

# SAPFI: um sistema de alerta para evitar aglomerações em filas de espera de praças de alimentação

Lucas H. M. Alcântara<sup>1</sup>, Nathan S. de Faria<sup>1</sup>, Lesandro Ponciano<sup>1</sup>

<sup>1</sup>PUC Minas em Contagem  
Bacharelado em Sistemas de Informação

lucashmalcantara@gmail.com, nathansfaria@hotmail.com, lesandrop@pucminas.br

**Resumo.** *Filas de atendimento surgem quando há uma demanda maior do que a capacidade de atendimento. Com isso, aglomerações de pessoas podem ser geradas. A pandemia de COVID-19 no ano de 2020 ressaltou a importância de evitar ou amenizar aglomerações, visto que esta é uma situação que favorece a disseminação de doenças. Outros problemas também podem surgir a partir da aglomeração de pessoas, como estresse, claustrofobia e brigas. Dentre as situações que propiciam aglomerações, estão as filas de esperas de restaurantes em praças de alimentação, que muitas vezes não podem ser evitadas. Sendo assim, torna-se necessário o emprego de sistemas de informação para ajudar a amenizá-las, assim proporcionando aos estabelecimentos comerciais oferecer um serviço mais seguro e de melhor qualidade para seus clientes. O objetivo deste trabalho é desenvolver um Sistema de Alerta de Posição em Filas (SAPFI) que permite aos clientes acompanharem a distância a quantidade de pessoas na fila de um determinado estabelecimento e alertar os clientes que já estão na fila quando o seu pedido estiver próximo de ser finalizado. Desta maneira, o sistema desenvolvido ajuda as pessoas a manterem o distanciamento social e, como consequência, a diminuir aglomerações. O sistema desenvolvido apresenta funcionalidades necessárias para resolver o problema pelo qual foi proposto e se mostra promissor para implantação em cenários reais.*

## 1. Introdução

A fila de espera é uma forma de organização social muito comum e indissociável da vida urbana. O surgimento de uma fila de espera ocorre quando há uma demanda maior do que a capacidade de atendimento e, com isso, problemas como estresse, ansiedade e aglomeração de pessoas podem ser observados como produtos dessa situação [Iglesias 2007]. A pandemia de COVID-19, com início em dezembro de 2019, do novo coronavírus (SARS-CoV-2), trouxe à tona os riscos e a importância de se evitar aglomerações [Aquino et al. 2020].

A doença causada pelo SARS-CoV-2 possui alta transmissibilidade e, portanto, novos e numerosos casos surgem a cada dia. De acordo com dados publicados pela Organização Mundial da Saúde (OMS) no dia 27 de maio de 2021, a situação global possui mais de 168 milhões de casos confirmados de COVID-19 e um número de mortes que ultrapassa 3,4 milhões<sup>1</sup>. Sabendo-se da alta transmissibilidade do vírus, o distanciamento social é uma das principais medidas para controle da pandemia de COVID-19 [Aquino et al. 2020].

---

<sup>1</sup>Situação global COVID-19 - Acesso disponível em: <https://covid19.who.int/>. Último acesso em 27 de maio de 2021

Aglomeracões podem causar diversos problemas como, por exemplo, brigas, estresse, claustrofobia e transmissão de doenças. Esses problemas sociais devem ser avaliados para que medidas possam ser implementadas para evitá-los. Durante uma pandemia, uma aglomeração é extremamente perigosa para o agravamento das contaminações, pois basta que apenas uma pessoa tenha a doença para que várias sejam contaminadas, simplesmente por estarem muito próximas umas das outras. Uma situação recorrente que pode gerar aglomerações são as filas de espera, em que diversas pessoas podem estar fisicamente próximas para aguardar o atendimento. Nesse contexto, a questão abordada neste trabalho é: como projetar e implementar um aplicativo de alerta para ajudar a reduzir ou evitar aglomerações causadas por filas de espera?

Em diversos segmentos do mercado, a geração de filas presenciais pode ser evitada através de agendamento para prestação do serviço. Por exemplo, uma clínica médica pode evitar o surgimento de uma fila presencial através do agendamento do atendimento médico, onde as pessoas são alocadas de forma a não gerar gargalo no fluxo de atendimento. Porém, em alguns casos não é possível aplicar o processo de agendamento, pois a decisão do cliente é tomada quando a necessidade surge, como é o caso de restaurantes de praças de alimentação com grande fluxo de pessoas. Nessas situações, o cliente deve: (i) entrar em uma fila para realizar o seu pedido e; (ii) após receber o *ticket* de atendimento, permanecer próximo ao local de informe, geralmente observando um painel informativo ou aguardando ser chamado.

Para esses casos, uma forma de evitar o surgimento de aglomerações é disponibilizar um sistema de alerta integrado ao sistema de gerenciamento de filas do estabelecimento, que permita ao cliente acompanhar sua posição na fila sem a necessidade de estar próximo ao painel informativo. Também é importante que o sistema informe a previsão de quando seu atendimento será efetivado. Além disso, fornecer ao usuário informações sobre a fila de atendimento de determinado estabelecimento, antes mesmo de se deslocar até o local, também é uma funcionalidade importante, pois possibilita ao usuário escolher o estabelecimento mais vazio ou desistir de se deslocar, caso esteja muito cheio.

O objetivo deste trabalho é propor e desenvolver um aplicativo para dispositivos móveis, denominado Sistema de Alerta de Posição em Filas (SAPFI), que forneça aos usuários informações de filas de atendimento de restaurantes antes mesmo de se deslocarem até o local e, quando estiverem no local, acompanhar a situação atualizada do seu pedido até que ele seja finalizado e a retirada seja feita. Desta forma, o aplicativo possibilita uma maior comodidade e permite que os usuários mantenham o distanciamento social enquanto aguardam seus pedidos, dada a mobilidade proporcionada pelo dispositivo móvel.

Este trabalho está estruturado em 6 seções. A Seção 2 apresenta o referencial teórico. A Seção 3 descreve os trabalhos relacionados. A Seção 4 apresenta a metodologia usada na elaboração do trabalho. A Seção 5 contempla a especificação, projeto, implementação e avaliação do SAPFI. Por fim, a Seção 6 apresenta a conclusão.

## **2. Referencial Teórico**

Esta seção apresenta uma análise teórico-conceitual dos principais tópicos associados ao trabalho, são eles: sistemas de alerta, sistemas de gerenciamento de filas e aglomerações.

## 2.1. Sistemas de Alerta

Um sistema de alerta pode ser abstraído em dois componentes: o componente de avaliação de dados e o componente de interação com o usuário [Poseiro et al. 2014]. O componente de avaliação de dados integra a informação relevante para a identificação de um alerta. O sistema proposto usa a quantidade de pessoas, previsões de tempo de espera e a posição do usuário na fila de forma integrada para alertar o usuário quando o seu pedido estiver disponível ou próximo de ser finalizado; e para alertar o usuário quando for a melhor ocasião para se deslocar até o estabelecimento, levando em consideração as condições da fila. O componente de interação com o usuário propôs um aplicativo para dispositivos móveis, onde o usuário pode realizar a parametrização para acompanhar o seu atendimento ou receber o alerta para quando a fila de um determinado estabelecimento estiver mais vazia. O SAPFI não objetiva prever uma situação de aglomeração ou sobrecarga na capacidade de atendimento para gerar o alerta; ele processa os dados para alertar sobre uma situação que está ocorrendo.

## 2.2. Sistema de Gerenciamento de Filas

Existem duas classificações principais para filas de espera: presenciais e não presenciais [Iglesias 2007]. A fila de espera presencial pode ser definida como uma estrutura constituída por pessoas cada qual com sua personalidade, e demanda comportamento específico, como a permanência em pé ou sentado, respeitando uma determinada ordem de chegada ou prioridade. Incluem-se nessa categoria as filas organizadas por agendamento prévio ou por sistemas de senhas, visto que o atendimento é presencial [Iglesias and Günther 2009]. As filas não-presenciais são filas invisíveis, ou seja, não exigem a presença física do usuário. Alguns exemplos de filas não-presenciais são as filas de atendimento telefônico e sistemas de chamados *online* [Iglesias and Günther 2009]. Este trabalho está no escopo de filas presenciais, mais especificamente as que utilizam sistemas de senhas, pois esses fornecem os dados necessários para o funcionamento do SAPFI.

As filas presenciais podem ser divididas em dois tipos: (1) filas únicas, onde as pessoas ficam organizadas fisicamente de acordo com a ordem de chegada, no modelo “o primeiro que entra é o primeiro que é atendido” (típicas de restaurantes brasileiros); (2) filas múltiplas, o mesmo modelo de atendimento das filas únicas, todavia com múltiplos postos de atendimento (típicas de supermercados). Também pode ser incluída na classificação de filas presenciais a fila por agendamento prévio ou por sistema de senhas, onde o usuário deve comparecer para o atendimento no horário marcado (típica de clínicas para atendimento)[Iglesias 2007].

Sistemas de gerenciamento de filas são soluções de *software* e *hardware* para gerenciar filas e fluxos de atendimentos [Alsaeed and Alhazmi 2019] [Leite et al. 2016]. Em geral, esses sistemas são voltados para empresas que possuem grande afluência de clientes, possibilitando maior controle, como por exemplo, melhor disposição dos clientes no espaço físico disponível e gerenciamento dos dados coletados pelo sistema, bem como agilidade no atendimento. No que tange ao cliente dessas empresas, essa classificação de sistema pode proporcionar economia de tempo, comodidade e redução no nível de insatisfação com o serviço prestado [Alsaeed and Alhazmi 2019] [dos Santos Oliveira et al. 2017] [Ghazal et al. 2016].

Um exemplo de *software* nessa categoria é o Sistema de Gerenciamento do Aten-

dimento versão Livre (SGA). É desenvolvido pela Empresa de Tecnologia e Informações da Previdência Social (DATAPREV), totalmente baseado em tecnologias de *software* livre e de código aberto (*open source*). Sistemas de gerenciamento de filas podem agilizar e aperfeiçoar o atendimento, como foi observado pelo Sistema Nacional de Emprego (Sine) de Itaúna<sup>2</sup> e Unidade de Atendimento ao Trabalhador (UAT), do bairro Gameleira em Belo Horizonte<sup>3</sup>, após a implantação do SGA.

### 2.3. Aglomerações

De acordo com o dicionário Priberam<sup>4</sup>, aglomeração é o conjunto de coisas ou de pessoas juntas, é por junto ou juntar no mesmo lugar. Aglomerações podem ser encontradas em diversos locais diferentes, são causadas, como já definido anteriormente, pela junção de várias pessoas em um pequeno espaço. Aglomerações são relativas, ou seja, não se trata apenas de quantidade, mas sim da disposição física dos indivíduos em um determinado espaço, seja este espaço aberto ou fechado. Em uma situação de pandemia, há diretrizes que indicam que aglomerações acima de 250 pessoas devem ser evitadas, outras que acima de 50 pessoas já há perigo e em alguns casos, agrupamentos de mais de 10 pessoas já são considerados nocivos<sup>5</sup>. De acordo com a Organização Mundial da Saúde (OMS, do inglês WHO – *World Health Organization*), em situações de aglomeração, é recomendável estabelecer um distanciamento social de pelo menos um metro da próxima pessoa e, em ambientes fechados, esta distância deve ser maior ainda<sup>6</sup>.

## 3. Trabalhos Relacionados

Dentro desta seção são apresentados os trabalhos relacionados ao escopo deste trabalho, que são soluções de sistemas de informação para filas de espera. Dentre estes trabalhos há soluções comerciais e estudos de caráter empírico.

Em Alsaeed e Alhazmi (2019), os autores discutem sobre os diversos transtornos causados pela espera em filas de atendimento, como: frustração pelo desperdício de tempo na espera, superlotação, congestionamento e transmissão de doenças. Foi realizado um estudo sobre diversos aplicativos de gerenciamento de filas e suas características, com o objetivo de sugerir um sistema inteligente de gerenciamento de filas com base espacial. Como resultado do estudo, os autores obtiveram diversos dados sobre diferentes sistemas de gerenciamento de filas e, com isso, definiram que um sistema ideal deve minimizar o tempo de espera, ser aplicável a todo tipo de organização, utilizar da geografia a favor

---

<sup>2</sup>Sistema SGA Sine Itaúna - Acesso disponível em: <https://g1.globo.com/mg/centro-oeste/concursos-e-emprego/noticia/nova-ferramenta-promete-agilizar-atendimentos-do-sine-de-itauna-mg.ghtml>. Último acesso em 18 de outubro de 2020

<sup>3</sup>Sistema SGA UAT Gameleira, Belo Horizonte -Acesso disponível em: <https://setelagoas.com.br/noticias/minas/36353-sedese-trabalha-na-modernizacao-das-unidades-do-sine-em-minas-gerais>. Último acesso em 18 de outubro de 2020

<sup>4</sup>Sobre do Aglomerações - Acesso disponível em: <https://dicionario.priberam.org/aglomera%C3%A7%C3%A3o>. Último acesso em 18 de outubro de 2020

<sup>5</sup>Até quantas pessoas podem se reunir com segurança sem provocar o avanço do coronavírus? - Acesso disponível em: <https://www.nationalgeographicbrasil.com/ciencia/2020/03/coronavirus-covid-19-transmissao-aglomeracao-multidao-contagio-pandemia-isolamento>. Último acesso em 11 de março de 2021

<sup>6</sup>Coronavirus disease (COVID-19) advice for the public - Acesso disponível em: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public>. Último acesso em 11 de março de 2021

da solução, permitir relatórios e avaliações, editar e cancelar a requisição de atendimento (*i.e.*, *ticket*) e computar o tempo exato de chegada para comparecer para o atendimento. O artigo estuda soluções para filas, como neste trabalho, porém, faz um estudo comparativo entre soluções do mercado, enquanto este propõe a implementação de um sistema.

Em dos Santos Oliveira et al. (2017), os autores propuseram o desenvolvimento de um sistema de gerenciamento de filas para restaurantes universitários que utiliza o conceito de ocultação da percepção de espera. Um exemplo apresentado pelos autores no artigo mostrou que a ocultação da percepção de espera foi usada pela primeira vez no parque temático *Walt Disney World* em 1998. A ideia inicial era eliminar o ato de esperar em pé na fila no local da atração permitindo que os clientes pudessem ser registrados em uma fila virtual através de um computador. O sistema desenvolvido pelos autores pode ser descrito em três etapas: (1) Entrada na fila virtual, (2) Consultar posição e tempo de espera e (3) validação. O sistema proposto pelos autores apresenta semelhanças em comparação ao que foi desenvolvido neste trabalho, como: desenvolvimento móvel multiplataforma, notificações e informações de posição da fila. Contudo, o sistema apresentado por eles não apresenta uma solução que permite aos usuários acompanharem e serem alertados sobre as filas de espera dos estabelecimentos antes de entrarem nelas.

Em Ghazal et al. (2016), foi desenvolvido um sistema de gerenciamento inteligente de filas que proporciona atualizações visuais e de áudio em tempo real para solicitações de serviços através de um aplicativo móvel. O sistema é uma solução de Internet das Coisas (IoT, do inglês *Internet of Things*) que explora a infraestrutura existente de *smartphones* para atribuir *tickets* aos clientes. O sistema permite que o usuário gere um *ticket* por meio de Comunicação por Campo de Proximidade (NFC, do inglês *Near Field Communication*) ou *Quick Response Code* (QR Code). A partir daí, o usuário pode acompanhar o andamento da fila através de seu *smartphone*, tendo a liberdade de permanecer ou não na sala de espera. O usuário também tem à disposição um conjunto de multimídia para entretenimento, como a opção *streaming* de áudio de TV, *download* de revistas e jornais, tudo isso com vistas a melhorar a qualidade do tempo de espera. O sistema SAPFI propõe melhoria no tempo de espera e permite que o usuário espere pelo atendimento dentro ou fora da sala de espera, característica compartilhada com o sistema proposto por Ghazal et al. (2016), porém o SAPFI apresenta a funcionalidade que permite ao cliente a visualização prévia da situação da fila de um determinado estabelecimento, antes mesmo de se deslocar até o local para fazer o pedido.

A plataforma Orchestra 7, desenvolvido pela empresa Qmatic<sup>7</sup>, é um sistema modular que vai desde um sistema de gerenciamento de filas básico até uma plataforma robusta para o acompanhamento completo da jornada do cliente na empresa. Por ser uma plataforma modular, ela oferece o ajuste exato à estrutura da organização, oferecendo ferramentas essenciais como *tickets* móveis para os clientes formarem filas virtuais, notificação para a equipe, informando informações sobre os clientes, terminais de *check-in* para instruir os clientes a seguirem as próximas etapas, entre outros. Além disso, também são oferecidos serviços mais avançados que auxiliam as empresas a oferecerem os melhores serviços, como o *feedback* pós atendimento do cliente, conteúdo Digital *Signage* para envolver e informar os clientes, além de relatórios criados por *Business Intelligence* (BI), que fornecem análises importantes sobre a empresa e o atendimento por

---

<sup>7</sup>Qmatic - Acesso disponível em: <https://www.qmatic.com/>. Último acesso em 04 de outubro de 2020

ela oferecido.

A solução Orchestra 7 apresenta, como semelhanças ao sistema desenvolvido nesse trabalho, a disponibilização de um aplicativo móvel, para que os clientes dos estabelecimentos possam acompanhar em tempo real o andamento da fila no qual se encontram. Porém, como principal diferença, esse sistema não aparenta ter uma integração com outros sistemas de gerenciamento de filas, ficando obrigatório aos estabelecimentos realizarem toda a instalação dos sistemas Qmatic para terem acesso a funcionalidades como a de alerta de posição em fila em tempo real via aplicativo para dispositivos móveis.

O Get In<sup>8</sup> é uma plataforma voltada para bares e restaurantes que tem como objetivo oferecer uma solução digital para esses segmentos através de serviços de cardápio digital, gestão de filas de espera e gestão de reservas *online*, assim possibilitando a esses estabelecimentos aumentarem sua gama de clientes, oferecer serviços melhores e aproveitarem ao máximo o potencial da empresa. Para os clientes, o aplicativo oferece várias vantagens: além de disponibilizar cardápios variados de acordo com a localidade do usuário, ainda oferecem a comodidade de realizar reservas e entrar em filas de esperas totalmente *online* e de forma facilitada. O aplicativo Get In, assim como o sistema apresentado nesse artigo, oferece um sistema de acompanhamento de posição de fila em tempo real para o cliente, além de exibir uma estimativa de horário de atendimento. Porém, a principal diferença é que o aplicativo tem área de atuação mais focada em restaurantes de reserva, enquanto o sistema deste trabalho é voltado para restaurantes que não possuem a possibilidade de reserva.

O sistema proposto por Punn e Agarwal (2019) é um exemplo de sistema de alerta que propõe uma abordagem para gerar sinais de alerta sempre que houver grandes chances de ocorrer uma situação de risco causada por aglomerações em pontes de passagem de pedestres, como pisoteamento. O sistema utiliza algoritmos de detecção de objetos na análise para gerar os alertas. Outro exemplo de sistema de alerta pode ser visto em Poseiro et al. (2014), que objetiva alertar autoridades responsáveis sobre inundações em zonas costeiras e portuárias. Esse sistema possui um módulo de previsão de agitação marítima para determinar os efeitos que podem resultar em algum risco de inundação para a região de interesse. Identificada uma situação perigosa, o sistema envia automaticamente uma mensagem de alerta para as autoridades responsáveis. Em Costa (2016), é implementado o Sistema de Alerta de Risco Epidemiológico (SARE) a fim de agilizar a emissão de alertas aos órgãos competentes para a tomada de providências sanitárias. O sistema gerencia informações de diagnóstico e comunica em tempo real as ocorrências de testes positivos e doenças, agregando confiabilidade e segurança na transmissão dos dados.

Para efeito de comparação, é realizado um levantamento geral das funcionalidades dos sistemas que apresentaram maior similaridade ao proposto neste trabalho, conforme observado na Tabela 1. A comparação mostrou que o sistema com a maior gama de funcionalidades em comum com o SAPFI foi o Orchestra 7. As funcionalidades posição atual na fila e notificações são consideradas como funcionalidades básicas, por estarem presentes em todos os sistemas comparados.

---

<sup>8</sup>Get In - Acesso disponível em: <https://www.getinapp.com.br/>. Último acesso em 04 de outubro de 2020

**Tabela 1. Análise de características dos sistemas relacionados**

	[dos Santos Oliveira et al. 2017]	[Ghazal et al. 2016]	Orchestra 7	Get In	SAPFI
Posição atual da fila	X	X	X	X	X
Horário estimado para atendimento	X		X		X
Notificações	X	X	X	X	X
Histórico de senhas			X		X
Visualização prévia da situação da fila					X

## 4. Metodologia

Este trabalho apresenta a proposta, implementação e validação de um *software*, onde um sistema de alerta para o gerenciamento de informações de filas de espera de restaurantes é desenvolvido e tem sua usabilidade avaliada por meio de questionários aplicados a usuários do sistema.

### 4.1. Etapas

Para o desenvolvimento do trabalho, são realizados os seguintes processos:

- I) Levantamento dos Requisitos
- II) Projeto do Sistema
- III) Sistema Desenvolvido
- IV) Resultado da Avaliação de Usabilidade

Na etapa de levantamento de requisitos, são identificadas as principais funcionalidades e regras de negócio para implementação do sistema. Parte desses requisitos vem da etapa de análise de sistemas relacionados, parte de análises de mercado e parte de pesquisas avançadas sobre a área de foco do trabalho. Ainda nesta etapa, é desenvolvido o diagrama de casos de uso (*use case*), utilizando-se da ferramenta de modelagem de diagramas Astah Community<sup>9</sup>, tendo como finalidade demonstrar os requisitos funcionais do sistema e a interação direta dos atores para com o sistema. Além disso, são desenvolvidos os *wireframes*, uma forma visual de expressar a ideia e estrutura de uma aplicação [Eckschmidt et al. 2019]. Os *wireframes* são artefatos de *design* de baixa fidelidade que apresentam de forma simplificada como o produto final deve funcionar [Rios et al. 2018]. Para o desenvolvimento desses artefatos foi utilizada a ferramenta Figma<sup>10</sup>, uma ferramenta de criação e teste de *design* colaborativo.

Na etapa de projeto do sistema, é feita a criação dos modelos que vão nortear o processo de desenvolvimento. Estes modelos são os diagramas de componentes e implantação, desenvolvidos na ferramenta Astah Community, e entidade e relacionamento, desenvolvido utilizando a ferramenta de modelagem de diagramas do DBEaver: diagrama de componentes, para demonstrar os componentes internos do sistema e suas

<sup>9</sup>Astah Community - Acesso disponível em: <https://astah.net/products/astah-community/>. Último acesso em 10 de maio de 2021

<sup>10</sup>Figma - Acesso disponível em: <https://www.figma.com/>. Último acesso em 10 de maio de 2021

ligações; diagrama de implantação, para demonstrar a arquitetura de *software e hardware* necessária para o funcionamento da aplicação; diagrama de entidade e relacionamento, para demonstrar a estruturação do banco de dados e seus relacionamentos.

Na etapa de sistema desenvolvido, é feita a criação de toda a estrutura algorítmica do sistema e todas as funcionalidades e regras de negócios são implementadas de forma a dar ao sistema as funções e usabilidades esperadas. O ambiente de desenvolvimento é detalhado na Seção 4.2. Também é apresentado o processo de publicação da versão final do sistema.

Na etapa resultado da avaliação de usabilidade, é construído o questionário para avaliação da usabilidade do sistema usando a abordagem do *System usability Scale* (SUS). Também é desenvolvido o material necessário para contextualizar e direcionar os participantes ao ambiente simulado para uso do sistema. Por fim, os testes são aplicados para gerar os dados necessários para a etapa de apresentação e análise dos resultados.

## 4.2. Ambiente de Desenvolvimento

O ASP.NET Core é um *framework* de código aberto (do inglês, *open source*) para o desenvolvimento de *softwares* modernos e de alto desempenho<sup>11</sup>, e é a tecnologia escolhida para o desenvolvimento da API - baseada no estilo arquitetural *Representational State Transfer* (REST) - do sistema SAPFI. Dentre os motivos da escolha do ASP.NET Core, estão a capacidade da tecnologia atender aos requisitos levantados, o conhecimento dos autores perante a linguagem de programação C# (linguagem do *framework*) e possibilidade de publicar a solução em plataformas diferentes - dada a característica *cross-plataform* do *framework*. O Visual Studio Community 2019 é o Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*) escolhido, pois é a principal ferramenta para desenvolvimento usando C# e é disponibilizada pela Microsoft de forma gratuita para pesquisas acadêmicas<sup>12</sup>.

O React Native é uma biblioteca JavaScript para o desenvolvimento de aplicações híbridas. O código escrito em JavaScript pode ser exportado para as plataformas Android e iOS e é renderizado com código nativo<sup>13</sup>. A escolha do React Native se dá por este ser uma ferramenta robusta, com documentação abrangente e por auxiliar na produtividade no desenvolvimento de aplicações móveis.

O Sistema de Gerenciamento de banco de dados (SGBD) PostgreSQL foi a escolha para a persistência dos dados. O motivo da escolha foi baseado na comunidade ativa que o PostgreSQL tem e sua licença PostgreSQL Licence que permite o uso, cópia, modificação e distribuição para qualquer propósito sem cobrança de taxas<sup>14</sup>.

O Docker é uma plataforma, utilizada principalmente por desenvolvedores, para gerenciar a infraestrutura do desenvolvimento de uma aplicação, fazendo o papel de uma espécie de servidor, porém de forma mais isolada e simplificada. Trabalha com o conceito

---

<sup>11</sup>Introdução ao ASP.NET Core - Acesso disponível em: <https://docs.microsoft.com/pt-br/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-3.1>. Último acesso em 10 de maio de 2021

<sup>12</sup>Visual Studio Community - Acesso disponível em: <https://visualstudio.microsoft.com/pt-br/vs/community/>. Último acesso em 10 de maio de 2021

<sup>13</sup>React Native - Acesso disponível em: <https://reactnative.dev/>. Último acesso em 10 de maio de 2021

<sup>14</sup>PostgreSQL - Acesso disponível em: <https://postgresql.org/>. Último acesso em 10 de maio de 2021

de *containers*, que empacota os códigos e dependências de uma aplicação e facilita a execução deste aplicativo entre diferentes ambientes computacionais. Além disso, os *containers* possibilitam que o trabalho seja executado de maneira muito mais leve e segura se comparado aos modos de infraestrutura tradicionais<sup>15</sup>. Foi usado como infraestrutura para execução da API e do banco de dados no momento do desenvolvimento do sistema.

O sistema GIT<sup>16</sup> foi adotado como meio para o controle de versão de código em conjunto com a plataforma GitHub<sup>17</sup>. A Tabela 2 apresenta de maneira sintética as ferramentas usadas no ambiente de desenvolvimento deste trabalho.

**Tabela 2. Ferramentas do ambiente de desenvolvimento**

Funcionalidade	Ferramenta
API REST	.NET CORE 3.1
Banco de dados	PostgreSQL 11
Gerenciador de <i>Containers</i>	Docker
Aplicação <i>mobile</i>	React Native 0.64
Controle de versão	GIT
Sistema Operacional	Windows 10

### 4.3. Processo de Desenvolvimento

O Scrum foi o *framework* de gerenciamento de projetos ágeis adotado no processo de desenvolvimento do sistema SAPFI, dada a sua característica de entrega de valor no menor tempo possível e na capacidade de agregar adaptações no sistema de maneira simplificada [Cruz 2013]. Essa abordagem foi utilizada em conjunto com o processo de *software* definido por Pressman e Maxim (2016), descrito em 5 etapas: Comunicação, Planejamento, Modelagem, Construção e Entrega [Pressman and Maxim 2016].

O fluxo de trabalho adotado foi o Git Flow<sup>18</sup>, que consiste em um modelo de gerenciamento de *branches* bem sucedido e validado pela comunidade. Foi adotado também o conceito de revisão de código (do inglês, *code review*) para qualquer funcionalidade a ser integrada nas versões incrementais do sistema. A medida que as versões incrementais eram desenvolvidas, o sistema era testado e as correções aplicadas.

As versões incrementais do sistema passam pelo processo de integração contínua e entrega contínua (CI/CD, do inglês *continuous integration/continuous delivery*), que gerou o aumento da produtividade no desenvolvimento. A plataforma GitHub permitiu a configuração do processo através da funcionalidade GitHub Actions<sup>19</sup>, que gera os artefatos da API e realiza a implantação (do inglês, *deploy*) na infraestrutura em nuvem da Microsoft Azure.

<sup>15</sup>Docker overview - Acesso disponível em: <https://docs.docker.com/get-started/overview/>. Último acesso em 10 de maio de 2021

<sup>16</sup>Sistema GIT - Acesso disponível em: <https://git-scm.com/>. Último acesso em 12 de maio de 2021

<sup>17</sup>Plataforma GitHub - Acesso disponível em: <https://github.com/>. Último acesso em 12 de maio de 2021

<sup>18</sup>Git Flow - Acesso disponível em: <https://jeffkreeftmeijer.com/git-flow/>. Último acesso em 12 de maio de 2021

<sup>19</sup>GitHub Actions - Acesso disponível em: <https://github.com/features/actions/>. Último acesso em 12 de maio de 2021

#### 4.4. Avaliação de Usabilidade

A avaliação investigativa do sistema foi realizada através da aplicação da escala SUS. A escala SUS é uma ferramenta de avaliação largamente utilizada [Lewis and Sauro 2018] [Boucinha and Tarouco 2013] para mensurar e avaliar a usabilidade de sistemas, produtos ou serviços [Pradini et al. 2019]. A decisão pelo uso da escala SUS no processo de avaliação do sistema foi feita com base em sua alta confiabilidade e simplicidade na aplicação [Gerald et al. 2019]. O método consiste na aplicação de um questionário composto por 10 questões, onde o usuário assinala sua resposta numa escala Likert que varia de Discordo Completamente a Concordo Completamente [Boucinha and Tarouco 2013]. A Tabela 3 apresenta todas as 10 questões que compõe o questionário SUS. O resultado é a soma da contribuição individual de cada item, que após a aplicação do cálculo proposto por John Brooke em 1986 obtém um índice de satisfação do utilizador que varia de 0 a 100 [Boucinha and Tarouco 2013].

O *score* pode ser classificado a partir de uma grade de valores que vai de 0 a 100, em que produtos com pontuação acima de 90 são considerados *excepcionais*, entre 80 e 90 *bons*, entre 70 e 80 *aceitáveis* e abaixo de 70 pode-se considerar que o produto possui problemas de usabilidade preocupantes [Bangor et al. 2009]. A grade foi usada como base para a interpretação dos resultados obtidos na aplicação da SUS no sistema SAPFI.

**Tabela 3. Questões do questionário SUS**

Nº	Questão
1	Eu acho que gostaria de usar esse sistema com frequência.
2	Eu acho o sistema desnecessariamente complexo.
3	Eu achei o sistema fácil de usar.
4	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5	Eu acho que as várias funções do sistema estão muito bem integradas.
6	Eu acho que o sistema apresenta muita inconsistência.
7	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8	Eu achei o sistema atrapalhado de usar.
9	Eu me senti confiante ao usar o sistema.
10	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

### 5. Especificação, Projeto, Implementação e Avaliação do SAPFI

Esta seção contempla a especificação do sistema por meio do levantamento e modelagem dos requisitos, as definições de projeto com os modelos e protótipos desenvolvidos, as considerações referentes à etapa final de desenvolvimento e publicação do sistema desenvolvido e, por fim, os resultados da avaliação de usabilidade.

#### 5.1. Levantamento dos Requisitos

A partir da observação e análise dos sistemas relacionados, é realizado o processo de engenharia de requisitos. O processo consiste em avaliar os dados levantados na etapa anterior para identificar e documentar os requisitos necessários para a implementação do sistema. A proposta do aplicativo SAPFI é possibilitar ao usuário inserir o número do

*ticket* recebido para acompanhar em tempo real e ser alertado sobre a situação do seu pedido, tudo através de seu *smartphone* e a distância. Além disso, o aplicativo oferece ao usuário a possibilidade de acessar os restaurantes aderentes ao sistema e visualizar informações referentes à fila de atendimento do local, como a quantidade de pessoas na fila e o tempo total necessário para o atendimento. O aplicativo também oferece a funcionalidade para criar um alerta de situação da fila, para que o usuário seja alertado quando a situação da fila estiver de acordo com o parâmetro de quantidade de pessoas inserido.

Na Figura 1 é demonstrado um diagrama de casos de uso que exhibe os atores Usuário e Sistema de Gerenciamento de Filas do Estabelecimento Comercial, e suas respectivas interações. O diagrama foi realizado no padrão da Linguagem de Modelagem Unificada (UML, do inglês *Unified Modeling Language*). No diagrama são especificadas todas as funções que o Usuário exerce e mostra a função do Sistema de Gerenciamento de filas no SAPFI, que é enviar os dados para atualização do estado da fila.

## 5.2. Projeto do Sistema

No projeto do sistema, são apresentados os diagramas de componentes, implantação, entidade e relacionamento, além da prototipagem através de *wireframes*.

### 5.2.1. Diagrama de Componentes

A Figura 2 apresenta o diagrama de componentes da UML desenvolvido para representar os componentes do SAPFI e suas interações. O SAPFI possui uma API REST (*Representational State Transfer*) que engloba o conjunto de componentes responsáveis por fornecer interfaces de comunicação para o aplicativo e para o sistema de gerenciamento de filas do estabelecimento comercial. Essa API também realiza a comunicação com o módulo de envio de *push notifications* do Firebase Cloud Messaging (FCM). O aplicativo SAPFI contempla um conjunto de componentes responsáveis por apresentar os dados e permitir integração do usuário. O aplicativo possui uma interface de comunicação que recebe *push notifications* do FCM. Os componentes classificados como repositório comunicam com a interface fornecida pelo SGBD PostgreSQL.

### 5.2.2. Diagrama de Implantação

O diagrama de implantação da UML (Figura 3) foi desenvolvido para demonstrar a execução da arquitetura do sistema SAPFI. A API, desenvolvida com base no *framework* ASP.NET Core 3.1, é executada no serviço *App Service* da plataforma Microsoft Azure. Essa API se comunica com o SGBD PostgreSQL na versão 11, executado no serviço *Azure Database for PostgreSQL server* também da plataforma Microsoft Azure, e com o FCM para o envio de *push notifications*. O aplicativo desenvolvido com base no React Native na versão 0.64 é executado no dispositivo móvel do usuário, cujo sistema operacional pode ser Android ou iOS. Os sistemas de gerenciamentos de filas, cada um sendo executado em sua própria infraestrutura, realiza comunicação com a API para atualização do estado da fila.

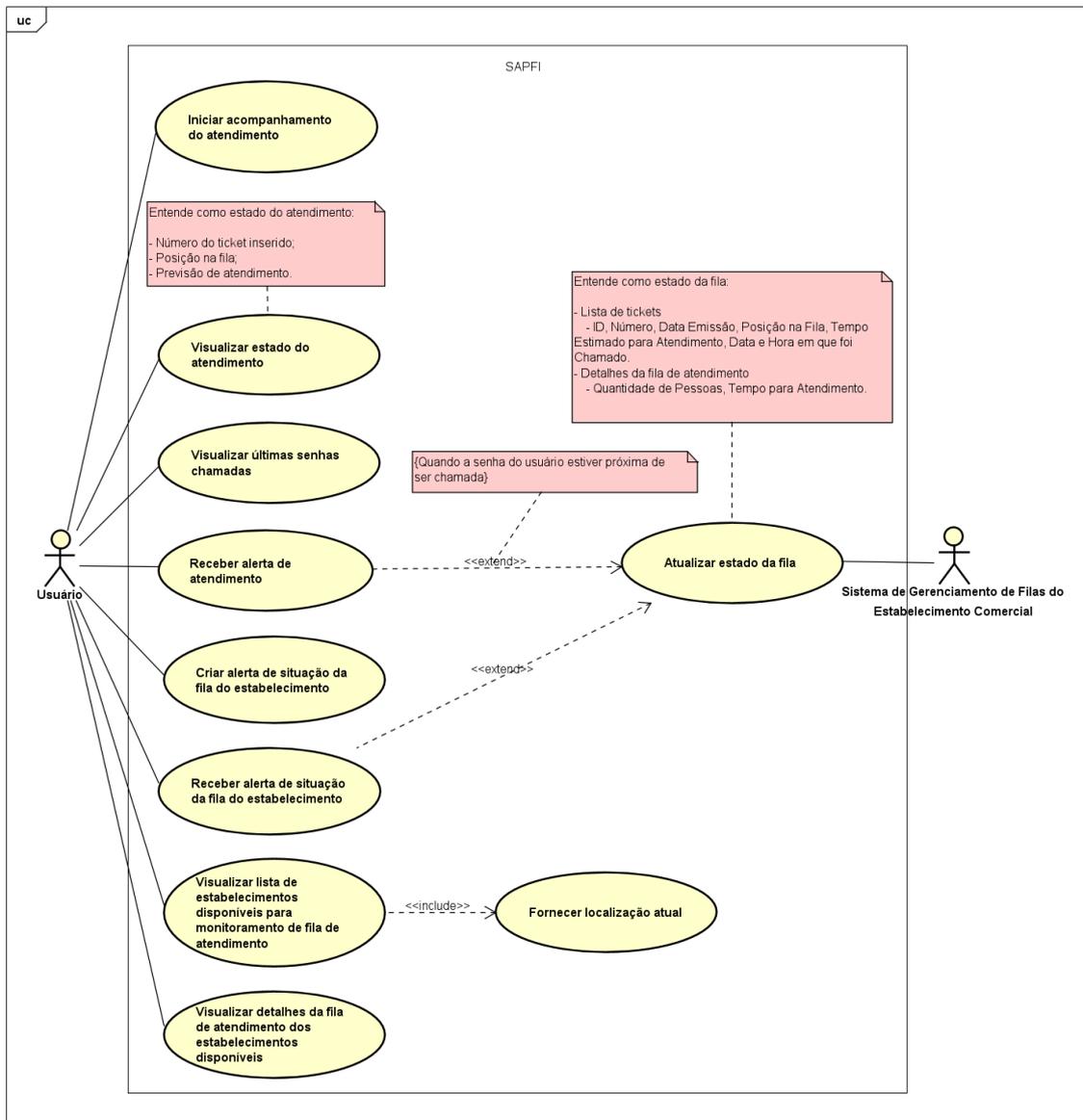
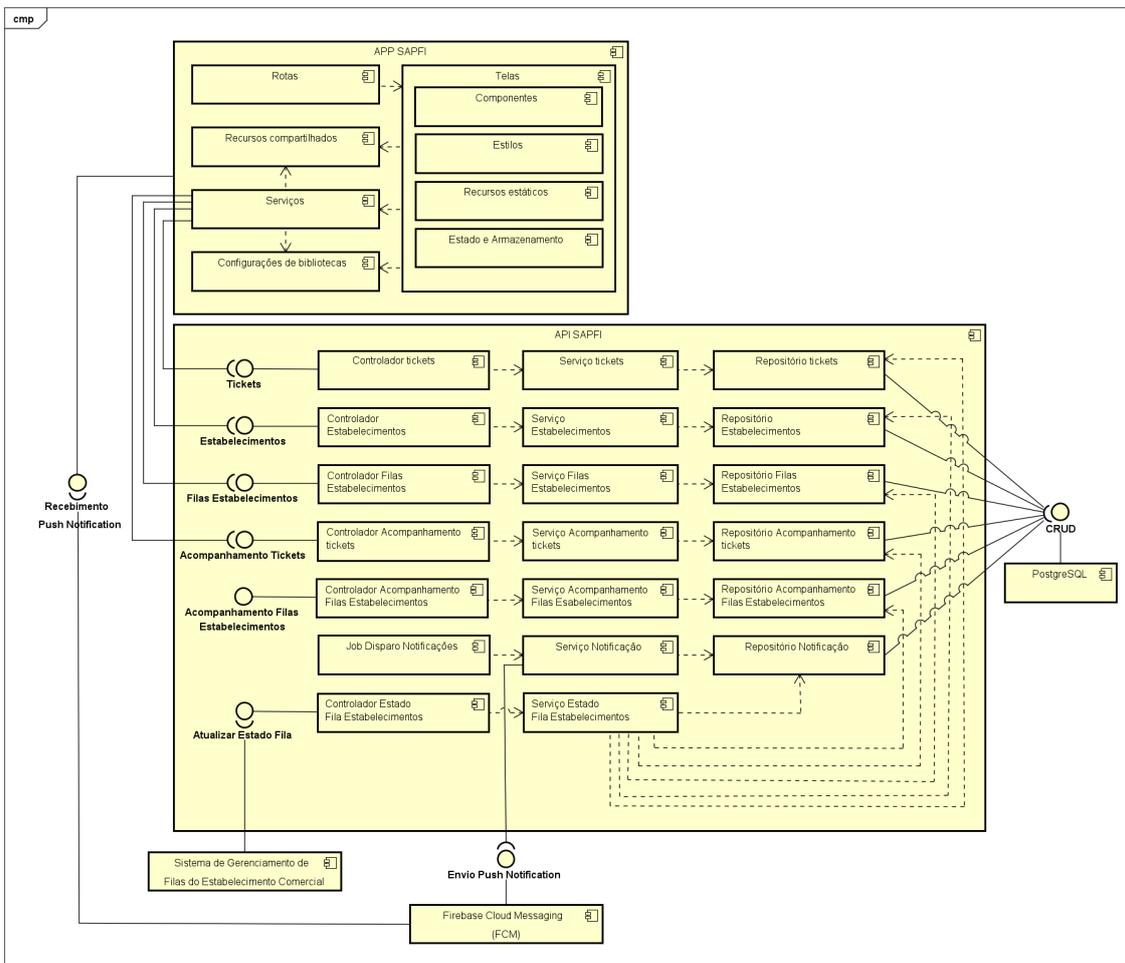
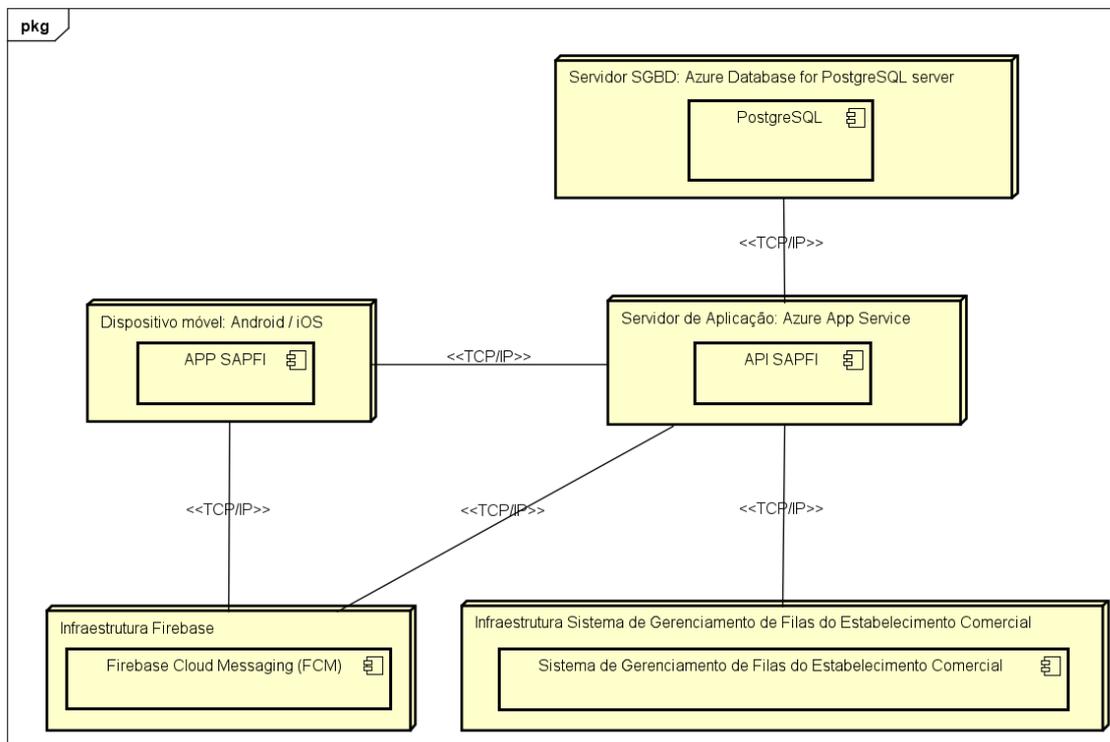


Figura 1. Diagrama de Caso de Uso - SAPFI



**Figura 2. Diagrama de Componentes - SAPFI**



**Figura 3. Diagrama de Implantação - SAPFI**

### 5.2.3. Diagrama de Entidade e Relacionamento

O diagrama de entidade e relacionamento, conforme exibido na Figura 4, descreve as tabelas, colunas e relacionamentos dentro do banco de dados do sistema SAPFI. A tabela *ticket* armazena todas as *tickets* de atendimento de um determinado estabelecimento comercial, representado pela tabela *company*. Os registros da tabela *ticket* são atualizados constantemente até que a entrega do pedido seja finalizada. As principais informações atualizadas nessa tabela são: a posição na fila (*line\_position*), tempo estimado para entrega (*waiting\_time*) e quando o pedido estiver pronto para retirada (*called\_at*). Essa tabela também permite retornar o histórico de pedidos que já foram atendidos, visto que existe a informação de quando o pedido foi identificado como pronto para retirada. O atributo *external\_id* representa a identificação do *ticket* no sistema de gerenciamento de filas do estabelecimento comercial. Esse atributo é usado como referência para que o sistema SAPFI possa atualizar adequadamente as informações. Um registro é criado na tabela *ticket\_follow\_up* quando o usuário inicia o acompanhamento de um determinado *ticket*. Essa tabela armazena o token do dispositivo do usuário (*device\_token*), que é usado no disparo da notificação direcionada que alerta quando o pedido estiver próximo ou pronto para retirada.

A tabela *line* armazena as informações da situação atual da fila de um determinado estabelecimento. A visão dos dados nesse contexto é macro, visto que ali estão armazenadas a quantidade total de *tickets* para serem atendidos (*number\_of\_tickets*) e o tempo total de espera para atendimento (*waiting\_time*). Um registro é inserido na tabela *line\_follow\_up* quando cria um alerta para a situação da fila de um estabelecimento. Ao

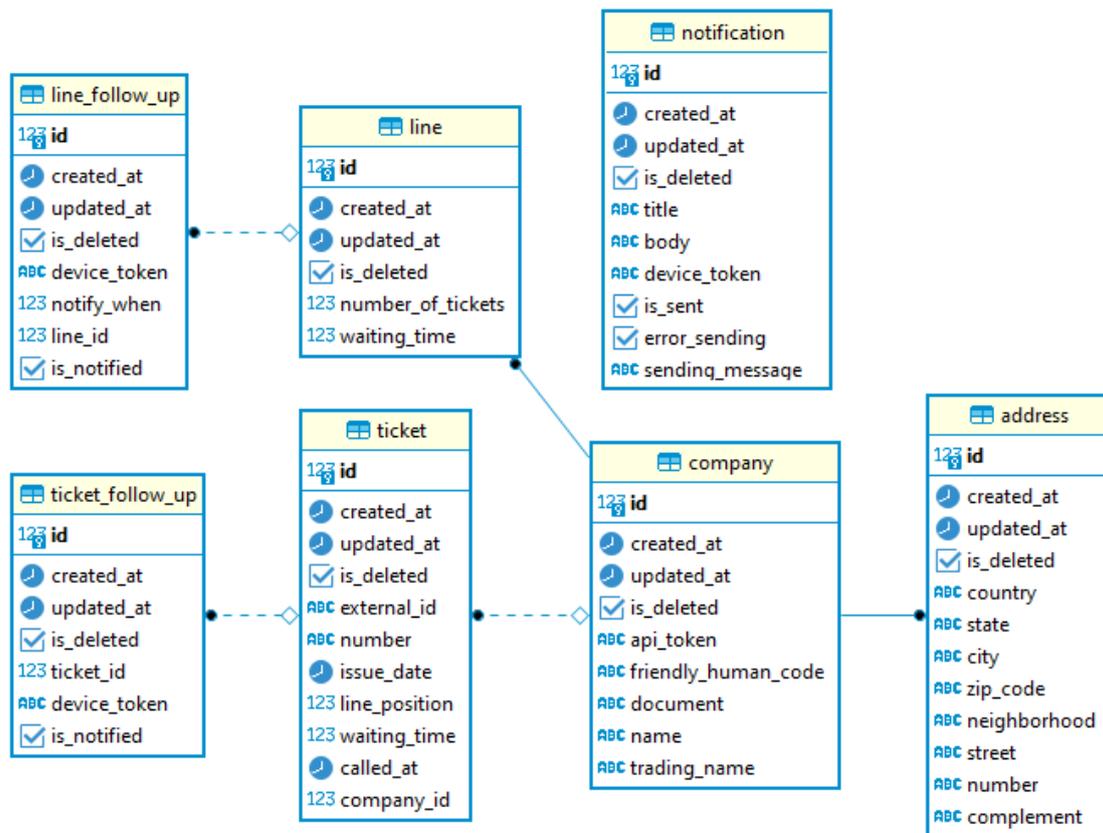


Figura 4. Diagrama de Entidade e Relacionamento - SAPFI

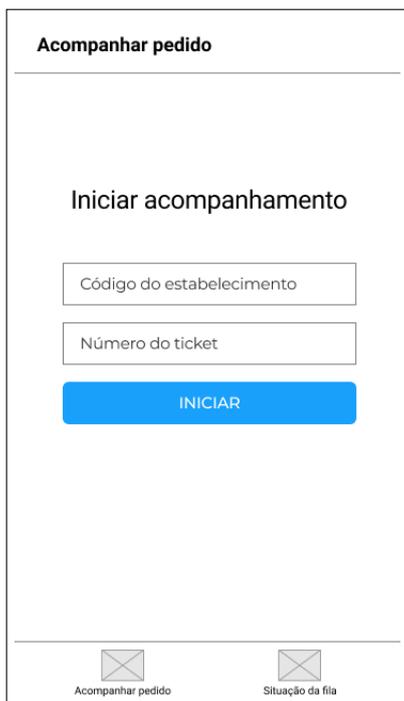
criar este alerta o usuário especifica quando deseja ser notificado (*notify\_when*). Também é persistido nessa tabela o token do dispositivo do usuário.

A medida em que o sistema realiza os processamentos, registros de notificações podem ser criados na tabela *notification*. Essa tabela armazena qualquer tipo de notificação, seja ela para o alerta de atendimento de um pedido ou da situação da fila de um estabelecimento. Uma vez que a mensagem é enviada, uma *flag* é atribuída como *true* (*is\_sent*) para evitar o reenvio.

#### 5.2.4. Wireframes

Os *wireframes* desenvolvidos refletem a prototipação das telas principais do sistema, que são: iniciar acompanhamento de pedido, acompanhamento de informações do pedido, filtro de estabelecimentos próximos, listagem de estabelecimentos próximos, acompanhamento de informações de filas e cadastro de alerta de quantidade de pessoas na fila.

Na Figura 5 é representada a tela de início e também de entrada de dados para iniciar um acompanhamento de pedido. Nela o usuário insere o código do estabelecimento (uma sequência de 4 letras que serve para identificar de forma fácil cada estabelecimento) e o número do *ticket*. Com as informações válidas e ao clicar em iniciar, o sistema redireciona o usuário para a próxima tela, representada pela Figura 6, onde são exibidas as



**Figura 5. Wireframe Iniciar acompanhamento de pedido - SAPFI**



**Figura 6. Wireframe Acompanhamento de Informações do Pedido - SAPFI**

informações referentes ao pedido do usuário, contendo seu número de *ticket*, sua posição na fila, o tempo estimado de espera e as últimas senhas chamadas.

Na Figura 7 é representada a tela de filtragem de estabelecimentos próximos para acompanhamento da situação da fila dos estabelecimentos cadastrados. Nela o usuário seleciona o estado e a cidade em que se encontra e ao clicar em filtrar é redirecionado para a tela de listagem de estabelecimentos próximos, representada pela Figura 8. Nesta tela o usuário encontra informações como nome e endereço de todos os estabelecimentos cadastrados condizentes com a região por ele escolhida.

Ao selecionar um estabelecimento, o usuário é redirecionado para a tela de acompanhamento de informações referente à fila do estabelecimento, representada pela Figura 9. Nesta tela o usuário tem acesso a informações em tempo real, como a quantidade de pessoas na fila e a estimativa para atendimento, além de um botão para criar alerta que, ao ser clicado, exibe um *Pop-up* como o representado na Figura 10, onde o usuário insere a quantidade de pessoas na fila que deseja e, ao clicar em confirmar, um alerta é salvo com os dados informados.

### 5.3. Sistema Desenvolvido

O sistema foi considerado como finalizado a partir do momento que todos os casos de uso foram implementados, testados e todos os ajustes e correções de *bugs* foram devidamente implementados e estão disponíveis na *branch* *develop* do versionamento GIT. Posteriormente, seguindo o fluxo do Git Flow, as *branches* *release* da API e do aplicativo móvel SAPFI foram criadas, validadas e integradas na *branch* principal (*main*) dos repositórios. A publicação da API seguiu o fluxo da *pipeline* configurada e foi publicada na Microsoft



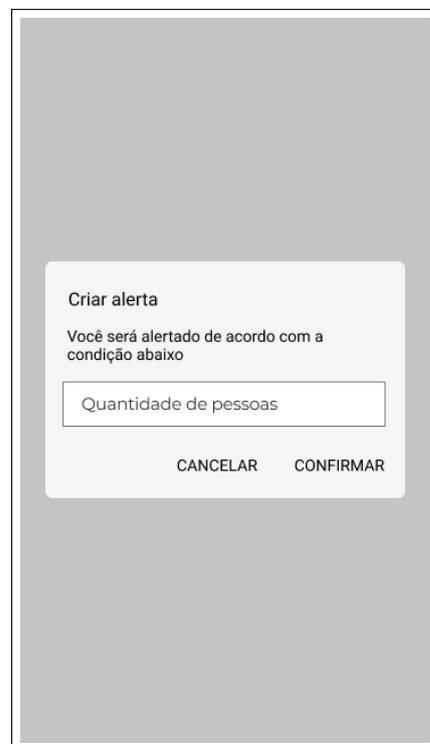
**Figura 7. Wireframe Filtro de Estabelecimentos Próximos - SAPFI**



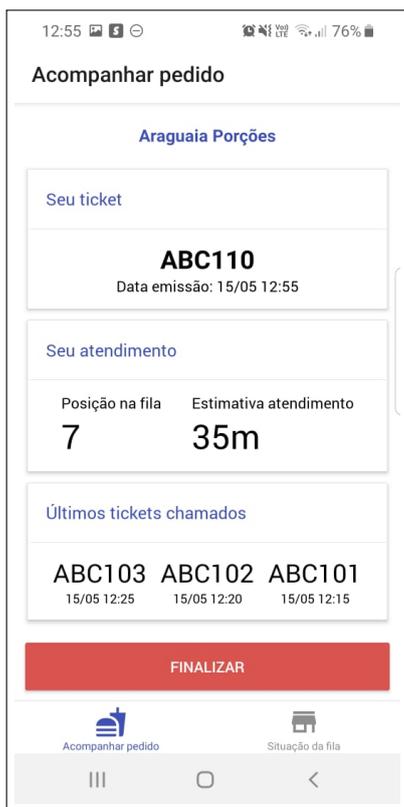
**Figura 8. Wireframe Listagem de Estabelecimentos Próximos - SAPFI**



**Figura 9. Wireframe Acompanhamento de informações referentes à fila - SAPFI**



**Figura 10. Wireframe Cadastro de alerta de quantidade de pessoas na fila - SAPFI**



**Figura 11. Tela do acompanhamento do pedido do aplicativo SAPFI, instalado em um dispositivo físico com Android 9 Pie**



**Figura 12. Tela da situação da fila de um estabelecimento do aplicativo SAPFI, instalado em um dispositivo físico com Android 9 Pie**

Azure.

No caso do aplicativo móvel, o *build* foi realizado e publicado na Google Play<sup>20</sup> (loja de aplicativos oficial do sistema operacional Android). As Figuras 11 e 12 exibem telas do aplicativo SAPFI instalado em um dispositivo físico. O vídeo de apresentação do sistema na perspectiva do usuário foi disponibilizado no YouTube<sup>21</sup>. A versão final do sistema está disponível no GitHub, sendo a versão para dispositivos móveis no repositório front-end-tcc-ii<sup>22</sup> e a versão que executa no servidor no repositório back-end-tcc-ii<sup>23</sup>.

#### 5.4. Resultado da Avaliação de Usabilidade

Em um primeiro momento, foi realizado um teste piloto para a avaliação da usabilidade do sistema. O sistema SAPFI depende da integração com os sistemas dos estabelecimentos comerciais para o seu funcionamento. Neste contexto, o piloto foi aplicado com a

<sup>20</sup>SAPFI na Google Play - Acesso disponível em: <https://play.google.com/store/apps/details?id=br.pucminas.tcc.sapfi>. Último acesso em 14 de maio de 2021

<sup>21</sup>Vídeo de apresentação do SAPFI no YouTube - Acesso disponível em: <https://youtu.be/BhpY8whw5XA>. Último acesso em 14 de maio de 2021

<sup>22</sup>Repositório aplicativo móvel SAPFI no GitHub - Acesso disponível em: <https://github.com/lucashmalcantara/front-end-tcc-ii>. Último acesso em 27 de maio de 2021

<sup>23</sup>Repositório API SAPFI no GitHub - Acesso disponível em: <https://github.com/lucashmalcantara/back-end-tcc-ii>. Último acesso em 27 de maio de 2021

finalidade de se entender qual seria a melhor dinâmica para propiciar um ambiente simulado que permitisse ao usuário compreender a proposta do sistema. Dados fictícios de 3 estabelecimentos foram inseridos no sistema para proporcionar o uso. Os dados das filas e dos *tickets* foram atualizados periodicamente, simulando o fluxo de atendimento dos estabelecimentos disponíveis. No total, 5 participantes usaram o sistema e responderam o questionário SUS, desenvolvido através da ferramenta Google Forms<sup>24</sup>. O *score* obtido foi mais alto do que o esperado, o que levou a uma reformulação do processo de forma a garantir a liberdade de expressão do usuário para proporcionar a imparcialidade dos resultados.

Para a segunda aplicação do teste de usabilidade, identificou-se a necessidade de desenvolver um módulo simulador na API que proporcionasse um ambiente mais fiel à realidade. Esse módulo foi desenvolvido como um serviço hospedado (derivado da interface *IHostedService* do .NET Core), que é executado em *background* periodicamente atualizando o estado da fila dos estabelecimentos fictícios disponíveis. Os usuários puderam baixar o aplicativo diretamente da Google Play e, assim, realizar os testes.

Foram feitos ajustes no questionário SUS<sup>25</sup>, proporcionando ao usuário maior privacidade e segurança ao responder as perguntas, para excluir qualquer elemento que pudesse influenciar as respostas. Também foi inserida na simulação mais uma empresa fictícia, totalizando 4 restaurantes. O questionário foi aplicado a sete novas pessoas consideradas como público-alvo e usuários finais do aplicativo.

A Figura 13 apresenta um gráfico composto pelo *score* individual de cada questão do questionário e o *score* final da escala SUS. O *score* individual das questões no gráfico pode ser analisado da seguinte forma: para questões ímpares, o *score* indica o quanto os usuários concordam com a afirmação; para questões pares, o *score* indica o quanto os usuários discordam da afirmação. O resultado final obtido gerou um *score* de 76,07 na escala SUS, que enquadra o sistema como *aceitável* [Bangor et al. 2009].

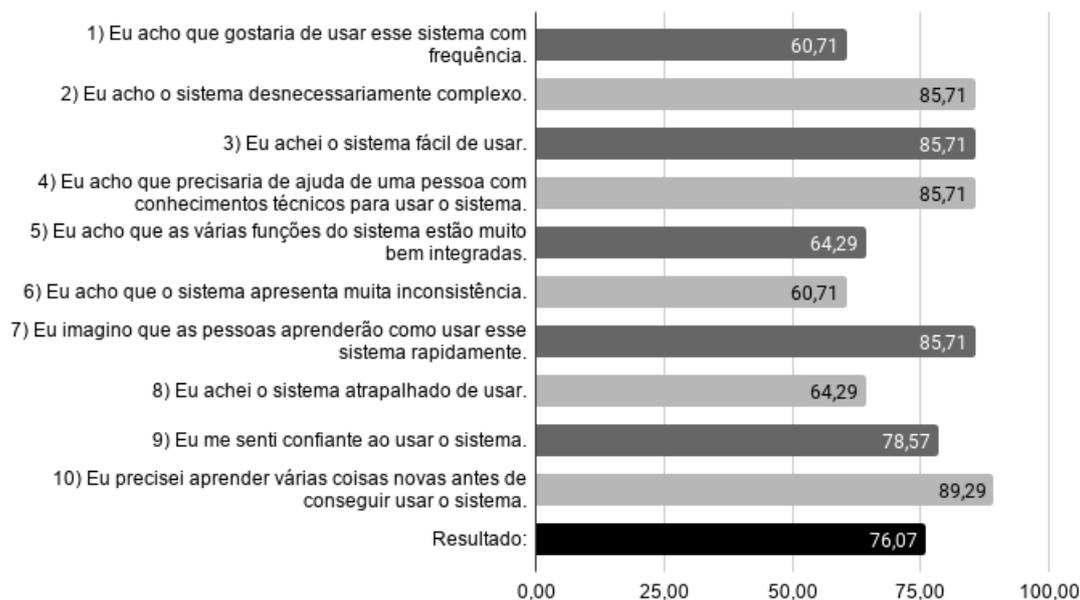
Analisando os resultados obtidos para cada questão, observa-se que a Questão 10, que está relacionada com a necessidade de aprender coisas novas para usar o sistema, obteve o maior *score*: 89,29. Nesta questão, o *score* obtido indica o quanto os usuários discordam da afirmação e pode ser justificado pelas seguintes características do sistema: uso de linguagem ubíqua, textos curtos, telas minimalistas e presença de *affordance* nos elementos da interface.

Ainda analisando os resultados por questão, pode-se destacar as de número 1 e 6 como sendo as que mais impactaram negativamente no resultado, ambas obtendo o *score* de 60,71. A Questão 1 refere-se à frequência de uso do sistema, cujo *score* pode ser justificado pela presença de funcionalidades que podem ser utilizadas em ocasiões específicas, como a vontade do usuário em ir a um restaurante. Entende-se que novas funcionalidades devem ser agregadas para garantir o aumento na frequência de uso do sistema. A Questão 6 refere-se a inconsistências identificadas no uso do aplicativo. Nesta questão, o *score* obtido indica o quanto os usuários discordam da afirmação e um aspecto

---

<sup>24</sup>Google Forms - Acesso disponível em: <https://www.google.com/intl/pt-BR/forms/about/>. Último acesso em 15 de maio de 2021

<sup>25</sup>Questionário do aplicativo SAPFI - Acesso disponível em: <https://forms.gle/xaqp7aVacsVH66ff7>. Último acesso em 15 de maio de 2021



**Figura 13. Gráfico dos resultados obtidos por questão na aplicação do questionário SUS**

que pode ter influenciado esse resultado é o uso do sistema em um ambiente simulado, que pode não ter qualificado uma experiência completa de uso como em um ambiente real.

## 6. Conclusão

Este trabalho apresentou o desenvolvimento e teste de usabilidade de um sistema de alerta de posição em filas denominado SAPFI, composto por dois módulos principais: a API e o aplicativo móvel. Os resultados apresentados pelo teste de usabilidade indicam que as funcionalidades disponíveis pelo sistema são úteis e funcionam corretamente. A forma com que o desenvolvimento foi conduzido, bem como a definição adequada dos requisitos e as modelagens do sistema, foram fatores importantes para o sucesso da implementação do trabalho.

Adicionalmente, após a conclusão do desenvolvimento do sistema, foi feita a aplicação do questionário SUS para avaliar os resultados obtidos perante o uso por usuários reais em um ambiente simulado. O questionário gerou um *score* de 76,07 na escala SUS, que classifica o sistema como útil e aceitável para o propósito pelo qual foi desenvolvido. Observou-se alguns pontos de melhoria, como a inclusão de funcionalidades que permitissem o uso mais constante do sistema e ajustes de inconsistências.

A principal contribuição apresentada por este trabalho foi a proposta e implementação de um aplicativo que tem como objetivo informar e alertar sobre a situação das filas de restaurantes de praças de alimentação, permitindo aos usuários evitar aglomerações, manter o distanciamento social, economizar tempo e aguardar a preparação de seus pedidos com maior comodidade.

Como trabalhos futuros, seria relevante realizar testes de usabilidade com mais

usuários, levantando mais opiniões acerca do funcionamento do sistema e dos pontos de melhoria. Também pretende-se realizar a integração do sistema SAPFI com sistemas de gerenciamento de filas de estabelecimentos reais. Além disso, mais funcionalidades podem ser adicionadas, como: i) mecanismo de busca de estabelecimentos próximos baseado em geolocalização; ii) melhoria no processo de inserção dos dados do *ticket* para acompanhamento como, por exemplo, por leitura de QR Code; iii) permitir mais flexibilidade na parametrização dos alertas; iv) permitir o acompanhamento de múltiplos *tickets* ao mesmo tempo. Também é cogitada a realização de um estudo para analisar a possibilidade de expansão do sistema para outros segmentos de mercado.

## Referências

- Alsaeed, W. and Alhazmi, K. (2019). An intelligent spatial-based queue management system. In *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pages 1–7. IEEE.
- Aquino, E. M., Silveira, I. H., Pescarini, J. M., Aquino, R., and Souza-Filho, J. A. d. (2020). Medidas de distanciamento social no controle da pandemia de covid-19: potenciais impactos e desafios no brasil. *Ciência & Saúde Coletiva*, 25:2423–2446.
- Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123.
- Boucinha, R. M. and Tarouco, L. M. R. (2013). Avaliação de ambiente virtual de aprendizagem com o uso do sus-system usability scale. *RENOTE-Revista Novas Tecnologias na Educação*, 11(3).
- Cruz, F. (2013). *Scrum e PMBOK unidos no Gerenciamento de Projetos*. Brasport.
- dos Santos Oliveira, W. C., Costa, S. S., dos Santos, D. V., de Sousa Lima, T. M., and da Silva, M. L. (2017). Fila virtual: Um modelo inovador para filas de restaurantes universitários. *VI Encontro Acadêmico de Computação - UFMA*.
- Eckschmidt, F. et al. (2019). Interface gráfica para criação de modelos e ferramentas de gestão visual de projetos. PCC (graduação) - Universidade Federal de Santa Catarina. Centro de Comunicação e Expressão. Design.
- Geraldes, W. B., Martins, E. R., and Afonseca, U. R. (2019). Avaliação da usabilidade do scratch utilizando o método system usability scale (sus). In *Anais da X Escola Regional de Informática de Mato Grosso*, pages 25–30. SBC.
- Ghazal, M., Hamouda, R., and Ali, S. (2016). A smart mobile system for the real-time tracking and management of service queues. *International Journal of Computing and Digital Systems*, 5(04).
- Iglesias, F. (2007). *Comportamentos em filas de espera: Uma abordagem multimétodos*. PhD thesis, Universidade de Brasília.
- Iglesias, F. and Günther, H. (2009). A espera na vida urbana: uma análise psicossocial das filas. *Psicologia em estudo*, 14(3):537–545.
- Leite, G. d. F. et al. (2016). Estudo da eficiência energética em sistemas de informação: focado em um sistema de gerenciamento de filas. TCC (graduação) - Universidade Federal de Santa Catarina. Campus Araranguá. Engenharia de Energia.

- Lewis, J. R. and Sauro, J. (2018). Item benchmarks for the system usability scale. *Journal of Usability Studies*, 13(3).
- Poseiro, P., Reis, M., Fortes, C., Sabino, A., and Rodrigues, A. (2014). Aplicação do sistema de previsão e alerta de inundações hidralerta: Caso de estudo da costa da caparica. *Proc. 3as Jornadas de Engenharia Hidrográfica, Lisboa, Portugal*.
- Pradini, R. S., Kriswibowo, R., and Ramdani, F. (2019). Usability evaluation on the sipr website uses the system usability scale and net promoter score. In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 280–284.
- Pressman, R. and Maxim, B. (2016). *Engenharia de Software-8ª Edição*. McGraw Hill Brasil.
- Rios, E., Matos, I., and Lima, P. (2018). Relato de experiência projeto de extensão: Curso ux e ui design. In *Anais do XVIII Escola Regional de Computação Bahia, Alagoas e Sergipe*, pages 535–543, Porto Alegre, RS, Brasil. SBC.