário já registrado pode autenticar na plataforma, onde ele terá acesso a todo o sistema de filas. |
| RF5 | O usuário do RU pode se registrar na plataforma. | O usuário pode se cadastrar na plataforma. |
| RF6 | O usuário administrador pode conferir a fila de usuários por horário. | O usuário que administra os acessos presenciais ao RU tem que ser capaz de observar todos os usuários da fila. |
| RF7 | O usuário administrador pode configurar horários de início e fim do atendimento por turnos. | O usuário que administra filas é capaz de configurar no sistema os horários de início e fim do atendimento presencial. |
| RF8 | O usuário administrador pode configurar o número de horários disponíveis por dia. | O usuário que administra filas é capaz de configurar no sistema a quantidade de horários disponíveis por dia. Ex: Se horário início é 9:00, e fim 21:00, com uma quantidade de 12 horários por dia teremos um horário de 9 às 10, de 10 às 11, etc. |
| RF9 | O usuário administrador pode configurar a quantidade de vagas por horário. | O usuário que administra filas é capaz de configurar no sistema a quantidade de vagas disponíveis por horário. |

3.2.2. Requisitos Não-funcionais

Do mesmo modo da tabela acima, a Tabela 2 lista e descreve os requisitos funcionais presentes no sistema.

Tabela 2. Requisitos não-funcionais

| Requisito | Descrição |
|-----------|--|
| RNF 1 | A senha do usuário deve ser criptografada antes de ser armazenada |
| RNF 2 | O sistema deve ser implementado visando a disponibilização em uma infraestrutura baseada em nuvem. |
| RNF3 | O sistema deve possuir disponibilidade mínima de 98% de <i>uptime</i> nos períodos de 9 às 20h de segunda a sexta. |
| RNF4 | As atividades realizadas pelo orquestrador de filas devem poder ser desativadas de maneira independente das APIs via <i>feature flag</i> . |
| RNF5 | Apenas usuários autenticados (que possuem um token válido) podem utilizar a plataforma |
| RNF6 | Uso de Design responsivo nas interfaces gráficas, isto é, a interface do sistema deverá se comportar adequadamente independente do <i>front-end</i> que será utilizado para acesso – <i>Browser</i> , <i>Smartphone</i> ou <i>Tablet</i> . |

3.2.3. Regras de Negócio

Na Tabela 3, são listadas as regras de negócio que estão presentes no sistema.

Tabela 3. Regras de negócio

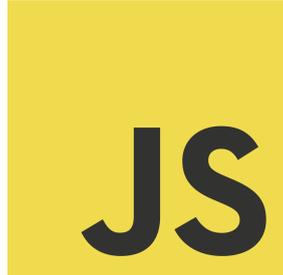
| Regra | Descrição |
|-------|--|
| RN 1 | O usuário deve estar previamente autenticado no sistema para realizar o agendamento. |
| RN 2 | Um usuário (unicamente identificado por um CPF e previamente autenticado na |

| | |
|-------------|---|
| | plataforma) não pode ter mais de um agendamento ativo. |
| RN 3 | O usuário administrador só pode adicionar uma nova configuração de filas para o próximo em diante. |
| RN 4 | O usuário só pode entrar em uma fila por vez. Ou seja, para entrar em uma nova fila, o usuário deve sair de qualquer fila em que esteja presente antes. |

3.3. Seleção das tecnologias

3.3.1. JavaScript

Figura 3. Logotipo JavaScript



Fonte: WILLIAMS, 2011

JavaScript, seu logotipo é apresentado na Figura 3, é uma linguagem de programação usada por desenvolvedores para criar páginas interativas da Internet. As funções de JavaScript podem melhorar a experiência do usuário durante a navegação em um site, podendo ser desde a atualização do feed na página da mídia social até a exibição de animações e mapas interativos. Ao ser aplicado como uma linguagem de *script* do lado do cliente, ele é uma das tecnologias principais da World Wide Web (WWW). Por exemplo, ao navegar na Internet, é possível visualizar a qualquer momento um carrossel de imagens, um menu suspenso “clique para visualizar” ou mesmo mudar dinamicamente as cores dos elementos de uma página da Web. Tudo isso graças ao JavaScript.

3.3.2. Node.js

Figura 4. Logotipo Node.js



Fonte: SILVERWINDX, 2017

O Node.js, com seu logotipo na Figura 4, pode ser definido como um ambiente de execução JavaScript *server-side*. Isso significa que com esse *framework* é possível criar

aplicações Javascript para rodar como uma aplicação autossuficiente (*standalone*) em uma máquina, não dependendo de um browser para a execução, como estamos acostumados.

Neste trabalho, buscamos utilizar o Node.js na intenção de realizar a implementação do software utilizando a mesma linguagem de programação em todo o projeto da construção da aplicação **Minha Bandeja**, o que ajuda consideravelmente a reduzir a complexidade do desenvolvimento e manutenção.

3.3.3. TypeScript

Figura 5. Logotipo TypeScript



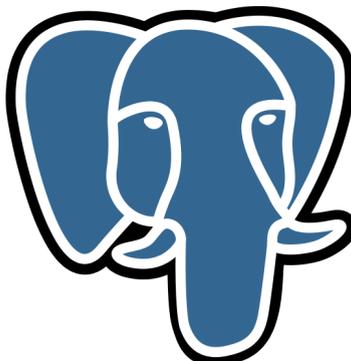
Fonte: TYPESCRIPT, 2012

TypeScript, tendo seu logotipo oficial na Figura 5, é uma linguagem de programação de código aberto desenvolvida pela Microsoft. É um superconjunto sintático estrito de JavaScript e adiciona tipagem estática opcional à linguagem. A tipagem fornece uma maneira de descrever a forma de um objeto, fornecendo melhor documentação e permitindo que o TypeScript valide se seu código está funcionando corretamente. Como TypeScript é um superconjunto de JavaScript, os programas JavaScript existentes também são programas TypeScript válidos.

O typescript foi utilizado neste trabalho por ser a linguagem padrão da ferramenta Nest.js, além de auxiliar imensamente na redução dos erros cometidos durante a codificação por possuir tipagem, algo que o javascript não possui por padrão.

3.3.4. Postgres

Figura 6. Logotipo Postgres



Fonte: LUNDIN, 2008

O Postgres, ou PostgreSQL, com seu logotipo do elefante na Figura 6, é um sistema de banco de dados relacional, ou seja, que armazena e fornece acesso a dados relacionados entre si, com o uso de chaves primárias, índices e restrições de integridade. O Postgres possui código aberto com mais de 35 anos de desenvolvimento ativo, o que lhe rendeu uma forte reputação de confiabilidade e desempenho.

O **Minha Bandeja** utiliza o Postgres como um banco de dados para armazenar as informações mais sensíveis do sistema, como as informações de usuários cadastrados, configurações utilizadas pela aplicação para criação automática e as informações das filas.

Sua seleção foi dada ao fato de que por possuir código aberto, possibilita com facilidade a adoção de uma infraestrutura local (nos servidores da unirio, por exemplo), além de ser um banco já previamente conhecido pelos autores deste trabalho.

3.3.5. Redis

Figura 7. Logotipo Redis



Fonte: PRIOGLIO, 2013

O Redis, com seu logotipo na Figura 7, é um armazenamento de estrutura de dados em memória de código aberto (licenciado pelo BSD) usado como banco de dados, cache, agente de mensagens e mecanismo de streaming. O Redis fornece estruturas de dados como tabelas *hash* e listas, também possuindo transações e diferentes níveis de persistência em disco e fornece alta disponibilidade.

O **Minha Bandeja** utiliza o Redis para armazenar a listagem dos usuários nas filas, sendo um recurso rápido e confiável para armazenagem deste tipo de informação que será acessada e modificada constantemente.

A sua seleção foi dada ao fato de que os autores deste trabalho já possuíam um conhecimento prévio da tecnologia antes da produção do mesmo.

3.3.6. NestJS

Figura 8. Logotipo NestJS



Fonte: STAROŃ, 2017

NestJS, com seu logotipo na Figura 8, é um *framework* para criar aplicativos Node.js eficientes e escaláveis do lado do servidor. Ele usa JavaScript progressivo, é construído e totalmente compatível com TypeScript (mas ainda permite que os desenvolvedores codifiquem em JavaScript puro).

O Nest fornece um nível de abstração acima dessas estruturas Node.js comuns (Express), mas também expõe suas APIs diretamente ao desenvolvedor. Isso dá aos desenvolvedores a liberdade de usar a miríade de módulos de terceiros disponíveis para a plataforma subjacente.

O **Minha Bandeja** utiliza o NestJS na construção do servidor consumido pela aplicação web.

Sua seleção foi dada, principalmente, pela praticidade no desenvolvimento de aplicações robustas com qualidade, e baixa curva de aprendizado.

3.3.7. GIT

Figura 9. Logo Git



Fonte: Wget, 2016

Git é um sistema de controle de versão gratuito e de código aberto projetado para ser utilizado desde projetos pequenos a projetos grandes, de maneira rápida e eficiente.

O Git funciona com ramificações do projeto, que podem ser criadas partindo de um ponto específico sem afetar o código no ramo principal. Antes que essa ramificação seja fundida na principal, uma análise manual pode ser feita por parte dos desenvolvedores em cima das modificações, ou seja, diferenças encontradas pela ferramenta entre estes dois ramos. Quando um desenvolvedor decide que é hora de fundir o ramo, este abre um pedido de fusão, ou *Merge Request*.

Neste trabalho, o Git foi utilizado durante o desenvolvimento para que ambos os autores pudessem trabalhar de maneira independente, ainda que em conjunto.

3.3.8. Sonar Cloud

Figura 10. Logo Sonar



Fonte: Siska Meijer, 2018

Sonar Cloud é um serviço baseado em nuvem de análise estática de qualidade e segurança. Sua principal funcionalidade é a de gerar relatórios sobre códigos alertando sobre possíveis problemas encontrados no código fonte.

Neste trabalho, esta ferramenta foi selecionada pela praticidade de configuração e conhecimento prévio por parte dos autores.

3.3.9. VueJS

Figura 11. Logotipo VueJS



Fonte: YOU, 2014

VueJS, com seu logotipo na Figura 9, é um *framework* progressivo para a construção de interfaces de usuário. Ao contrário de outros *frameworks* monolíticos, Vue foi projetado desde sua concepção para ser adotável incrementalmente. A biblioteca principal é focada exclusivamente na camada visual (*view layer*), sendo fácil adotar e integrar com outras bibliotecas ou projetos existentes. Por outro lado, é perfeitamente

capaz de dar poder a sofisticadas Single-Page Applications quando usado em conjunto com ferramentas modernas e bibliotecas de apoio.

A escolha do VueJS para este trabalho se deu devido à baixa curva de aprendizado, vasta documentação disponível e comunidade ativa, além de ser utilizado na construção das páginas web da aplicação.

3.4. Atividades de qualidade

Para garantir que o trabalho pudesse continuar seguindo em frente com qualidade durante seu desenvolvimento, foram realizados testes unitários e de integração, além da análise estática realizada pelo SonarCloud.

Estas atividades foram configuradas para serem executadas automaticamente na abertura de novos *Merge Requests* no repositório Git. Se alguma dessas análises falha, o desenvolvedor que abriu o *Merge Request* precisa resolver os problemas encontrados antes de prosseguir.

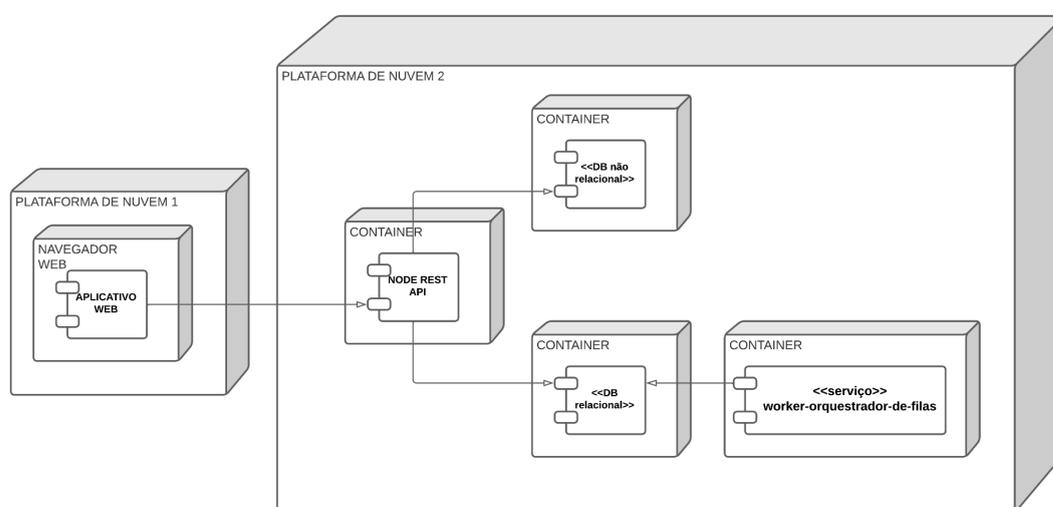
3.5. Arquitetura Proposta

A arquitetura proposta no desenvolvimento da aplicação deste trabalho foi a de cliente-servidor¹¹. Esse modelo mostrou-se mais apropriado por conta das características de disponibilidade de recursos, isto é, a capacidade computacional de cada parte pode ser alterada separadamente, possibilitando a escalabilidade necessária que foi mencionada no Requisito Não-Funcional 01.

Esta arquitetura também mostra-se ideal para implantação na infraestrutura em nuvem ou em outro local, já que é agnóstica ao recurso computacional disponível, por exemplo: a aplicação servidora pode estar rodando em uma máquina em um serviço de nuvem e as aplicações requerentes em outros locais.

A Ilustração 2, abaixo, conta com um diagrama representando de forma resumida a arquitetura proposta neste trabalho.

Ilustração 2. Diagrama da Arquitetura Proposta



Fonte: Autoral

3.5.1. Aplicativo Web

Trata-se da aplicação web que será operada pelo usuário administrador e/ou o usuário do RU. Esta aplicação (chamada de cliente) deve estar hospedada em uma plataforma de nuvem para que o acesso possa ser facilmente realizado de qualquer dispositivo conectado à internet.

3.5.2. Node Rest API

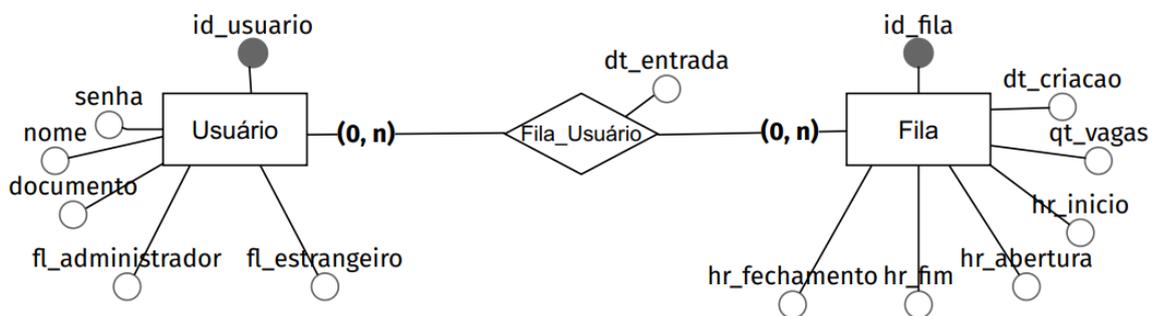
Trata-se do servidor a ser consumido pela aplicação cliente. Este servidor possui a responsabilidade de fazer a interface do cliente com os bancos de dados e aplicar todas as regras de negócio previstas para o projeto.

3.5.3. Banco de Dados relacional

Trata-se do banco de dados que armazena as informações sensíveis do sistema (como os usuários, as filas, e a relação de filas e usuários). A relação de usuários e filas é uma camada de redundância, visto que esta informação está armazenada de forma não relacional para otimizar o acesso no segundo banco de dados (não relacional).

O campo *documento* recebe um nome genérico pois pode ser utilizado tanto para CPF quanto para passaporte, caso *fl_estrangeiro*, que é um campo representando um booleano, seja verdadeiro.

Ilustração 3. Diagrama de Entidade-Relacionamento



Fonte: Autoral

3.5.4. Banco de Dados não relacional

Trata-se do banco de dados que armazena as informações das filas, como a ordem dos usuários numa determinada fila. A sua utilização foi pensada de forma a reduzir o tempo de resposta que o usuário tem na consulta ao banco relacional. As informações inseridas neste banco têm um tempo de vida (TTL) pré-definido, ou seja, são efêmeras.

O que torna o acesso ao Redis, que foi o banco não relacional escolhido neste trabalho, mais rápido em relação ao relacional, é o fato de que o dado não é armazenado em disco, mas em memória¹². Dessa forma, o número de instruções de CPU é reduzido, e a velocidade da consulta é aumentada.

Na Ilustração 4 podemos ver a forma como as filas são armazenadas no Redis. As filas são listas da tupla de *usuarioId*, *nome* e *usuarioDocumento*. A posição da tupla na lista, é a posição do usuário na fila. Toda fila tem uma chave de acesso, que é formada pelo prefixo “filas”, a data da fila, e o seu id, que é um UUID (traduzido como identificador único universal). As diferentes informações são separadas com o separador

4. Apresentação da plataforma

Neste capítulo, apresentaremos as funcionalidades da plataforma, detalhando-as e mostrando o valor entregue por cada uma.

4.1. Implementação das Filas

Foi visualizado que o problema de filas do restaurante universitário poderia ser representado no sistema facilmente utilizando a estrutura de dados Fila, no qual horários distintos teriam filas distintas, e uma pessoa só seria capaz de entrar em uma fila por vez. Também foram utilizadas propriedades da Teoria das Filas, como a capacidade, informada pelo administrador do sistema, a fim de equalizar a capacidade real de atendimento do RU.

Ilustração 5. Representação de filas do sistema



Fonte: Autoral

4.2. Funcionalidades para ambos os perfis

4.2.1. Registro de novo usuário

Figura 12. Formulário de cadastro

Digite as informações requisitadas

Nome

Documento

Senha

Confirme a senha

Força da senha:

Sou estrangeiro

Criar conta e acessar à plataforma

Fonte: Autoral

Logo ao abrir a página de login e clicar no botão de “Novo usuário”, o usuário será redirecionado à tela de cadastro, onde poderá encontrar o formulário observado na figura 11. Este formulário solicita nome, documento (CPF ou passaporte para estrangeiros) e senha.

A informação do documento será utilizada para validar se o usuário é único. Ou seja, o mesmo documento não pode ser cadastrado mais de uma vez.

Ao ser cadastrado com sucesso, o usuário é automaticamente redirecionado à tela de *dashboard*, já autenticado.

4.2.2. Acesso à plataforma

Figura 13. Formulário de acesso



Formulário de acesso à plataforma. O formulário é exibido em uma tela com fundo verde e azul escuro. No topo, há o texto "Digite documento (CPF ou passaporte) e senha". Abaixo, há dois campos de entrada: "Documento (CPF ou passaporte para estrangeiros)" e "Senha". Um botão azul com o texto "Acesso à plataforma" está centralizado abaixo dos campos. No canto inferior esquerdo, há o link "Novo usuário".

Fonte: Autoral

A primeira tela que pode ser vista pelo usuário que deseja acessar a plataforma contém o formulário de login. Ao digitar o documento (CPF para brasileiros e passaporte para estrangeiros), o usuário pode pressionar o botão “Acessar”, que irá redirecionar o usuário à *dashboard*.

4.2.3. Visualizar as filas

Figura 14. Visualização das filas do dia

| FILA | VAGAS | ABERTURA | FECHAMENTO | CAPACIDADE OCUPADA | AÇÃO |
|----------------|-------|----------|------------|--------------------|--------|
| 17:30 às 17:59 | 50 | 09:00 | 17:30 | 0% | Entrar |
| 18:00 às 18:29 | 50 | 09:00 | 18:00 | 0% | Entrar |

Fonte: Autoral

Logo após acessar a plataforma, a primeira página à qual o usuário é redirecionado é a *Dashboard*. Nesta página, o usuário tem as informações de todas as filas disponíveis para o dia. As informações dispostas para cada fila são: Horário de início e fim, quantidade de vagas, horário de abertura da fila, horário de fechamento da fila e a capacidade ocupada da fila (representada por um percentual de completude).

O usuário só poderá entrar em uma fila caso não esteja em uma outra. No caso de o usuário estar em uma fila, todas as outras aparecem com o botão “entrar” desabilitado.

Importante notar que as filas são excluídas de tempos em tempos, de forma que as do passado (que não serão mais consultadas) não geram impacto negativo nesta visualização.

4.2.4. Entrar na fila

Figura 15. Visualização de filas com foco na fila qual o usuário solicitou entrada

| FILA | VAGAS | ABERTURA | FECHAMENTO | CAPACIDADE OCUPADA | AÇÃO |
|----------------|-------|----------|------------|--------------------|--------|
| 14:30 às 14:59 | 50 | 09:00 | 14:30 | 0% | Entrar |
| 15:00 às 15:29 | 50 | 09:00 | 15:00 | 2% | Sair |
| 15:30 às 15:59 | 50 | 09:00 | 15:30 | 0% | Entrar |

Fonte: Autoral

Quando o usuário deseja solicitar acesso à uma fila, basta que este pressione o botão “Entrar”, localizado na coluna de “Ação”.

Após entrar em uma fila, todas as outras ficam automaticamente indisponíveis, e o botão de entrar é desabilitado.

4.2.5. Sair da fila

Como pode ser observado na figura 14, quando o usuário está em uma fila, o botão de “Entrar” é substituído por um botão de “Sair”. Ao pressionar este botão, o usuário é imediatamente retirado da fila.

4.2.6. Sair da plataforma

Figura 16. Menu com botão de sair da plataforma



Fonte: Autoral

Ao clicar no nome do usuário, um pequeno menu pode ser visto com a opção de sair da plataforma. Clicando neste botão o usuário é automaticamente redirecionado para tela de login.

Esta ação não tem nenhum tipo de impacto nas informações das filas, podendo o usuário sair e entrar no sistema quando quiser, permanecendo em uma fila que tenha entrada previamente.

4.3. Funcionalidades exclusivas de usuário administrador

4.3.1. Visualizar as filas

Figura 17. Visualização de filas do administrador

Filas Ativas

| FILA | VAGAS | ABERTURA | FECHAMENTO | CAPACIDADE OCUPADA | AÇÃO |
|----------------|-------|----------|------------|--------------------|--|
| 14:30 às 14:59 | 50 | 09:00 | 14:30 | 0% | Entrar Ver fila |
| 15:00 às 15:29 | 50 | 09:00 | 15:00 | 2% | Entrar Ver fila |

Fonte: Autoral

No caso de um administrador do sistema, a visualização das filas contém a adição de um botão “Ver fila”, que será utilizado por parte do administrador para consultar uma fila com os usuários pertencentes a esta, como pode ser observado na figura 16.

4.3.2. Visualizar usuários na fila

Figura 18. Visualização de uma fila por um administrador

Fila Selecionada: 15:00 às 15:29

| POSIÇÃO | NOME | DOCUMENTO |
|---------|----------------|-------------|
| 1 | Usuario Modelo | 88327451782 |
| 2 | Walter | 04405334730 |

Filas Ativas

| FILA | VAGAS | ABERTURA | FECHAMENTO | CAPACIDADE OCUPADA | AÇÃO |
|----------------|-------|----------|------------|--------------------|--|
| 14:30 às 14:59 | 50 | 09:00 | 14:30 | 0% | Entrar Ver fila |
| 15:00 às 15:29 | 50 | 09:00 | 15:00 | 4% | Sair Ver fila |

Fonte: Autoral

O administrador possui a permissão de observar todos os usuários existentes numa fila com posição, nome e documento. Para tal, basta clicar no botão “Ver fila” na coluna de ação de uma determinada fila.

Na figura 17, vemos duas tabelas. A tabela da parte superior, que tem no título “Fila selecionada”, apresenta uma fila que foi selecionada pelo administrador do sistema a ser observada. Neste caso, a fila é das 15:00 às 15:29. Esta tabela apresenta os usuários que entraram na fila, na ordem que entraram. Vemos então a posição, o nome do usuário e o documento. Este documento é o que será utilizado pelo administrador do sistema para confirmar na porta de entrada do restaurante universitário, fisicamente, que o aluno tentando entrar está realmente na fila.

4.3.3. Alterar configurações de filas

Na tela de configuração de filas, é possível alterar as informações utilizadas pelo sistema para a criação automática de filas do dia. Os campos disponíveis são horário de início de atendimento, horário de fim de atendimento, quantidade de horários disponíveis, quantidade de vagas por horário e a data de início de vigência da nova configuração.

Essas configurações impactam em como o Worker vai criar as filas a partir da vigência prevista. Por exemplo, na figura 18, o almoço tem início às 10h e fim às 16. Considerando os 12 horários disponíveis, serão criadas filas a cada 30 minutos, totalizando às 16 filas para este período. Além disso, cada fila vai ter no máximo 50 vagas. Caso essas vagas sejam totalmente preenchidas, não será mais possível entrar na fila.

Uma regra de negócio aplicada nesta funcionalidade é que uma nova configuração nunca pode ser criada para o dia atual. Apenas para datas posteriores.

Figura 19. Formulário de configuração de filas

| Almoço | Jantar |
|---|---|
| Hora de início de atendimento (0 a 24) <input type="text" value="10"/> | Hora de início de atendimento (0 a 24) <input type="text" value="17"/> |
| Hora de fim de atendimento (0 a 24) <input type="text" value="16"/> | Hora de fim de atendimento (0 a 24) <input type="text" value="23"/> |
| Quantidade de horários disponíveis (Quantas filas serão criadas) <input type="text" value="12"/> | Quantidade de horários disponíveis (Quantas filas serão criadas) <input type="text" value="12"/> |
| Quantidade de vagas por horário <input type="text" value="50"/> | Quantidade de vagas por horário <input type="text" value="50"/> |
| Data de início de vigência da nova configuração <input type="text"/> | Data de início de vigência da nova configuração <input type="text"/> |
| <input type="button" value="Nova configuração para almoço"/> | <input type="button" value="Nova configuração para janta"/> |

Fonte: Autoral

4.3.4. Ver novas configurações de fila agendadas

Figura 20. Visualização de configurações agendadas de configuração de filas

| Configurações futuras: Almoço | | | | | |
|-------------------------------|-------------|----------|-----------------|-------------------|--|
| DATA DE INICIO | HORA INICIO | HORA FIM | QT. DE HORARIOS | VAGAS P/ H ORARIO | AÇÃO |
| 26/01/2023 | 10 | 14 | 16 | 50 | <input type="button" value="Deletar"/> |

| Configurações futuras: Jantar | | | | | |
|-------------------------------|-------------|----------|-----------------|-------------------|--|
| DATA DE INICIO | HORA INICIO | HORA FIM | QT. DE HORARIOS | VAGAS P/ H ORARIO | AÇÃO |
| 26/01/2023 | 16 | 20 | 16 | 25 | <input type="button" value="Deletar"/> |

Fonte: Autoral

Na tela de configuração de filas, disponível apenas para usuários administradores, é possível visualizar agendamentos de novas configurações de filas. Toda vez que uma nova configuração é adicionada, ela aparece automaticamente nesta seção.

A Figura 20 apresenta as tabelas de **almoço** e **jantar**, contendo as configurações criadas para datas no futuro. Nesta tabela, o administrador pode consultar as configurações planejadas para uma data posterior no sistema, assim como deletá-las.

4.3.5. Excluir configuração de fila agendada

Caso o administrador queira remover uma configuração agendada, independente do motivo, este pode simplesmente clicar no botão “Deletar”, como pode ser observado na Figura 20.

5. Uso hipotético do sistema

5.1. Configurações iniciais

Todas as filas do sistema são criadas automaticamente assim que um novo dia se inicia. Essa responsabilidade é exercida pelo próprio sistema. Entretanto, as informações que o sistema utiliza para realizar a criação das novas filas são inseridas pelo administrador do sistema, que é um funcionário do próprio restaurante universitário.

Antes de realizar esse cadastro, o administrador precisa de algumas informações. São elas:

- Horário de início e fim do almoço;
- Horário de início e fim do jantar;
- Número de assentos disponíveis a serem ocupados;
- Tempo médio de duração de uma fila completa (para preencher todos os assentos no horário)

Com essas informações em mãos, o usuário administrador entra acessa a plataforma, e vai até a tela de configurações de filas, que pode ser localizada na barra lateral esquerda da aplicação. O administrador verá a tela representada na figura 18.

Após preencher o formulário com as informações mencionadas acima, o administrador deverá selecionar uma data de início de vigência, ou seja, data qual a aplicação é esperada começar a funcionar com aquela configuração. Após selecionar a data, clicar em “Nova configuração para almoço” e/ou “Nova configuração para jantar”, logo após isso, o sistema será responsável por automaticamente gerar as filas a partir da data selecionada, e nada mais precisa ser feito.

5.2. Utilização no dia a dia pelos alunos e servidores

Antes de poder acessar a plataforma, os usuários devem fazer o cadastro na aplicação, com informações simples como nome, documento (cpf ou passaporte, no caso de estrangeiros) e senha de acesso, tal como podemos observar na figura 11. Usualmente, o usuário utilizará seu documento e senha, tal como podemos observar na

figura 12, para fazer o login na plataforma diariamente, onde este será redirecionado para a ***Dashboard***, onde o usuário terá acesso à todas as funcionalidades de interação com as filas.

As filas ficam disponíveis todos os dias, por padrão, a partir das 9h da manhã, tanto de almoço quanto de jantar. Isso significa que todos os usuários interessados em fazer uma refeição no restaurante universitário naquele dia podem se planejar de acordo, para logo no primeiro horário reservarem seus lugares nas filas dos horários almejados.

Ao acessar a plataforma, o usuário vai ver apenas as filas do dia em foco (a aplicação exclui automaticamente as filas anteriores). Como podemos observar na Figura 13 (pág. 36) a informação principal de uma fila, que será utilizada pelos usuários como referência, é seu horário (de início e fim), sendo o fim geralmente 1 minuto antes do início do próximo horário. Junto a esta informação, o usuário também vê a quantidade de vagas e a ocupação da fila, que é importante pois os usuários que desejam fugir de confusões e/ou possíveis atrasos podem optar por entrar em filas mais vazias.

Após escolher um horário de interesse, o aluno seleciona a opção “Entrar”, após isso ele já estará ocupando um lugar naquela fila. Caso o usuário queira entrar em uma fila que já atingiu 100% de ocupação, ele receberá uma mensagem de erro, notificando-o de que a fila atingiu o limite máximo de usuários. Isso é importante, também, pois o usuário tem uma informação que pode ajudá-lo a se planejar, para fazer uma refeição em um outro horário que ainda tenha vagas, ao invés de simplesmente entrar numa fila onde ele muito provavelmente não será atendido em tempo hábil.

O sistema impede que um usuário entre em mais de uma fila ao mesmo tempo. Isso significa que caso o usuário queira entrar numa nova fila antes de sua fila ter sido deletada (que é feito automaticamente pelo sistema em até uma hora após o fechamento da fila), seja por motivo de confusão ou um clique errado, este deve sair da fila em que segura uma posição naquele momento, para que possa habilitar novamente a possibilidade de entrar em uma outra fila.

Quando chega o horário da fila, o usuário deve se dirigir até o restaurante universitário com seu documento em mãos (CPF ou passaporte, no lugar de estrangeiro). Lá, ele fará interface com um usuário administrador, que terá as informações da fila daquele horário, e conferirá todas as informações, dando preferência para os usuários das primeiras posições da fila.

5.3. Utilização no dia a dia pelos administradores do sistema

Para cada horário de fila disponibilizado aos alunos e servidores, o administrador pode consultar as informações de nome, documento e posição na fila.

Quando chega o horário de uma determinada fila, o administrador deve clicar no botão “ver fila” da respectiva fila, tal como pode ser observado na figura 14, para que seja possível verificar todos os usuários que fazem parte daquele horário.

Os usuários serão identificados rapidamente por seus documentos, e caso seja confirmada sua presença na fila, serão liberados para entrar no restaurante universitário. No caso de não estarem na fila, estes serão instruídos a entrar na fila do próximo horário diretamente pela aplicação.

6. Conclusão

A seguir serão apresentadas as principais conclusões e considerações obtidas ao longo do processo de desenvolvimento e construção deste trabalho.

6.1. Contribuições

Quando os autores deste trabalho idealizaram o desenvolvimento da plataforma proposta, era sabido que o problema das imensas filas não seria e não é de simples solução. Na construção deste sistema, o cerne sempre foi projetar um *software* onde a comunidade acadêmica da UNIRIO pudesse usufruir de uma melhora na qualidade de vida em sua vida universitária, onde o planejamento, mesmo sendo um intervalo de tempo e não uma hora exata, seria o mecanismo onde levaria a esse objetivo.

6.2. Limitações

Primeiramente, é de grande importância ressaltar que o sistema apresentado neste trabalho se trata do *minimum viable product* (MVP) da plataforma Minha Bandeja, isto é, tendo sido implementado as funcionalidades essenciais. Além disso, é importante lembrar que a instituição não foi consultada durante o processo de desenvolvimento, o que não nos permitiu endereçar aspectos técnicos específicos do sistema no que concerne a sua operacionalização dentro da Universidade. Visto isso, faz-se necessário relatar algumas limitações da aplicação.

Atualmente, umas das restrições é não ser possível trocar a senha para realizar o acesso à plataforma. Acrescentado a isso, não há uma validação para saber se um usuário faz parte da comunidade acadêmica da UNIRIO, por não existir integração com a base de dados da instituição.

Além disso, optamos por uma aplicação WEB por questão de suporte a multi plataforma, pois basta apenas um navegador conectado à internet para acessar o aplicativo. Contudo, isso implica em algumas limitações que o sistema não possuiria se houvesse um aplicativo móvel para *smartphones* disponível nas lojas dos sistemas operacionais *mobile*. Pode-se destacar, por exemplo, o envio de notificações para

informar sobre o andamento das filas, uma autenticação à plataforma mais rápida utilizando sistemas de biometria e até mesmo um aprimoramento na usabilidade.

6.3. Trabalhos futuros

Como trabalhos futuros, a plataforma poderia se tornar um sistema utilizado em diversas outras universidades que possuem o mesmo problema de longas filas ou que desejam utilizá-lo antes mesmo dele acontecer. Desse modo, outras funcionalidades também poderiam ser agregadas à aplicação para aumentar a segurança dos seus usuários e melhor usabilidade. Pode-se destacar as seguintes implementações:

- Integração com a base de dados da UNIRIO para validade das informações cadastrais;
- Notificações sobre o estado das filas;
- Recuperação de senha utilizando um envio de e-mail;
- Autenticação na plataforma por outros meios, como integração com redes sociais ou com o próprio ecossistema de identificação da universidade (id.unirio);
- Emissão de relatórios sobre o uso do RU;
- *Dashboard* enriquecida com dados e métricas sobre o uso do RU;
- Mecanismos de avaliação da plataforma;
- Pagamento pela plataforma via meios digitais;

Do mesmo modo que as funcionalidades foram pontuadas acima, seria interessante o desenvolvimento de um aplicativo para dispositivos móveis com as funcionalidades já presentes na plataforma WEB, utilizando-se da aplicação *back-end* construída neste trabalho, trazendo mais usabilidade e comodidade para seus usuários.

Além de tudo, um trabalho futuro relevante seria a implantação concreta do sistema, podendo servir de base para futura discussão dentro da instituição junto à Pró-Reitoria de Assuntos Estudantis (PRAE), que não foi possível por limitações de tempo e questões burocráticas que tange a execução. Porém, tem-se a confiança de que este trabalho irá alavancar a discussão sobre a dor de longas filas, os impactos que isto tem na UNIRIO e trazer à luz as necessidades da instituição.

Referências Bibliográficas

- ¹ DOMINGUES, Manoel. **Sistema de Controle de Acesso e Agendamento para o Restaurante Universitário**. 2019. Disponível em: <https://www.researchgate.net/publication/341273320_Sistema_de_Controlde_de_Acesso_e_Agendamento_para_o_Restaurante_Universitario>. Acesso em: 23 jul. 2022.
- ² **Inaugurado o restaurante-escola da UNIRIO nesta sexta-feira, 26 de agosto**. UNIRIO, 2016. Disponível em: <<http://www.unirio.br/news/inaugurado-o-restaurante-escola-da-unirio-nesta-sexta-feira-26-de-agosto-1>>. Acesso em: 20 jan. 2023.
- ³ **Quantitativo de refeições 2022**. UNIRIO, 2022. Disponível em: <<http://www.unirio.br/estudante/prae/nutricao-prae-1/graficos-qtd-refeicoes-restaurante-quantitativo-de-refeicoes-2022>>. Acesso em: 20 jan. 2023.
- ⁴ MAIA, Priscila. **Restaurante Escola**, 2023. Disponível em: <<http://www.unirio.br/prae/nutricao-prae-1/setan/restaurante-escola>>. Acesso em: 16 jan. 2023.
- ⁵ CARTER, Michael W.; PRICE, Camille C.; RABADI, Ghaith. **Operations Research: A Practical Introduction**. Boca Raton: CRC Press, New York, 2^a ed. 2017.
- ⁶ HILLIER, Frederick S.; LIEBERMAN, Gerald J. **Introdução à Pesquisa Operacional**. AMGH Editora Ltda, 9^a ed. 2013.
- ⁷ TOMÉ, Fernanda *et al.* **Teoria das filas: um estudo de caso em uma biblioteca da serra gaúcha**. São Paulo, 2019.
- ⁸ FOGLIATTI, Maria Cristina; MATTOS, Néli Maria Costa. **Teoria de filas**. Rio de Janeiro: Ed. Interciência, 2006.
- ⁹ FIELDING, Royet *al.* **Hypertext transfer protocol– HTTP/1.1**, Report, 1999.
- ¹⁰ FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. Ph.D. dissertation, University of California, 2000.
- ¹¹ PFLEEGER, Shari Lawrence. **Engenharia de Software: Teoria e Prática**, São Paulo: Prentice Hall, 2^a Edição, 2004.
- ¹² **Introduction to Redis, Redis**. Disponível em: <<https://redis.io/docs/about/>>. Acesso em: 27 jan. 2023.