

Desenvolvimento de um protótipo de *self-checkout* por aplicativo para supermercados

Ellen L. Fernandes¹, Antonio M. M. Hachisuca¹, Eliane N. Pereira¹

¹Universidade Estadual do Oeste do Paraná (UNIOESTE)
Caixa postal 961 – 85.870-900 – Foz do Iguaçu – PR – Brazil

ellie.frnds@gmail.com, {antonio.hachisuca, eliane.pereira}@unioeste.br

Abstract. *The sales process in stores ends with queues at the cashiers, so to provide agility, some establishments offer self-service. This article describes the development of a prototype shopping application for smartphones, which aims to improve the self-checkout experience in supermarkets, so that customers have the autonomy to perform all the steps of their purchase without the need for a store employee to assist them. For this purpose, a survey of related works and technologies was carried out to establish the proposal implemented in the project, and as a result the application was obtained, able to assist in the process of scanning the items, calculates the total purchase amount and gives the consumer the option of making payment in an application for this purpose.*

Resumo. *O processo de venda em comércios é finalizado com filas nos caixas então, para prover agilidade, alguns estabelecimentos, oferecem o autoatendimento. Este artigo descreve o desenvolvimento de um protótipo de aplicativo de compras para smartphones, que visa aprimorar a experiência de self-checkout nos supermercados, para que os clientes tenham autonomia de realizar todas as etapas de sua compra sem a necessidade de um funcionário da loja auxiliá-lo. Para tanto, foi feita uma pesquisa de trabalhos correlatos e tecnologias para estabelecer a proposta implementada no projeto, e como resultado obteve-se o aplicativo que realiza a digitalização dos itens, calcula o valor total da compra e da a opção de efetuar o pagamento em um aplicativo com esse propósito.*

1. Introdução

As filas no final do processo de aquisição de bens e/ou serviços fazem parte do cotidiano das pessoas, no Brasil e em outros países do mundo, sendo comum que onde quer que elas estejam tenham que esperar alguns minutos, ou horas, para serem atendidas. Porém, a tecnologia vem ajudando cada vez mais a mudar esse cenário, possibilitando que tarefas que podiam ser resolvidas apenas indo ao estabelecimento agora possam ser realizadas remotamente, especialmente desde que se iniciou a pandemia de Covid-19 em que o isolamento e o distanciamento social foram necessários.

Por isso, algumas empresas com grande movimentação diária de pessoas buscaram e buscam formas para reduzir o tempo de espera de seus clientes em filas, utilizando técnicas como *delivery*, autoatendimento e uso de aplicativos de *smartphones* [Yuen 2018].

As instituições financeiras, por exemplo, já ofereciam aos seus clientes a possibilidade de realizar atividades remotas em suas contas por meio do *Internet Banking*.

Porém, a experiência nessas instituições foi aprimorada, devido ao uso dos *smartphones*, sendo possível agora criar uma conta em um banco sem sair de casa, além de poder pagar boletos, consultar saldo e outras transações bancárias.

O Governo Federal do Brasil está automatizando alguns de seus serviços, e devido a isso, agora é possível ter acesso a documentos, antes somente físicos, como a carteira de trabalho e a carteira nacional de habilitação em sua versão digital, pelo *smartphone*;

A Google, pensando em diminuir as longas filas geradas pelo trânsito, adicionou ao Google *Maps* a possibilidade de evitar os congestionamentos através da divulgação da situação do tráfego em tempo real. Dessa maneira, caso o usuário sempre passe por uma determinada rua e ela estiver com trânsito lento o aplicativo irá sugerir rotas alternativas.

Outro campo que vem tentando mudar a experiência de seus usuários com a redução das filas são os supermercados. Primeiramente foi utilizada a técnica de *delivery*, possibilitando que os consumidores realizem suas compras pela *internet* e recebam os produtos em casa. Essa opção ainda é oferecida, porém, os consumidores já estão acostumados “a comprar pessoalmente o produto: olhar, tocar o alimento, ver qual corte de carne está mais apetitoso”[Tooge 2020]. Dessa forma, essa estratégia não é a mais eficaz, pois os clientes continuam visitando o estabelecimento e terão que esperar para passar suas compras no caixa. Então, pensando em outras maneiras para diminuir as filas, alguns supermercados implantaram o sistema de *self-checkout*, também conhecido como *self-service checkout* (SCO) ou *semi-attended customer-activated terminal* (SACAT).

O sistema de autoatendimento transfere para o cliente a responsabilidade de finalizar a sua compra, sem a necessidade de passar os produtos escolhidos em um caixa convencional. O SCO pode ser composto por um leitor de código de barras, uma balança para pesar produtos como frutas e vegetais, uma área para guardar os itens e um sistema de pagamento, geralmente aceitando dinheiro, cartão de crédito ou débito [Taylor 2016]. Segundo pesquisa realizada pelo Ibope Inteligência, com mais de 2 mil pessoas, foi apontado que 86% desses consumidores esperam que os supermercados possuam o sistema de *self-checkout* nos próximos anos [De Chiara 2019].

O projeto relatado nesse artigo teve como objetivo criar um protótipo para um SCO no contexto de venda em um supermercado. Na seção que segue serão demonstrados algumas propostas semelhantes implantadas tanto no Brasil quanto no exterior. A seção 3 apresenta as tecnologias adotadas para o desenvolvimento do protótipo e a arquitetura proposta. A seção 4 descreve o aplicativo Smartmarket implementado e alguns apontamentos. E na seção 5 destaca-se a conclusão e trabalhos futuros.

2. Trabalhos relacionados

Como descrito anteriormente, a fila no final do processo de venda de serviços e produtos é um desafio recorrente para as empresas. Sendo assim, alguns exemplos de soluções adotadas por empresas internacionais e nacionais aplicando a técnica SCO foram identificadas para auxiliar na compreensão das estratégias do autoatendimento para agilizar a etapa de finalização das compras. A seguir serão descritas algumas dessas iniciativas.

O Hyper-mart, hipermercado localizado na China, utiliza a técnica de *self-checkout* e permite ao cliente as opções de pagar as suas compras escaneando um *Quick Response* (QR) *Code* por meio de seu *smartphone* ou posicionando-se em frente a uma

câmera para o reconhecimento facial, mas, para essa opção também é necessário informar o número do telefone registrado no país [Muralha 2019].

A Amazon Go é uma loja de conveniência que não possui caixas. Para entrar na loja é preciso desbloquear a passagem com o auxílio do aplicativo da loja no *smartphone*. O pagamento dos itens é creditado automaticamente no cartão do cliente, após ele escolher o que quer comprar e sair da loja. Mas, para essa solução funcionar, a loja possui diversos sensores no teto e nas prateleiras, além de utilizar inteligência artificial para o reconhecimento facial do cliente e para identificar os itens escolhidos [Igreja 2019].

A Zaitt é uma loja de conveniência brasileira, semelhante à Amazon Go. O cliente escolhe os itens que deseja comprar e os coloca em uma sacola. A principal diferença entre a Zaitt e a Amazon Go é no momento do pagamento. A alternativa encontrada pela Zaitt, para substituir os sensores presentes na loja americana, foi a de trocar o código de barras por um *QR Code* em todos os itens, para facilitar a leitura dos mesmos. Assim, para finalizar a compra, será feito o reconhecimento facial do cliente e de todos os itens presentes na sacola. Após essa identificação, será exibida uma lista com os produtos que foram escolhidos, se tudo estiver correto o cliente confirma a compra e pode sair da loja [NOVAREJO 2019].

A Caper introduziu nos supermercados um modelo de compras inovador, com o intuito de oferecer uma experiência completa de autoatendimento. O *self-checkout cart* da Caper, figura 1, possui: uma interface para comunicação com os usuários, máquina de cartão de crédito ou débito, leitor de código de barras, sensores para identificar quando o consumidor coloca um item no carrinho, sensor para pesar hortifrúteis e uma câmera para reconhecimento de imagem [Sanchez-Iriarte 2019]. Dessa forma, o cliente não precisa retirar seus produtos para passar no caixa, pois tudo foi realizado no próprio carrinho de compras.



Figura 1. Carrinho da Caper. Fonte: [Constine 2019, Sanchez-Iriarte 2019]

O Sam's Club, supermercado do grupo Walmart, criou um aplicativo para realizar o *self-checkout*, em que é possível ler o código de barras dos itens e pagar diretamente pelo *smartphone* [Tech 2018]. Porém, a rede no Brasil, segundo [em Dia 2018] o pagamento pelo aplicativo ainda não é oferecido. Assim, o cliente precisa ir para um caixa, sendo necessário apenas mostrar o código de barras gerado no aplicativo, que contém o valor total da compra. O que ajuda a reduzir o tempo de espera na fila, pois todos os itens já foram lidos previamente com o auxílio do aplicativo.

Dentre os trabalhos relacionados encontrados, percebe-se que algumas lojas conseguem evitar totalmente as filas, pois não possuem caixas, seja ele convencional ou por autoatendimento. Além dessas lojas oferecerem ao cliente mais autonomia, como por exemplo, a de realizar o pagamento de suas compras por meio de um aplicativo, não sendo necessário que um funcionário fique a disposição para realizar a finalização do processo de compra. Com isso, também, é possível que o estabelecimento funcione 24 horas por dia.

Os desafios dos supermercados ainda são variados para se conseguir diminuir as filas na finalização das compras, porque, diferentemente, das lojas de conveniência o cliente que procura esse comércio compra diversos itens. E, mesmo utilizando a técnica de *self-checkout* o cliente continua perdendo o mesmo tempo que gastaria em um caixa convencional, ou até mais, pois precisa retirar item por item do seu carrinho para passar no leitor de código de barras, além de ter que empacotar e guardar as compras no carrinho novamente. Portanto, nota-se que a etapa de finalização da compra, em que o cliente precisa retirar os itens do carrinho para colocar no caixa, é o principal fator para os supermercados ainda não conseguirem diminuir as filas de maneira significativa. Para o superintendente da Associação Paulista de Supermercados (APAS) “O grande desafio é fazer com que este caixa não exista. [...]. A partir de hoje. Seria essencial a opção de nem precisar tirar do carrinho.”[Oliveira 2020].

3. Tecnologias de Desenvolvimento e Arquitetura da Proposta

A proposta desse projeto é a de desenvolver um protótipo, capaz de proporcionar ao consumidor uma experiência com tempo zero em filas de caixas, seja ele tradicional ou por autoatendimento. Para isso, será utilizada uma abordagem semelhante à apresentada pelo Sam’s Club. Porém, com o diferencial de poder fazer o pagamento pelo *smartphone*, com a geração de um *QR Code*, utilizando o modo *Merchant-Presented Mode (MPM)*, que será descrito em seguida, para os usuários concluírem o pagamento em carteiras ou bancos virtuais.

Esse projeto partiu da premissa que não acontecerão tentativas de furtos dos itens do supermercado. Ou seja, não foi implementada nenhuma alternativa para assegurar ao estabelecimento que todos os itens escolhidos pelos clientes foram pagos.

A Figura 2 apresenta a arquitetura computacional do projeto, que será utilizada para apresentar as tecnologias adotadas na implementação da proposta.

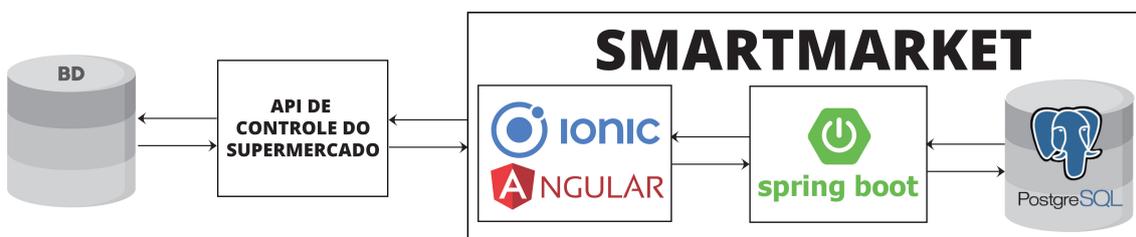


Figura 2. Arquitetura computacional do projeto

Para desenvolver o projeto foi preciso implementar uma aplicação *web* auxiliar. Isso foi necessário, pois, pensando em um cenário real, o supermercado já teria uma aplicação com o propósito de gerenciar o estoque de seus produtos, assim como outras

informações que sejam do interesse do estabelecimento. Portanto, não faria sentido o supermercado ter que informar novamente esses dados no sistema do aplicativo de compras. Então, a *Application Programming Interface* (API) de Controle do Supermercado possui o papel de manter os fabricantes e os produtos que são vendidos. O manter fabricantes faz referência as ações: cadastrar, atualizar, excluir e exibir em forma de lista ou de página. O manter produtos possui essas mesmas ações e também a de consultar produtos, que será utilizada pelo aplicativo de compras (*Smartmarket*) para obter os dados do produto quando o cliente fizer a leitura do código de barras.

O aplicativo do carrinho de compras virtual (*Smartmarket*) possui o papel de manter: a conta do cliente, o carrinho de compras e os itens. Além disso, também é responsável pela geração do *QR Code* que será utilizado para o pagamento.

As tecnologias apresentadas a partir daqui são as utilizadas para a implementação do *Smartmarket*, que é o ponto central da proposta desse projeto. A primeira delas é o *framework* Ionic que oferece uma biblioteca de componentes de *User Interface* (UI) otimizados, para dispositivos móveis e *desktops*. Uma de suas vantagens é proporcionar agilidade na construção dos aplicativos com alta interatividade como, por exemplo, o *plugin Barcode Scanner* que faz a leitura do código de barras através da câmera do dispositivo móvel, retornando para a aplicação o dado contido no código [Ionic 2020, Ionic 2021].

O Angular foi adotado por ser uma plataforma *open-source*, desenvolvido pela Google e utiliza a linguagem de programação TypeScript. Essa linguagem é um superconjunto do JavaScript, com o acréscimo de tipagem do código e recursos adicionais, como: classes, interfaces, *enum*, *async/await*, entre outros. Mas, após compilar um programa em TypeScript ele será convertido para JavaScript, retirando todas as representações de tipos, pois eles não existem em tempo de execução. [Bierman et al. 2014].

O Spring Boot é um *framework* que visa simplificar o desenvolvimento de uma aplicação, fornecendo opções de configurações automáticas para o diagnóstico e solução de dependências através do *starter dependencies* [Walls 2015]. Alguns exemplos de dependências são: o Spring *Security* utilizado para configurações de segurança como, por exemplo, a autenticação do aplicativo e o Spring *Data JPA* para comunicação com o banco de dados.

Como parte da solução para o pagamento será utilizado o *QR Code*, que no Brasil é emitido dentro dos padrões especificados pelo Banco Central, sendo chamado de *BR Code*. Esse tem o intuito de padronizar os meios de pagamento utilizados no país, além de possibilitar que um *QR Code* aceite diversos tipos de arranjos de pagamento. O padrão do *BR Code* foi elaborado seguindo os documentos ISO/IEC 18004, que especifica como deve ser criado um *QR Code*, e o *QR Code Specification for Payment Systems* da EMV®, que estrutura os dados de pagamento [do Brasil 2020].

Para gerar um *QR Code*, que efetue transações comerciais, é necessário que seja definido primeiramente o modo de pagamento. Para que se saiba a estruturação dos dados que estarão codificados no *QR Code*. Existem dois modos de pagamento: o *Consumer-Presented Mode* (CPM) e o *Merchant-Presented Mode* (MPM). No modo CPM é necessário que o estabelecimento faça a leitura do *QR Code* contendo os dados do cliente. Já no MPM é o cliente que precisa escanear o código que possui as informações do comerciante [EMV® 2020a, EMV® 2020b], sendo esse o modo escolhido para esse projeto.

4. Smartmarket: self-checkout app

O aplicativo, Smartmarket, foi elaborado para permitir que os usuários consigam realizar as suas compras de forma intuitiva, sem complicações e que possam finalizar a compra com agilidade. Para isso foram desenvolvidas poucas funcionalidades, dando foco as que realmente fazem parte da execução do objetivo do aplicativo, que é: o registro dos itens, cálculo do valor da compra e a geração do QR Code para o pagamento.

Para a comunicação com os usuários foram desenvolvidas algumas interfaces. A primeira interface do aplicativo é a tela de *login*, Figura 3.a, em que o usuário deve informar o seu *username* e a sua senha. Ou, caso ainda não tenha cadastro no aplicativo, é necessário informar os dados: nome, *e-mail*, *username* e senha, como mostra a figura 3.b.

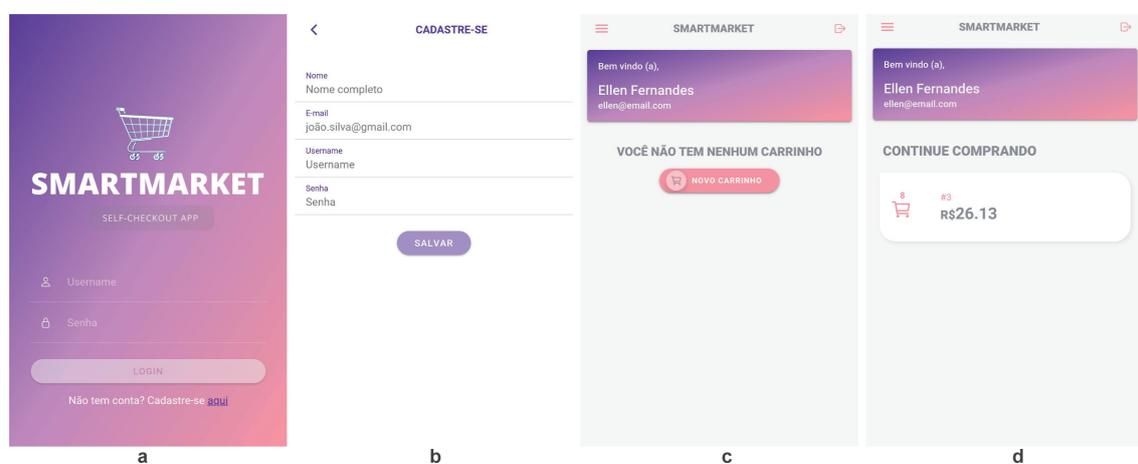


Figura 3. a) Login, b) Cadastro, c) Home vazia e d) Home com carrinho.

Quando o cliente entra no aplicativo pela primeira vez, será exibida uma mensagem que o cliente não possui nenhum carrinho, e para começar as suas compras ele precisa criar um novo carrinho, Figura 3.c. Mas, se o cliente já realizou essa etapa, será exibido o carrinho de compras, com as informações do valor total da compra e a quantidade de itens, como mostra a Figura 3.d.

A tela do carrinho de compras, Figura 4.a, exibe todos os itens adicionados no carrinho, com o seu preço unitário e o preço total de acordo com a quantidade adicionada. Para ajustar a quantidade ou excluir um determinado item do carrinho o cliente precisa arrastar o item para o lado esquerdo. Caso a ação escolhida for a de excluir um item, será exibido um alerta na tela, para confirmar se o cliente realmente deseja remover o item. Nessa tela também está a opção “escanear item”, que irá abrir a câmera do *smartphone* do cliente, para fazer a leitura do código de barras, Figura 4.b.

A interface de pagamento, exibida na Figura 4.c, conta com o BR Code, nele foram adicionados dois arranjos: um para pagamento por PIX, que pode ser realizado em qualquer banco ou carteira virtual, e outro para pagamento em uma carteira virtual específica, no caso foi escolhido o Picpay. Mas, esses arranjos de pagamento podem ser modificados de acordo com as opções oferecidas pelo estabelecimento.

Abaixo do BR code é exibido um resumo da compra, com a quantidade total dos itens presente no carrinho, a forma de pagamento, que no caso desse aplicativo é



Figura 4. a) Carrinho virtual, b) Lendo o código de barras, c) Pagamento e d) Comprovante.

apenas por QR code, e o valor total da compra. Também foi adicionado um botão de “enviar comprovante”, que ajuda a simular a confirmação do pagamento do cliente, pois, o carrinho possui uma *flag* que indica o seu *status* de pagamento, que pode ser pendente ou pago. Dessa forma, enquanto o carrinho estiver pendente ele permanecerá na tela inicial do aplicativo e pode ser editado, ou seja, o cliente pode continuar sua compra, adicionando e removendo itens do carrinho. Mas, se o cliente enviar o comprovante e o *status* for alterado para “pago”, o carrinho é enviado para o histórico e não poderá mais ser modificado. Na Figura 4.d é exibido um resumo de uma compra que já foi enviada para o histórico.

5. Conclusão e Trabalhos Futuros

O intuito deste projeto foi o de apresentar mais um problema cotidiano que poderia ser resolvido usando um aplicativo. Dessa forma, realizando o autoatendimento pelo *smartphone* é possível reduzir os custos de implantação do sistema, pois, não é necessário equipar ou modificar a loja para o funcionamento da solução, além do cliente não precisar da assistência de um funcionário do mercado para fazer o pagamento de suas compras. Além disso, a etapa de finalização das compras pode ser executada rapidamente, já que o cliente não precisa retirar os itens do carrinho e nem mesmo aguardar em uma fila para ser atendido. Logo, a experiência de compras consegue ser aprimorada, fazendo com que o cliente tenha total controle sobre o tempo que ele levará para começar e finalizar as suas compras, pois, esse fator dependerá exclusivamente dele. Portanto, alguns benefícios que podem ser citados do aplicativo para *self-checkout* são: autonomia e rapidez na finalização das compras para o cliente e possibilidade de flexibilização do horário de atendimento para o estabelecimento.

Em testes iniciais foi possível validar as funcionalidades propostas do projeto, conseguindo recuperar as informações dos produtos através da leitura do código de barras no aplicativo, realizando o cálculo do valor total do produto e total da compra, assim como a geração correta do QR Code com o valor total da compra, sendo possível fazer o pagamento em um banco ou uma carteira virtual.

Algumas funcionalidades necessárias para a versão final do aplicativo são: a

confirmação de pagamento, pois como foi explicado, neste projeto foi utilizado uma *flag* para simulação, e a partir dela pode ser realizada a atualização do estoque do supermercado; identificação de produtos pesáveis; pagamento pelo aplicativo, podendo ser incluídas as opções de pagamento como crédito, débito, PIX, entre outras formas aceitas pelo estabelecimento; segurança, para garantir que os produtos do supermercado não sejam furtados. Outras funcionalidades interessantes, mas não essenciais, para futuras implementações são: itens favoritos; conversão de moeda, informando os valores dos produtos e o valor total da compra conforme a moeda escolhida pelo consumidor; Cadastro de Pessoas Físicas (CPF) para nota fiscal, como campo opcional; detalhar informação dos produtos, adicionando o peso do produto, foto, entre outras informações.

Referências

- Bierman, G., Abadi, M., and Torgersen, M. (2014). Understanding typescript. In *European Conference on Object-Oriented Programming*, pages 257–281. Springer.
- Constine, J. (2019). Meet caper, the ai self-checkout shopping cart.
- De Chiara, M. (2019). No supermercado do futuro, autoatendimento é a função mais desejada.
- do Brasil, B. C. (2020). Manual do br code.
- em Dia, J. H. (2018). Sam's clube cria aplicativo que lê código de barras e calcula valor total da compra.
- EMV® (2020a). Emv® qr code specification for payment systems (emv qrcps): Consumer-presented mode.
- EMV® (2020b). Emv® qr code specification for payment systems (emv qrcps): Merchant-presented mode.
- Igreja, A. (2019). Testando a loja amazon go em san francisco - arthur igreja.
- Ionic (2020). Ionic cross-platform mobile app development.
- Ionic (2021). Barcode scanner.
- Muralha, P. (2019). Mercado na china: só bizarrices? — pula muralha.
- NOVAREJO (2019). Zaitt - como funciona o primeiro supermercado 100% autônomo?
- Oliveira, P. (2020). “o supermercado é essencial, mas o consumo mudará” afirma superintendente da apas.
- Sanchez-Iriarte, E. (2019). Make shopping magic.
- Taylor, E. (2016). Supermarket self-checkouts and retail theft: The curious case of the swipers. *Criminology & criminal justice*, 16(5):552–567.
- Tech, A. (2018). Sam's club scan and go app tutorial.
- Tooge, R. (2020). Vendas online de supermercados quase dobram após o coronavírus, entregas atrasam e exigem investimentos das empresas.
- Walls, C. (2015). *Spring Boot in action*. Simon and Schuster.
- Yuen, D. (2018). Zero customer wait time: How to minimize customer wait time with mobile phone apps.