



DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA GESTÃO DE PEDIDOS DO BANCO ALIMENTARIO DE LA PLATA

ARTIGO ORIGINAL

SOUZA, Arllen Demétrio Mota de¹, SILVA, Evaldo de Oliveira da²

SOUZA, Arllen Demétrio Mota de. SILVA, Evaldo de Oliveira da. **Desenvolvimento de um aplicativo móvel para gestão de pedidos do banco alimentario de La Plata.**

Revista Científica Multidisciplinar Núcleo do Conhecimento. Ano. 07, Ed. 01, Vol. 03, pp. 103-125. Janeiro de 2022. ISSN: 2448-0959, Link de acesso:

[https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/aplicativo-](https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/aplicativo-movel)

[movel](https://www.nucleodoconhecimento.com.br/engenharia-da-computacao/aplicativo-movel), DOI: 10.32749/nucleodoconhecimento.com.br/engenharia-da-computacao/aplicativo-movel

RESUMO

Este trabalho descreve o desenvolvimento e as funcionalidades de utilização de uma aplicação móvel desenvolvida na plataforma Android para gerenciar os pedidos de alimentos doados pela ONG Banco Alimentario de La Plata. O Banco Alimentario de La Plata vem passando por problemas como: falta de voluntários para atender pedidos por telefone, acúmulo de filas, alto custo manual no processo de retirada de alimentos para os beneficiários e um controle de estoque ruim com acúmulo de papel. Esses pontos tornam difícil a gestão de distribuição de alimentos, sendo esta a problemática explorada no decorrer do presente artigo. Portanto, a aplicação tem a finalidade de reduzir tempo na realização dos pedidos, reduzir o número de ligações telefônicas feitas pelas instituições e beneficiários e visa minimizar a necessidade dos beneficiados de se deslocarem pessoalmente para realizarem os pedidos de alimentos. Dessa forma, o objetivo deste trabalho é apresentar a implementação de uma aplicação de *software* para ser acessada por dispositivos móveis, a fim de automatizar a gestão de pedidos do Banco Alimentario de La Plata. Finalmente, espera-se, com o uso desta aplicação móvel, que sejam melhorados o tempo e o custo do processo de distribuição de alimentos, agilizando o processo interno de distribuição e logística.

Palavras-chave: Desenvolvimento Mobile, Android, Desenvolvimento de Aplicativos, Aplicativo Android, Aplicativo Móvel.



1. INTRODUÇÃO

O objetivo deste trabalho é apresentar a implementação de uma aplicação de *software* para ser acessada por dispositivos móveis, a fim de automatizar a gestão de pedidos do Banco Alimentario de La Plata (BANCO ALIMENTARIO, 2019), uma ONG (Organização Não Governamental) situada na cidade administrativa de La Plata, Argentina.

O processo de empacotamento dos pedidos no Banco Alimentario é um processo difícil, pois são muitas as instituições que fazem pedidos por telefone ou pessoalmente no Banco Alimentario. Isso ocasiona a perda de tempo em excesso nesse processo.

Com a alta demanda diária por alimentos, visto que o BALP (Banco Alimentario de La Plata) abastece todas as comunidades da região La Plata, a escassez de voluntários existentes na ONG ocasiona alguns problemas, como atraso, acúmulo de filas, sobrecarga de voluntários já lotados na ONG e um controle de estoque indesejável com acúmulo de papéis.

Por meio de um acordo de cooperação firmado entre o Banco Alimentario de La Plata (BALP) e o Centro de Ensino Superior de Juiz de Fora (CESJF), possibilitou-se o contato entre ambas as entidades, para efeito de levantamento de requisitos, assim como o uso da infraestrutura de banco de dados para a aplicação móvel apresentada neste trabalho.

Até a implementação recente do aplicativo móvel proposto, o Banco Alimentario de La Plata possuía um tipo de processo não automatizado para atender os beneficiados na solicitação de pedidos de alimentos. Neste cenário, formam-se muitas filas na sede do Banco Alimentario de La Plata, provocando sobrecarga de demanda, e, portanto, sobrecarga de trabalho. Com isso, surgiu o impasse de que os voluntários se alocassem em outro setor, a fim de compor o grupo de funcionários, em prol do atendimento às instituições beneficiárias.



Nesse contexto, o aplicativo móvel em questão, também chamado de Banco Alimentario, é caracterizado pelo amplo número de objetivos específicos que canaliza, como a otimização do tempo de realização dos pedidos ou processamento de entrega dos alimentos. Além disso, ele, o aplicativo, possibilita que se realoque o voluntário responsável por atender o telefone em outra área. Com isso, as instituições e beneficiados destas que utilizam os serviços da ONG têm seu tempo otimizado, visto que estes, os beneficiados, não precisarão mais ir à instituição para fazer o pedido, tendo que voltar, depois de dois dias, para buscar os alimentos. Assim, ele poderá fazer o pedido da aplicação em qualquer localização.

Todavia, os funcionários e/ou voluntários poderão aplicar seus esforços em outros tipos de tarefas, já que o tempo outrora utilizado para realizar os pedidos dos comedores poderá ser direcionado, agora, para outra área. Ademais, atualmente existem cerca de 400 Instituições beneficiadas pelo Banco Alimentario de La Plata, e espera-se que o aplicativo móvel, de variadas aplicações, permita proporcionar maior mobilidade para um conjunto grande de pessoas.

Nesse sentido, foram especificadas duas aplicações. A primeira aplicação foi desenvolvida para a instituição beneficiária. Já a segunda tem o objetivo de oferecer funções administrativas, na qual o gestor responsável do Banco Alimentario de La Plata irá gerenciar os pedidos realizados pelas instituições.

Para o desenvolvimento do aplicativo, valeu-se de leituras de artigos científicos, além de entrevistas para levantamento de requisitos com os profissionais das áreas de logística e administrativas do Banco Alimentario de La Plata. Também foi efetuado um levantamento dos aplicativos com funcionalidades parecidas com as que deveriam constar na solução do projeto, na tentativa de encontrar funcionalidades similares.

Como soluções para o desenvolvimento do aplicativo, utilizou-se o *Framework Kodular* e o ambiente de desenvolvimento Android, visando minimizar o tempo para desenvolvimento da solução no tempo estabelecido entre fevereiro e junho de 2019.



O restante do texto se encontra organizado pelas seções descritas a seguir. A seção 2 corresponde à metodologia. A Seção 3 (três) é dedicada ao referencial teórico, à definição dos conceitos e técnicas utilizadas e descreve os requisitos especificados pela ONG. A Seção 4 (quatro) se relaciona com a implementação dos requisitos levantados e informados pelo Banco Alimentario de La Plata. Finalmente, a Seção 5 (cinco) descreve as considerações finais e os trabalhos futuros.

2. METODOLOGIA

Inicialmente, foi realizado um levantamento de dados para avaliar a complexidade e o tempo disponível para a realização do projeto. Ainda, usou-se uma plataforma de desenvolvimento *low-code* (plataforma visual de desenvolvimento de *software*), tendo em vista a velocidade que ela propõe para o desenvolvimento da aplicação. Levando em consideração o problema na gestão de pedidos de alimentos sem a ajuda de ferramentas informatizadas, a proposta do projeto foi minimizar os impactos que estavam sendo causados nesta gerência de distribuição de pedidos manuais.

O Android foi o sistema operacional escolhido para o aplicativo, com base no custo para publicação em loja. Tal opção foi seguida devido ao valor dos smartphones serem mais baratos. Por fim, também foi realizado um estudo interno com entrevistas, as quais evidenciaram que 98% dos beneficiários utilizam smartphones com sistema operacional Android.

3. REFERENCIAL TEÓRICO

Esta seção descreve os conceitos indispensáveis para a contextualização teórica deste projeto. Os conceitos abordados nesta seção dizem respeito à plataforma de desenvolvimento Kodular, o ambiente Android, a ONG e os requisitos para criação da aplicação.



3.1 BANCO ALIMENTARIO DE LA PLATA

O Banco Alimentario de La Plata surgiu em meados de 2000, com o objetivo reduzir a fome, a desnutrição e as más práticas alimentares na região. Para solucionar isso, usaram práticas como a recuperação de alimentos, a fim de serem distribuídos às organizações comunitárias que prestam serviços de alimentação a setores carentes, os quais desenvolvem ações conjuntas com a sociedade, baseadas em valores e capacidades conjuntas. Assim, um pilar muito importante do Banco Alimentario de La Plata é evitar o desperdício de alimento (BANCO ALIMENTARIO, 2019).

Desde então, a associação possui seu armazém e repartições na propriedade, doada pela Direção de Estradas da Província de Buenos Aires.

Diariamente se recuperam alimentos que podem ser consumidos, porém, não podem ser comercializados em empresas particulares por conta de um rótulo ou problema no pacote defeituoso no processo de fabricação, ou pelo motivo de estar próximo do vencimento. Depois que os produtos são levados ao depósito do Banco Alimentario, eles são separados pelos voluntários e classificados por data de vencimento e tipo do produto (BANCO ALIMENTARIO, 2019).

E, por último, após ser feito todo esse processo interno com os produtos, concede-se a entrega dos produtos, em que mais de 100 instituições beneficiárias vem ao Banco para retirar os alimentos, fazendo uma contribuição simbólica por quilo de alimento. Essas instituições, também chamadas de comedores, são, por exemplo, igrejas, abrigos, bancos de leite e lares (BANCO ALIMENTARIO, 2019).

3.2 DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

Nos últimos anos, a indústria de desenvolvimento de aplicativos móveis apresentou um crescimento agudo. Com o crescente número de pessoas acessando a internet por meio de seus smartphones, tornou-se crucial para os empresários, agora, ter um aplicativo móvel para maior envolvimento dos consumidores (DEVMEDIA, 2019).



O mundo contemporâneo, tecnológico e técnico, é um mundo que prioriza os dispositivos móveis. Com isso, os telefones celulares são uma prioridade crucial para as organizações envolverem, atraírem e encantarem seus usuários.

Com o desenvolvimento do *software* de PC de mesa, os programadores precisaram criar um aplicativo que pudesse operar com um conjunto mínimo de *hardware*. Isso é verídico, também, para a aplicativos móveis, embora as variações de *hardware*, nesse caso, sejam bem menores. Ao mesmo tempo, o hardware em *smartphones* e *tablets* não está, de forma alguma, próximo de *laptops* e PC's, o que significou que os aplicativos móveis precisaram ser projetados para mostrar um ótimo desempenho (RIBEIRO, 2015).

A resolução do problema de desempenho em um determinado dispositivo depende, em última análise, do desenvolvimento de um aplicativo nativo nesse dispositivo. Isso significa projetar o código especificamente para o *hardware* em um dispositivo específico. No caso de dispositivos iOS, tal ação se torna fácil, pois os desenvolvedores de dispositivos móveis só precisam de versões do aplicativo para iPhone e iPad, a fim de alcançar a usabilidade universal. Quanto aos dispositivos Android, no entanto, todos os smartphones e tablets se executam em diferentes *hardwares* e diferentes versões do sistema operacional (DEVMEDIA, 2019).

3.3 AMBIENTE ANDROID

O Android Inc. foi fundado por Andy Rubin, Rich Miner, Nick Sears e Chris White em Palo Alto, Califórnia, em outubro de 2003. Rubin descreveu o projeto Android como "um tremendo potencial no desenvolvimento de dispositivos móveis mais inteligentes e mais conscientes da localização do proprietário" (ANDROID, 2019, n.p.).

O principal objetivo da empresa era desenvolver um sistema operacional avançado para câmeras digitais. Tal objetivo propiciou a injeção econômica de investidores em abril de 2004. A partir disso, a empresa decidiu, então, que o mercado de câmeras não era grande o suficiente para seus objetivos e, 5 (cinco) meses depois, mudou seu



foco de ação, lançando o Android como um sistema operacional de celulares, o qual se oporia ao Symbian e ao Microsoft Windows Mobile.

O ambiente Android tem disponível várias opções de bibliotecas e *frameworks*. Estes ajudam no desdobramento de aplicações mobile e na interatividade com os dispositivos de celular, como, por exemplo, opções de conectividade, redes sem fio como WiFi e Bluetooth; e estabelece interoperabilidade com os serviços oferecidos pelo Google, como Google Maps, Google Drive e Gmail. A grande aposta do Android são os novos aparelhos celulares, mais conhecidos como smartphones, isto é, celulares com grande poder de processamento que integram vários recursos, como alta conectividade com a Internet, GPS (*Global Positioning System*), sensores, telas sensíveis ao toque etc. (OGLIARI e BRITO, 2014, n.p.).

3.4 FRAMEWORK KODULAR

O *Framework Kodular* se constitui como um ambiente de programação visual e intuitivo totalmente gratuito, que permite que todos – até mesmo crianças – criem aplicativos funcionais para smartphones e tablets. Tal *framework*, baseado em blocos, facilita a criação de aplicativos complexos e de alto impacto em um tempo significativamente menor do que os ambientes de programação tradicionais.

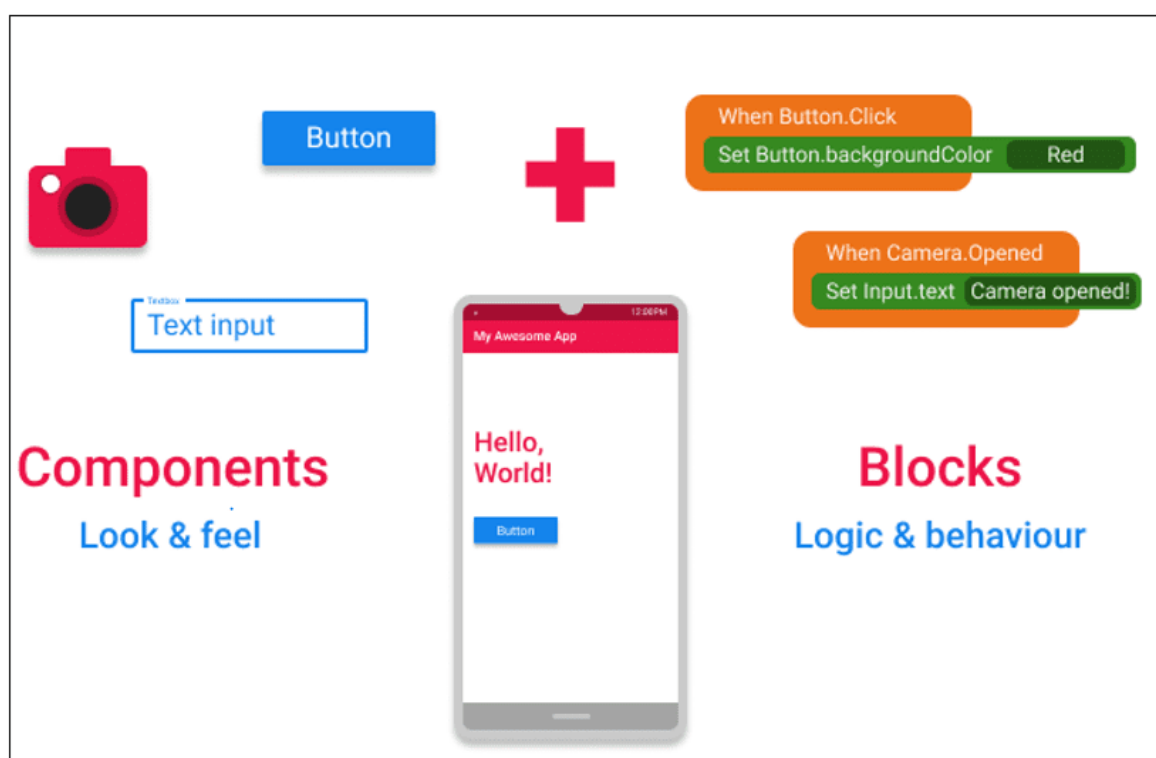
O Kodular apresentado na Figura 1 possui integração direta com diversos bancos de dados, inclusive os resididos em nuvem, facilitando, assim, a conexão e troca de informações de forma eficaz. A maneira em que isso é feito responde a várias ações e eventos no aplicativo, sendo projetado a partir dos blocos de comando (KODULAR, 2019).

A integração dos bancos de dados local (*Tiny DB*) com a nuvem (*Database Real-Time*) permite que as informações sejam trocadas ao longo dos procedimentos requisitados no uso da aplicação. Ademais, a forma como esse ambiente de desenvolvimento fornece esses recursos, facilita bastante a execução do operador, visto que, em poucos passos, poucos blocos de comando permitem a integração de dois grandes

bancos de dados. A maneira como é tratada o *front-end* da aplicação é, também, bastante fácil e intuitiva, sendo parecida com a do Android-Studio.

O projeto MIT (*Massachusetts Institute of Technology*) visa democratizar o desenvolvimento de *software*, capacitando todas as pessoas, especialmente os jovens, para passar do consumo de tecnologia para a criação de tecnologia (APP INVENTOR, 2019).

Figura 1. Componentes e Blocos



Fonte: Kodular, 2019

3.5 FIREBASE DATABASE-REALTIME

O *Firebase Realtime Database* é executado em tempo real, o que significa que, se um usuário do aplicativo atualizar os dados, os dados no *Google Server* serão atualizados imediatamente. Assim, atualizando o *Firebase*, consegue-se, imediatamente, que todos os dados de todos os outros usuários que estiverem usando o aplicativo também



se renovem. Vale a pena ressaltar, não obstante, que os dados não constam em tabelas, mas, apenas, na forma de documentos JSON.

O *Firebase* disponibiliza o diagnóstico de problemas em seu aplicativo móvel com relatórios detalhados de *bugs* e falhas. Permite, ainda, que o usuário priorize os relatórios por frequência e gravidade do impacto no painel do *Firebase Crash*, onde o usuário pode monitorar a integridade geral do seu aplicativo, bem como acompanhar os fluxos de outros usuários. Essas notificações, quando enviadas, chegam via e-mail ao usufruidor (FIREBASE, 2019).

Como o *DataBase Realtime*, mesmo na versão *free*, fornece um grande espaço de armazenamento, é possível guardar e compartilhar imagens, áudio, vídeo ou outro conteúdo gerado pelo usuário. Seu uso é fácil, uma vez que seu tipo de armazenamento de objeto é funcional, simples e econômico, além de construído para a escala do Google. Os SDKs do *Firebase* para *Cloud Storage* adicionam o elemento de segurança do Google a *uploads* e *downloads* de arquivos para suas execuções dentro do *Firebase*, independente da qualidade da rede (FIREBASE, 2019).

O próprio *Firebase* já possui hospedagem na web, com ferramentas criadas especificamente para aplicativos da web modernos. Quando o usuário faz o *upload* de seus recursos da web, envia-se automaticamente ao CDN global dos operadores, os quais fornecem um certificado SSL gratuito para que os usuários tenham uma experiência segura, confiável e de baixa latência, independentemente de onde estejam (FIREBASE, 2019).

No tocante ao monitoramento e controle, conseguiu-se diagnosticar os problemas de desempenho do aplicativo, os quais ocorrem nos dispositivos de seus usuários, utilizando, para isso, rastreios de monitoração do desempenho de partes específicas do aplicativo, visualizadas em um painel disponibilizado no *Firebase Web* (Plataforma própria do *firebase* para visualização de Dashboard com dados estatísticos coletados no aplicativo desenvolvido) (FIREBASE, 2019).



Esse banco de dados em tempo real é de suma importância para o controle de estoque na aplicação, uma vez que 2 (dois) usuários podem pedir o mesmo produto ao mesmo tempo, com o impasse de haver somente um no estoque. Automaticamente, aquele que pedir por último, mesmo que a diferença de tempo de petição seja por milésimas de segundo, terá uma mensagem em sua tela que lhe dirá que tal produto não se encontra mais em estoque.

3.6 DEFINIÇÃO DOS REQUISITOS PARA CRIAÇÃO DO APLICATIVO

Em uma reunião com os diretores do Banco Alimentario de La Plata foi definido alguns requisitos iniciais da aplicação, constando as necessidades da organização.

Os requisitos funcionais são:

- Requisito Funcional 1: O aplicativo deve conter um menu separado pela categoria dos alimentos;
- Requisito Funcional 2: O aplicativo deve conter uma tela de login, onde o usuário entra com suas credenciais (*login* e senha) para acessar a aplicação;
- Requisito Funcional 3: O aplicativo deve gravar o histórico dos pedidos efetuados pelas instituições, contendo as seguintes informações: data e código de retirada;
- Requisito Funcional 4: O aplicativo deve conter a opção de escolher um produto, de acordo à categoria do mesmo;
- Requisito Funcional 5: O aplicativo deve conter um botão para selecionar a quantidade do produto escolhido. Caso este estiver em falta, o aplicativo o informará ao usuário;
- Requisito Funcional 6: O sistema deve mostrar em sua tela inicial, após selecionado alguns produtos, o subtotal do valor, em pesos argentinos, do que foi colocado no carrinho;
- Requisito Funcional 7: O sistema deve exibir os produtos em lista após selecionado a categoria destes;



- Requisito Funcional 8: O sistema deve conter um botão de confirmação após selecionar os produtos, a fim de que o usuário avance para a próxima tela, onde encontrará a confirmação do que foi selecionado;
- Requisito Funcional 9: O sistema deve exibir, em uma lista, os produtos selecionados pelo usuário. Na tela do sistema, exibir-se-á o carrinho de compras virtuais e o subtotal da compra;
- Requisito Funcional 10: O sistema deve permitir ao usuário fazer a exclusão de qualquer produto que esteja na lista de pedido. Após o usuário excluir um produto, o sistema, automaticamente, subtrai do subtotal o valor daquele produto, para, logo a seguir, exibir uma mensagem.
- Requisito Funcional 11: O sistema deve conter dois campos: nome da instituição e nome do responsável. Este irá ao Banco Alimentario de La Plata para retirar o pedido, com a intenção de preencher aquilo que for requisitado na tela de finalização dos pedidos.
- Requisito Funcional 12: O sistema deve conter, na tela de finalização do pedido, um calendário para o usuário selecionar a data de retirada do alimento, respeitando as seguintes condições:
 - Não se pode fazer retirada segundas-feiras, sextas-feiras, sábados e domingos;
 - A retirada deve ser feita em, no mínimo, dois dias após o pedido.

Os requisitos não funcionais são requisitos relacionados ao uso da aplicação em relação ao desempenho, da usabilidade, confiabilidade, segurança, disponibilidade, manutenção e recursos envolvidos.

- Requisito não funcional 1: O aplicativo deve atender ao requisito de disponibilidade;
- Requisito não funcional 2: O sistema e o banco de dados deverão apresentar suporte para plataforma Android;
- Requisito não funcional 3: Apresentar telas com funções objetivas, evitando, com isso, acúmulo de componentes;



- Requisito não funcional 4: Apresentar uma caixa de diálogo em ações, que permita, quando necessário, informar algo ao cliente.

4. DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA GESTÃO DE PEDIDOS DO BANCO ALIMENTARIO DE LA PLATA

Neste tópico será abordado o funcionamento da aplicação, bem como a maneira em que ela foi montada, tendo como referência a definição dos requisitos abordados no tópico 3 deste documento. Será explicado, também, como se realizou a integração entre os bancos de dados *SQLServer* e *Firebase*.

4.1 ASPECTOS DE LEITURA E CARGA DO BANCO DE DADOS

Esta seção descreverá como foi realizada a integração de dados entre o banco de dados *SQLServer* do Banco Alimentario de La Plata com o banco de dados *Firebase* utilizado para a aplicação móvel. Inicialmente, será feita uma discussão sobre a biblioteca *libcurl*, que permite a execução de certas operações, como o comando *cURL*. Este é utilizado para enviar arquivos, usando, para isso, endereços da Web. Em seguida, na seção 4.2, descrever-se-á as funcionalidades implementadas, usando o *framework Kodular* e o ambiente Android como elementos principais.

O *libcurl* é uma biblioteca de transferência de URL gratuita para o cliente, com suporte para uma ampla variedade de protocolos. O *libcurl* é portátil, *thread-safe*, rico em recursos e com suporte para, praticamente, qualquer plataforma. É provavelmente a mais popular biblioteca de transferência de arquivos baseada em C e multi-plataforma em uso (ORACLE, 2019).

A *cURL* é uma ferramenta de linha de comando para obter ou enviar arquivos, que usa a sintaxe de *URL*. Como o *cURL* usa o *libcurl*, ele suporta o mesmo intervalo de protocolos comuns da Internet do *libcurl* (ORACLE, 2019).

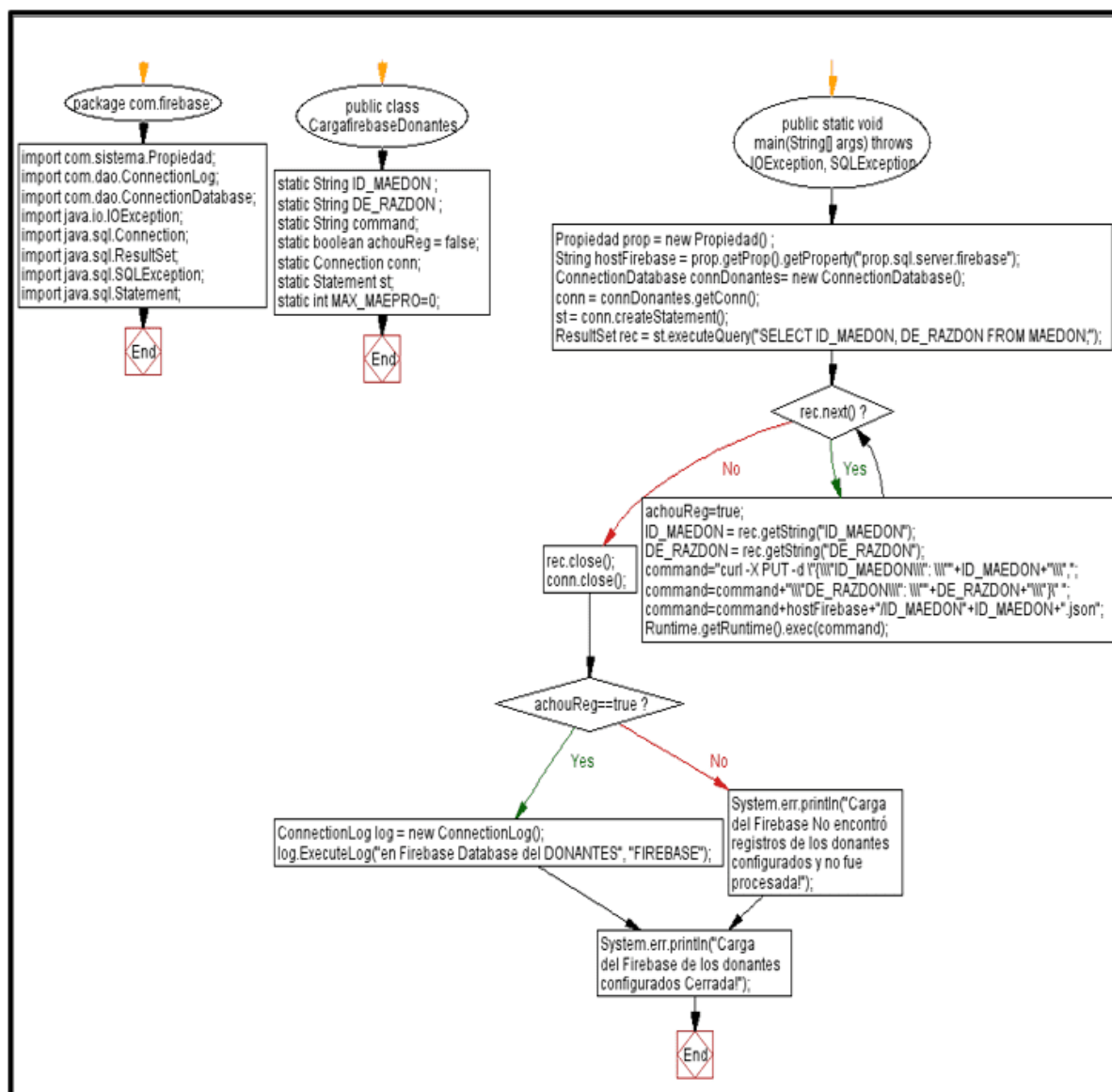
Após a execução de dados das tabelas solicitadas do SQL, estas são recuperadas e salvas em uma variável que, posteriormente, será concatenada e exportada como



arquivo *json*, usando uma solicitação *cURL* para fazer o upload dos dados do arquivo *json* para o *Firebase* (COSTA, 2013).

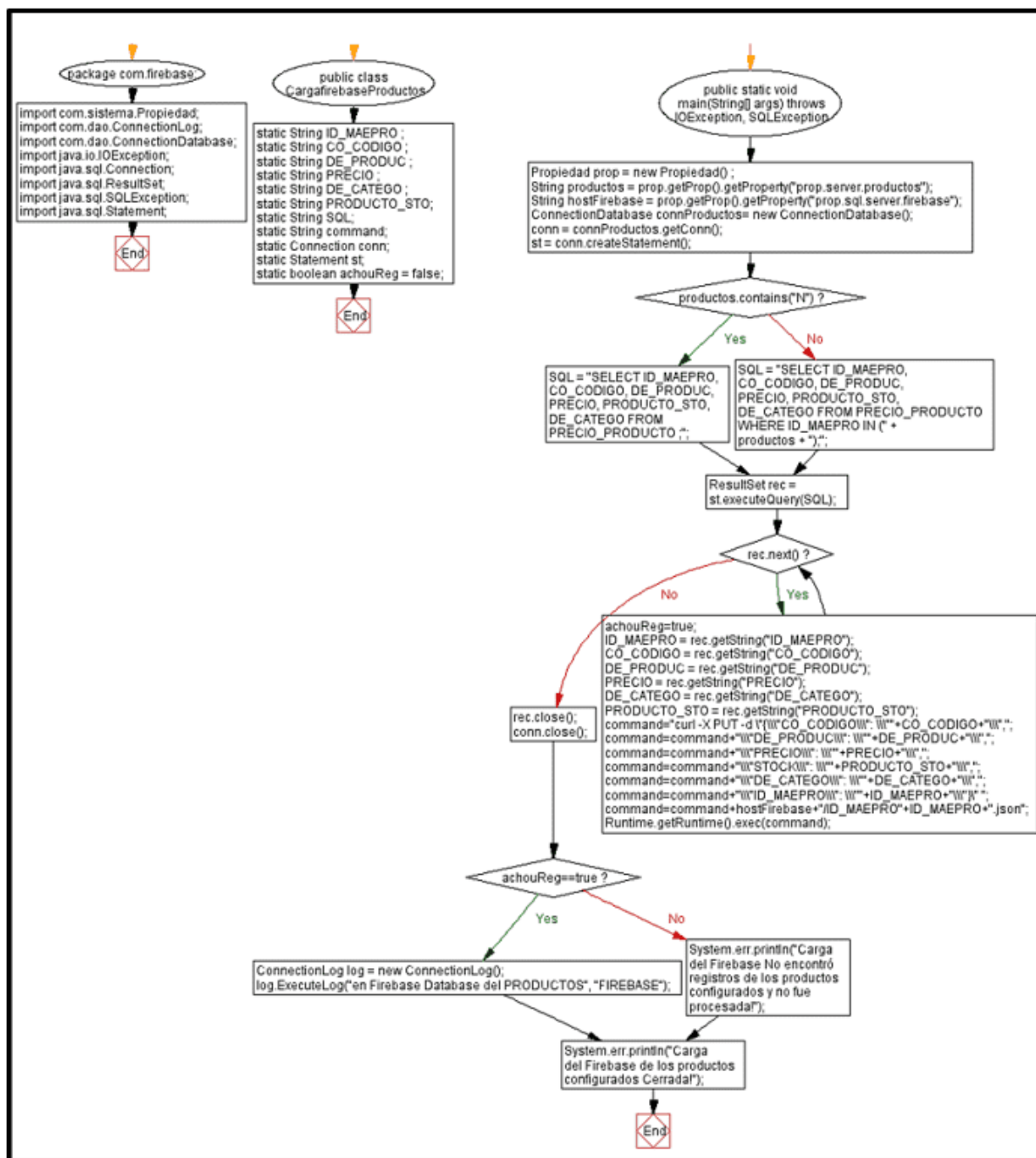
Na Figura 2, apresenta-se um fluxograma que descreve a forma em que ocorre o envio dos *ID's* e os nomes das instituições beneficiárias. Estes são usados para efetuar *login* no aplicativo. Já os dados são capturados no *SQLServer* através deste robô, o qual é programado para executar sua tarefa às 12:00. Por fim, os dados do *SQLServer* recuperados pelo robô são convertidos para *JSON*, que é o formato aceito na *Database Realtime* do *Firebase*.

Figura 2. Integração com a base de dados de instituições beneficiárias.



Fonte: do autor

Figura 3. Integração com a base de dados de produtos



Fonte: do Autor

Na Figura 3 se apresenta um fluxograma parecido com o indicado na figura 2 desta seção. Porém, tal fluxograma tem a função de alimentar o banco de dados do Firebase com informações dos produtos, como quantidade em estoque, id do produto, nome do produto, preço do produto e categoria do produto. Esses dados são recolhidos no



SQLServer através do mesmo robô utilizado – como é explicado na Figura 2. Essas informações dos produtos são obtidas por meio de uma solicitação cURL, que faz o upload dos dados (json), a fim de alimentar a base de dados na nuvem (FireBase).

4.2 FUNCIONALIDADES DESENVOLVIDAS

O aplicativo de forma geral, permite ao usuário fazer os pedidos, assim como possibilita que estes cheguem ao responsável por recebê-los no Banco Alimentario de La Plata. Paralelo a isto, os pedidos chegam em um outro aplicativo gerenciado por um administrador.

A aplicação consiste em uma interface simples e intuitiva para elevar ao máximo a experiência do usuário, tornando fácil a prática e utilização da aplicação. Isso torna-se factível porque ela, a aplicação, foi desenvolvida atendendo a alguns padrões de usabilidade, como objetividade, leiaute e interatividade. Assim, o usuário não ficará perdido em meio aos processos que tem de executar.

Nesta seção será abordado, também, algumas funcionalidades da aplicação, tais como:

- Processo de fazer um pedido;
- Tratamentos feitos com as datas disponíveis para realizar e fazer a retirada do pedido, de acordo à condição estabelecida pela ONG (Banco Alimentario de La plata).
- Adicionando produtos no carrinho;
- Subtotal dos produtos;
- Como acessar o histórico de pedidos feitos;
- *Logando* na aplicação.

Após o carregamento da tela inicial do aplicativo, o usuário é redirecionado para a tela de *login*. Nessa tela, colocará suas credenciais para ato seguido, acessar o aplicativo e fazer pedidos. Ainda, algumas outras funcionalidades foram implementadas na tela, como mensagens de erro, caso o usuário erre o *login* ou a senha (Figura 4). A



mensagem de erro é mostrada em espanhol, dado que esta é a língua nativa do país onde a aplicação é utilizada.

Figura 4. Tela de Login

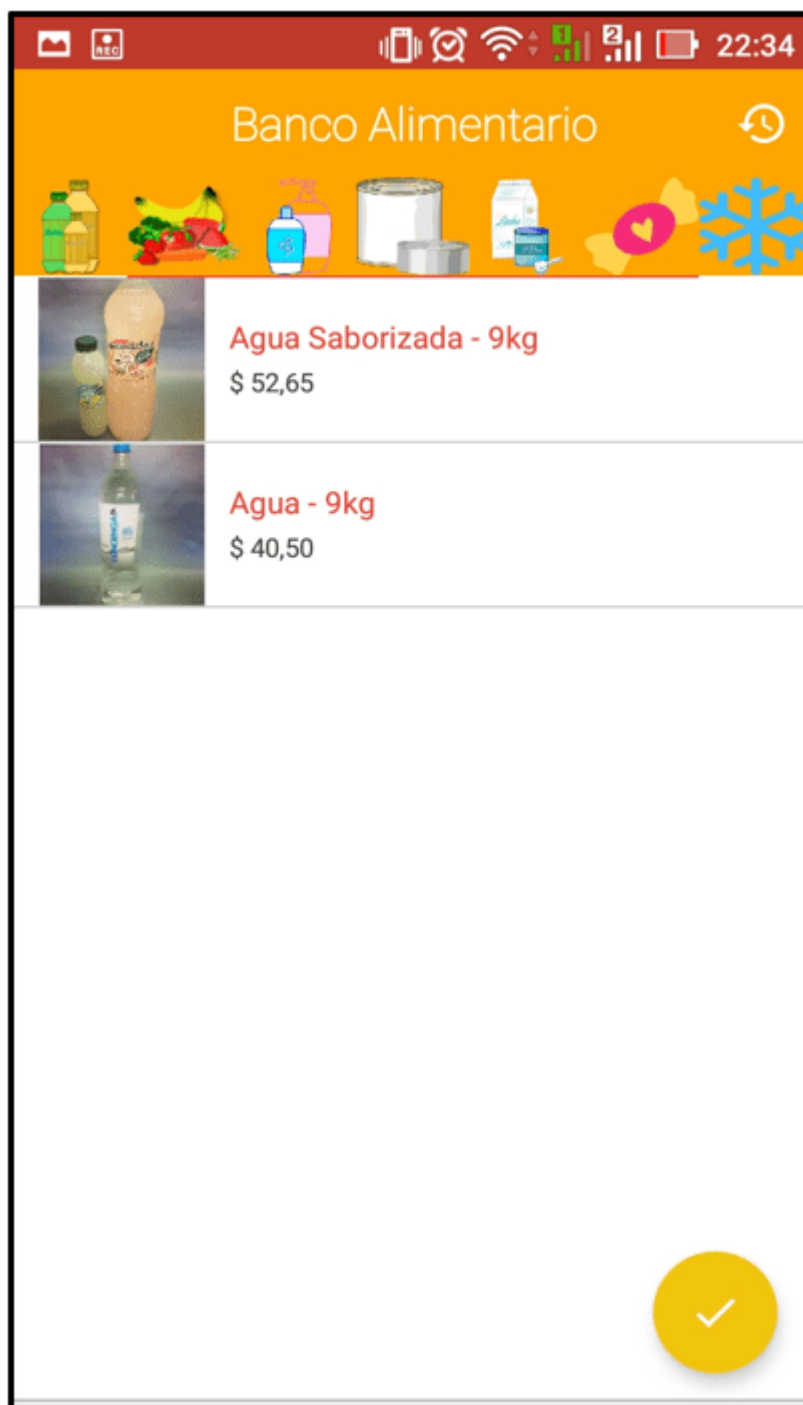


Fonte: do Autor

Após efetuar o login, o usuário é redirecionado para a tela em que é possível vislumbrar as categorias e os respectivos produtos (Figura 5). A interface é bem

intuitiva e, portanto, de fácil entendimento para o usuário, uma vez que este seleciona a categoria, e, logo em seguida, eleger o alimento que quer colocar em seu carrinho.

Figura 5. Tela de consulta de produtos por categoria



Fonte: do Autor

Na tela anterior, o usuário selecionará o produto. Logo em seguida será redirecionado para a tela apresentada na Figura 6, em que este deverá colocar a quantidade exigida de produto selecionado. Caso o produto selecionado não estiver disponível em estoque, o usuário receberá uma mensagem de tratamento do sistema, informando-o do impasse.

Figura 6. Tela de consulta por produto



Fonte: do Autor



Após selecionada a quantidade do produto, o usuário retorna à tela inicial de alimentos mostrada, na qual ele pode confirmar os produtos selecionados. Após esse passo, na tela atual do carrinho é mostrada a lista dos produtos selecionados, e na parte inferior da tela o subtotal da compra. O usuário pode segurar o *touch* em cima do produto, onde pode excluir do carrinho o que selecionar, podendo, também, voltar e adicionar mais produtos. Depois de confirmar os produtos do carrinho e avançar, o usuário preenche as informações para retirada dos produtos. A partir disso, a instituição deve, obrigatoriamente, informar seu nome ou código.

Em seguida, o usuário tem a opção de escolher guardar o pedido, a fim de efetuá-lo sem precisar preencher os dados novamente. Além disso, é possível consultar o histórico de pedidos, conforme se apresenta na Figura 7.



Figura 7. Tela de histórico de pedidos



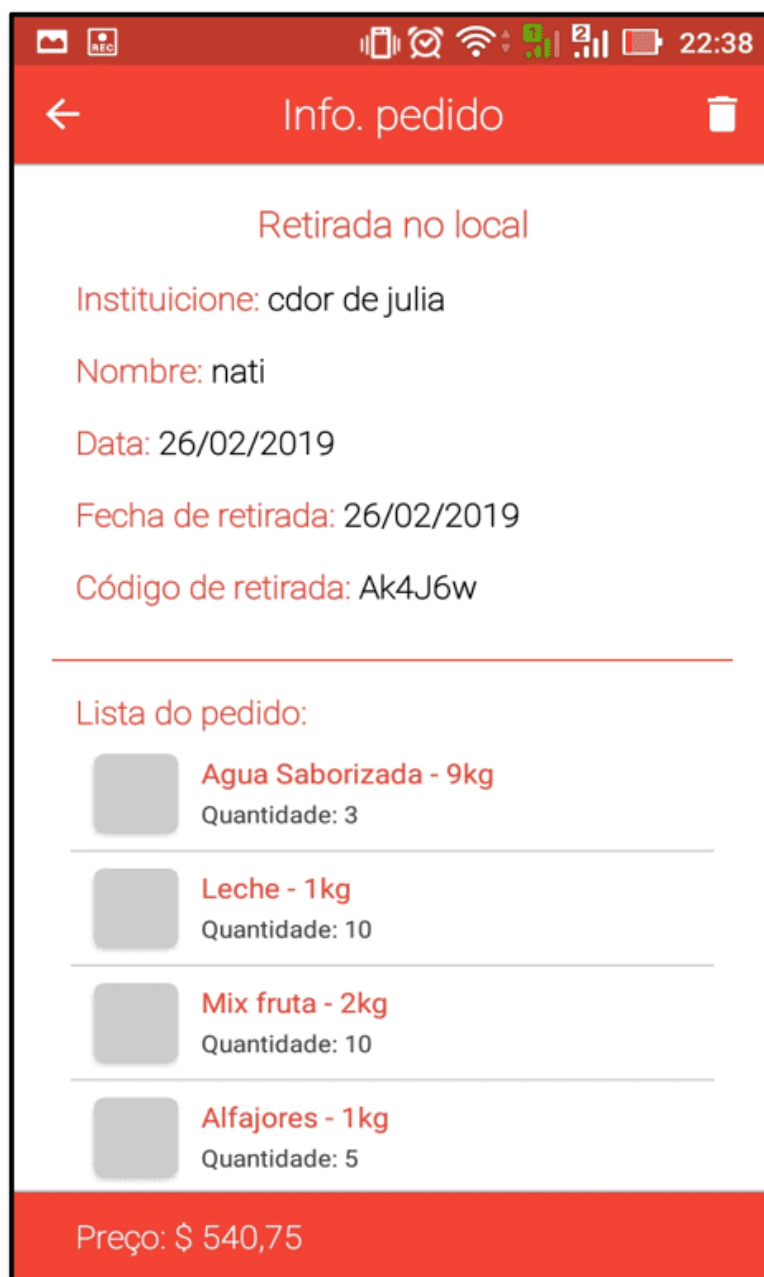
Fonte: do Autor



Nessa tela de administrador, conforme apresentado na Figura 7, mostra-se uma lista com todos os pedidos realizados pelos responsáveis de cada instituição. Os pedidos são feitos no aplicativo apresentado previamente neste documento.

Depois que o responsável pela gestão dos pedidos clique em algum pedido, este poderá ver as informações completas do que fora solicitado por ele. Tais petições serão mostradas numa interface semelhante à da figura 8. Tal tela contém, inicialmente, informações básicas e relevantes, como o nome do responsável pela instituição e seu código de retirada do pedido.

Figura 8. Tela de Informação de pedidos



Fonte: do Autor.

Ao clicar o administrador em um item da lista, esta se abrirá, expondo todas as informações do pedido, como: nome da instituição; nome do responsável pela busca do pedido; data em que o pedido foi feito; data em que o pedido vai ser retirado; e código de retirada.



O código de retirada é gerado quando o responsável pela instituição está quase finalizando o pedido. Esse código consegue ter acesso aos históricos de pedidos. Mais abaixo, na tela, é listada a quantidade dos produtos solicitados, e, por fim, o preço.

5. RESULTADOS

Assim que o aplicativo foi publicado na loja, os testes foram iniciados. Observou-se algumas melhorias no processo, que foram listadas logo abaixo.

- Os beneficiários dos alimentos puderam fazer pedidos simultâneos e em tempo real pelo aplicativo. Logo após o recebimento do pedido, este foi gerenciado pelo aplicativo de forma automatizada e rápida;
- Não foi mais necessário que um voluntário ficasse alocado no telefone para receber pedidos, visto que estavam sendo feitos pelo aplicativo;
- O voluntário que estava dedicado ao atendimento por telefone foi realocado, internamente, de área;
- O controle de estoque da ONG obteve um melhor controle, com a interação da base de dados em tempo real;
- As filas, outrora extensas na porta da ONG, acabaram. O agendamento pelo aplicativo com hora marcada facilitou o acesso dos beneficiários e a gestão de distribuição interna dos pedidos. Ou seja, o beneficiário, após as mudanças já aqui proferidas, chegará com um horário marcado via aplicativo, conseguindo que seu alimento já esteja disposto para ser retirado no momento de sua comparência;
- O controle das informações transitadas é guardado em sua base de dados, na nuvem, em tempo real. Os dados gerados foram utilizados para criar estatística, podendo ser utilizados para novos estudos comportamentais na distribuição de alimentos da ONG, com base em análise de dados.

Assim sendo, esses foram os principais ganhos obtidos com a troca da gestão de pedidos manual para a gestão feita com o aplicativo móvel.



6. CONSIDERAÇÕES FINAIS

Tendo em vista os fatos apresentados, conclui-se que o desenvolvimento de aplicativos móveis para a gestão de pedidos vem se tornando cada vez mais eficaz e necessário em um meio que utiliza de técnicas não otimizadas e muito esforço humano. Este tipo de esforço, como visto, pode ser substituído, sem prejuízos diretos, por sistemas capazes de fazer esse trabalho, evitando o gasto excessivo recursos, sem reduzir, com isso, sua rapidez de execução.

Com relação ao *Framework Kodular*, este foi de grande ajuda na agilidade com os processos de desenvolvimento, como abordado no tópico 3, seção 3.4. Ainda, o *framework* permite a criação de um aplicativo em menos de 2 minutos.

Todavia, foi de grande valia o desenvolvimento desse projeto no que tange ao crescimento pessoal e coletivo das pessoas envolvidas nesse empreendimento.

É importante salientar, ainda, que essa aplicação pode ser aprimorada. A seguir, algumas sugestões cabíveis:

- A criação de uma tela ou seção para o marketing digital do Banco Alimentario de La Plata, pois semanalmente acontecem eventos sociais com as instituições, a fim de arrecadar recursos para a ONG;
- Implementação de um componente chamado *One Signal* (ONE SIGNAL, 2019), que fornece mensagens *push*, *web push*, e-mail. No caso do aplicativo para dispositivos móveis, seria de utilidade enviar qualquer tipo de aviso em tempo real para os usuários do aplicativo, informando-os de algum alimento novo e/ou algum evento que irá acontecer; e qualquer outro tipo de mensagem necessária;
- Melhorias na interface de ambas as aplicações, tanto na de aplicação de pedidos, como na utilizada pelo administrador.



REFERÊNCIAS

APP INVENTOR. **With MIT App Inventor, anyone can build apps with global impact.** Disponível em: <https://appinventor.mit.edu/>. Acesso em: 05 jan. 2019.

COSTA, Danilo David Pereira. **Aplicativo para dispositivo móvel android para suporte ao processo de inventariamento de estoque.** 2015. 50 f. Trabalho de Conclusão de Curso (Especialização) - Universidade Tecnológica Federal do Paraná, Pato Branco, 2015. Disponível em: <repositorio.roca.utfpr.edu.br>. Acesso em: 29 jan. 2019.

DEVMEDIA. **Projetando e criando Aplicativos para Dispositivos Móveis.** Disponível em: <https://www.devmedia.com.br/projetando-e-criando-aplicativos-para-dispositivos-moveis/30671>. Acesso em: 25 fev. 2019.

FIREBASE. **Database Real-Time.** Disponível em: <https://firebase.google.com/docs/database?hl=pt-br>. Acesso em: 25 jan. 2019.

KODULAR. **Understanding Kodular.** Disponível em: <https://docs.kodular.io/guides/>. Acesso em: 10 fev. 2019.

MANZOTTI, Caio S. **Design de Interface em Dispositivos Móveis.** 2013.

ONE SIGNAL, **Push Notifications Delivered.** Disponível em: <https://onesignal.com/>. Acesso em: 18 jan. 2019.

ORACLE. **Sending API requests using cURL.** Disponível em: <https://docs.oracle.com>. Acesso em: 25 fev. 2019.

RIBEIRO, R.; FREIRE, P. Frameworks de Desenvolvimento Móvel Multiplataforma. In: **Conferência Da Associação Portuguesa De Sistemas De Informação**, 13., 2013. Anais. [S.l.: s.n.], 2013.

Enviado: Setembro, 2021.

Aprovado: Janeiro, 2022.

¹ Graduação.

² Orientador.