



MODERNIZAÇÃO DO ATENDIMENTO AO CONSUMIDOR EM BARES E RESTAURANTES

Juan Elias da Cunha

Ezequias Ferreira de Souza

Curso: Análise e Desenvolvimento de Sistemas Período: 6º

Área de Pesquisa: Desenvolvimento de Aplicativos Móveis

Resumo: O presente trabalho tem como objetivo observar a interação entre estabelecimentos do ramo alimentício e seus consumidores, identificando problemas frequentes e propondo um sistema acessível capaz de centralizar a realização das tarefas, afim de minimizar a ocorrência de problemas referentes ao atendimento, recordar dados para consultas posteriores e arquitetado de forma que seja fácil realizar manutenções ou alterações pertinentes, possibilitando seu crescimento futuro. Atualmente o método de atendimento mais comum bares e restaurantes coloca sob responsabilidade do cliente o controle do consumo e a responsabilidade sobre a comanda, exigindo que este pague uma multa em caso de perda (práticas ilegais segundo os artigos 39 e 51 do código de defesa do consumidor), quando isso não ocorre e o estabelecimento se responsabiliza pela comanda, o consumidor não tem acesso aos produtos adicionados, estando sujeito a um pagamento indevido e confiando na índole do responsável pelo atendimento. Ponderando sobre o questionamento: é possível desenvolver um aplicativo móvel capaz de simplificar o método de atendimento por comandas? A ferramenta proposta durante a elaboração deste trabalho tem como objetivo simplificar as tarefas já presentes no cotidiano dos comerciantes, interagindo também com seus consumidores a fim de simplificar a comunicação entre as partes.

Palavras-chave: Bares e Restaurantes; Comanda; Conflito; Aplicativos Móveis.

1. INTRODUÇÃO

Durante a visitação a bares e lanchonetes foi observado que, na grande maioria dos casos, o método de atendimento utilizado segue o modelo de comandas, neste o estabelecimento entrega ao consumidor um pedaço de papel onde são anotados produtos consumidos, alguns dos estabelecimentos visitados possuem um método um pouco mais sofisticado em que o pedaço de papel é substituído por um cartão de PVC contendo um código de barras, diferente do papel, este cartão pode ser reutilizado, uma vez que os itens anotados são excluídos no momento do pagamento. Em ambos os casos, o cliente deve apresentar sua comanda toda vez que realiza um pedido para que sejam anotados os produtos consumidos. Caso a comanda seja perdida ou danificada de forma que seja impossível sua leitura é cobrada uma multa do cliente, entretanto, esta forma de atendimento é prevista pelos artigos 39 e 51 do código de defesa do consumidor [BRASIL, 1990, on-line] e tornam respectivamente ilegal que a responsabilidade sobre o que foi consumido seja transferida ao cliente ou que a multa seja cobrada. Sendo assim, o controle sobre os produtos consumidos é de integral responsabilidade do estabelecimento. Onde a regra é devidamente respeitada, o estabelecimento retém a comanda e o consumidor não tem acesso aos produtos adicionados. Durante a visitação a um estabelecimento que segue as devidas regras, foi vivenciada a situação em que o atendente do estabelecimento anotou na comanda um produto que o consumidor alegava não ter consumido, uma vez que o cliente se tornou ciente do ocorrido somente no ato do pagamento, este optou por pagar pelo produto a fim de evitar demais problemas.

Pensando em situações como esta, pergunta-se: é possível desenvolver um aplicativo móvel capaz de simplificar o método de atendimento por comandas? Assim, foi iniciado o desenvolvimento do *software* para *smartphones*, posteriormente denominado Table4All. Este pode ser utilizado tanto pelo estabelecimento quanto por seus consumidores possibilitando que ambos tenham uma cópia da comanda e acesso a relatórios diversos.

2. DESENVOLVIMENTO

Após observados os problemas no que diz respeito aos estabelecimentos e seus clientes, iniciou-se o processo de elaboração de maneiras para simplificar os processos e solucionar as adversidades para ambos, colocando em cenário o

desenvolvimento de um aplicativo para *smartphones*, uma vez que esse é de fácil acesso para todos os envolvidos, necessitando apenas do aparelho portátil e de conexão com a Internet.

Para desenvolver o sistema foram empregadas as habilidades desenvolvidas durante o curso. O desenvolvimento é dividido em várias camadas como estudo de viabilidade, escolha do método de desenvolvimento, programação, *design*, interface homem-máquina e também leva em consideração aspectos contextuais como público alvo e legislação.

A começar pelo estudo de viabilidade, é possível ressaltar a ampla utilização do arcaico método de comandas, ocasionando à equivalente ocorrência dos problemas que o acompanham, entretanto, a utilização de um sistema para gestão das comandas pode acelerar o processo de atendimento ao cliente em até 30% além de reduzir a ocorrência de problemas durante o atendimento (COSTA e OLIVEIRA, 2018), criando assim um cenário propício para que o sistema a ser desenvolvido neste trabalho se torne uma ferramenta potencialmente eficaz.

Para realizar o desenvolvimento do sistema foi adotado o modelo de desenvolvimento em espiral. Segundo Boehm (2000), este consiste em realizar o desenvolvimento em um ciclo de etapas, realizando testes e coletando resultados sempre que uma nova função é implementada ao sistema, encontrando possíveis erros assim que são inseridos no sistema.

O sistema deve ser composto não somente de uma parte lógica que realiza os processos, mas também deve possuir um *design* agradável e que torne fácil a utilização e compreensão do *software* pelo usuário, enquanto a não observação às boas práticas desta área pode acarretar falhas, lentidão e até o insucesso do sistema (KRUG, 2014), portanto, foram adotadas duas técnicas para criar uma interface simples e amigável. A primeira delas é conhecida como cromoterapia, os estudos desta área mostram que a cor verde é a segunda favorita dos seres humanos além de remeter a sentimentos como calma e funcionalidade além de tranquilidade e segurança quando combinado com a cor branca (HELLER, 2007), portanto a combinação de verde e branco foi a escolhida para compor principalmente o sistema. O segundo estudo utilizado para criar a parte visual do *software* foi o Minimalismo, este é construído em torno da máxima “menos é mais” do arquiteto alemão Ludwig Mies van der Rohe, retratando a situação financeira da Alemanha após a Primeira Guerra Mundial (MAGILL, 1999, p. 2520). Esta técnica consiste em adicionar o mínimo

de elementos necessário para o entendimento do usuário e eficácia do produto, ainda segundo Krug, ornamentos podem confundir o usuário com partes funcionais do sistema.

Para desenvolver a aplicação a linguagem escolhida foi o React Native, desenvolvido e mantido pelo FacebookTM, sendo uma ótima alternativa por funcionar simultaneamente em Android e IOS os principais sistemas operacionais do mercado para dispositivos móveis, além de ser baseado no modelo de componentes, o que permite a reutilização de fragmentos do código (ESCUDELARIO e PINHO, 2020), sendo um case de sucesso ao ser utilizado não apenas pela desenvolvedora como por outras marcas reconhecidas e amplamente disseminadas como DiscordTM, TeslaTM, SkypeTM entre outros descritos pela própria criadora da linguagem no site oficial da linguagem. Foi utilizada ainda a ferramenta *Expo*, que gerencia os arquivos do projeto e possibilita a instalação de bibliotecas, ou seja, blocos de códigos criados para simplificar atividades recorrentes (FERNANDES, 2018). Entre elas, *React Navigation* para facilitar a navegação entre as telas e criação da barra de navegação presente na parte inferior das telas do aplicativo. Foi utilizada a biblioteca *React Native Paper* para simplificar a criação de botões do tipo rádio, utilizados na página de criação de usuário, para estilizar os componentes como botões e campos de texto foi utilizada a biblioteca *Styled Components*. Em determinados momentos de seu funcionamento, o *software* armazena informações no dispositivo do usuário para consultas posteriores, para tanto utiliza-se a biblioteca *Async Storage*. Para exibir imagens como a logo e os ícones do sistema, é necessário que estas estejam salvas nas pastas do sistema utilizando em formato SVG (abreviação para Gráficos Vetoriais Escalonáveis, imagens neste formato não perdem sua qualidade ao serem redimensionadas), e então a biblioteca *React Native SVG* trata de exibi-las utilizando os tamanhos especificados.

As informações básicas para a execução de qualquer aplicativo foram criadas automaticamente pela ferramenta descrita anteriormente, então, foi adicionada uma pasta com o nome “*source*” (do inglês: fonte). Dentro desta pasta foram criadas novas pastas, organizando os arquivos utilizados para a construção do *software*, como as telas, as imagens utilizadas, instruções para navegação entre as telas e ainda para a criação de campos de texto e botões de uso recorrente dentro da plataforma. Cada tela criada possui uma pasta única dentro do diretório de pastas, dentro da pasta correspondente, são mantidos dois arquivos, um com o nome de “*index*”, contendo as

instruções funcionais da aplicação e outro denominado “*styles*” que é responsável pela parte visual do conteúdo.

As informações do sistema são armazenadas em um servidor na internet para possibilitar seu acesso remoto, sendo assim, sempre que o *software* necessita das informações salvas no banco de dados, é criado um pacote de informações fornecidas pelo usuário, ou previamente salvas no dispositivo, em um formato específico denominado *JSON* (abreviação do inglês para Notação de Objetos em *JavaScript*, descrita por Crockford (1999) como uma lista ordenada de valores), então o pacote é enviado para uma API (abreviação do inglês para Interface de Programação para Aplicações), ou seja, um arquivo com instruções para utilização das informações enviadas e que é mantido no servidor do banco de dados, esta trabalha como um intermediário, traduzindo os dados e instruções. Logo, as informações são utilizadas para buscar as informações desejadas no banco de dados e criar um novo pacote *JSON* para então enviá-lo de volta ao *software* principal onde as informações serão utilizadas para exibição ao usuário.

Para o funcionamento do sistema é necessário que exista um banco de dados, um programa secundário onde são armazenados os dados do sistema, este fornece as informações requisitadas, no caso do Table4All, é hospedado na internet podendo ser acessado pelos usuários do sistema de qualquer lugar e não ocupa o dispositivo do usuário com informações desnecessárias. Sendo assim foi construído um banco de dados MySQL. Nele as informações são arquivadas em tabelas, que guardam em cada linha as informações referentes a um usuário separando suas informações em colunas, o nome de cada coluna utiliza o nome da tabela seguido de um *underscore* (travessão) e o nome da informação armazenada. Como exemplo, podemos apresentar a tabela de usuários do Table4All, nela podemos observar informações como um identificador sequencial único, criado automaticamente pelo sistema interno do banco de dados assim que a nova conta é criada, uma senha escolhida pelo usuário criptografada no padrão MD5 (*Message Digest algorithm*, equivalente em português a Mensagem Compilada, e 5 refere-se ao número da versão), que não pode ser decifrada utilizando cálculos matemáticos como ocorre com outros tipos de criptografia. Um recorte contendo todas as informações de dois determinados usuários podem ser encontradas nas Figuras 1 e 2.

Figura 1 - Recorte do banco de dados parte 1.

user_id	user_tipo	user_nome	user_sobrenome	user_senha
1	0	Juan	Elias da Cunha	ba4ab0c88abec839568de2962bc7b74e
2	0	Rian	Elias da Cunha	096a1a1ab32bb70f139ce94e8246e5dd

Fonte: Acervo pessoal.

Figura 2 - Recorte do banco de dados parte 2.

user_email	user_celular	user_avatar	user_mesa	user_since
juan.elias.cunha@gmail.com	12345678	NULL	0	2021-04-28 21:09:56
rian@hotmail.com	1	NULL	0	2021-04-30 00:30:46

Fonte: Acervo pessoal.

A tabela apresentada na imagem contém campos que ainda não são utilizados pelo sistema, guardando informações como avatar ou data de criação da conta, entretanto, a presença destes espaços possibilita a fácil expansão das funcionalidades já apresentadas. Algumas tabelas armazenam dados de outras tabelas, como o trecho da tabela de históricos apresentada na Figura 3.

Figura 3 - Recorte da tabela de históricos.

hist_id	hist_idusuario	hist_idmesa	hist_estab	hist_valor	his_data
10	1	3	12	35.00	2021-05-08 19:09:08
11	5	3	12	35.00	2021-05-08 19:09:08

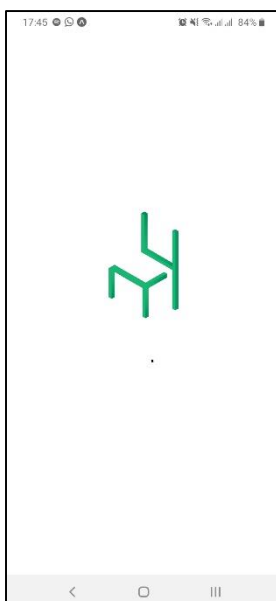
Fonte: Acervo pessoal.

Durante a criação desta tabela foi considerada a adição de um campo que não estava presente em seu estado inicial, a fim de armazenar o estabelecimento ao qual a mesa gravada pertence, pois assim, as consultas dos arquivos em PHP (abreviação do inglês para Pré-processador de Hipertexto, linguagem de programação utilizada na elaboração das API's) poderiam ser simplificadas em uma única consulta.

3. APRESENTAÇÃO DO SISTEMA

O acesso começa pela tela de pré-carregamento (Figura 4), que verifica se existem acessos anteriores do sistema no dispositivo. Em caso positivo uma tentativa de acesso é efetuada utilizando as credenciais encontradas, caso as informações sejam válidas as funcionalidades são liberadas ao usuário, caso as informações sejam inválidas ou inexistentes, o sistema exibe a tela de autenticação do usuário (Figura 5), onde, um determinado usuário, pode criar uma nova conta ou acessar uma conta previamente registrada utilizando o e-mail cadastrado junto à senha escolhida. Pensando em um cenário onde o usuário queira acesso imediato às funcionalidades do sistema sem a necessidade de cadastro, foi incluído um botão para acessar como visitante, que ao ser tocado, exibe um alerta de que o usuário em questão não terá acesso às informações da sessão no futuro, caso este concorde, recebe acesso às funcionalidades do sistema, caso contrário, o alerta é fechado e o sistema se mantém na tela de autenticação.

Figura 4 - Tela de pré-carregamento.



Fonte: Acervo pessoal.

Figura 5 - Tela de autenticação.



Fonte: Acervo pessoal.

Após passar pela etapa de autenticação, o usuário se encontrará na tela de boas-vindas (Figura 6), onde possui a opção de navegar pelas outras funcionalidades da aplicação: página de relatórios, perfil do usuário ou juntar-se a uma sessão utilizando um código fornecido pelo estabelecimento.

Figura 6 - Tela de Boas-vindas.



Fonte: Acervo pessoal.

Seguindo para a página de relatórios (Figura 7), é exibida uma lista com os relatórios disponíveis para o tipo de conta do usuário, ao tocar em alguma das opções, podem ser solicitadas informações para a exibição do relatório como no exemplo demonstrado na Figura 8. Esta página oferece fácil adição de novos relatórios considerando as necessidades dos usuários.

Figura 7 - Tela de relatórios do usuário.



Fonte: Acervo pessoal.

Figura 8 - Tela de inserção de informações.

The screenshot shows a mobile application interface titled "Vendas por Data". At the top, there is a green arrow pointing left and the title. Below the title, there is a green box with the text "Selecione a data". Inside this box, there are three white buttons with the numbers "13", "7", and "2021". Below the green box, there is a green button with the text "Confirmar". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Fonte: Acervo pessoal.

A Figura 9 mostra o resultado de uma consulta do relatório de vendas por data:

Figura 9 – Resultado da consulta.

The screenshot shows a mobile application interface titled "24/5/2021". At the top, there is a green arrow pointing left and the title. Below the title, there is a list of sales statistics, each in a green box with white text:

- Venda Total: R\$ 129.00
- Numero de mesas: 4
- Maior venda do dia: R\$ 65.00
- Menor venda do dia: R\$ 8.00
- Média do dia: R\$ 32.25

The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Fonte: Acervo pessoal.

Caso o usuário siga para a página de perfil (Figura 10), são exibidas informações como seu nome, e-mail e telefone, tocando uma informação é possível a alteração das informações pessoais, conforme mostra a Figura 11, exceto nome e sobrenome, também é possível alterar a senha de acesso.

Figura 10 - Tela perfil do usuário.



Fonte: Acervo pessoal.

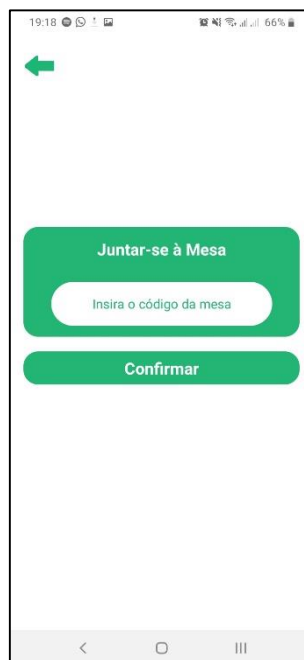
Figura 11 – Tela de alteração de dados do usuário.



Fonte: Acervo pessoal.

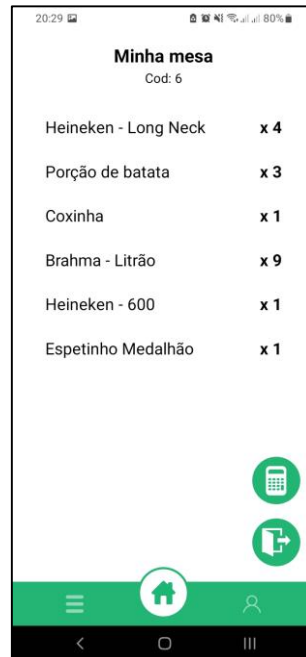
Na tela boas-vindas, é possível encontrar um botão posicionado na parte inferior direita da tela, padrão utilizado por empresas como *Twitter™*, *Google™* e *Facebook™* em seus produtos, devido ao fato de encontrar-se próximo ao polegar do usuário, sendo fácil de encontrar e alcançar. Quando tocado, o sistema exibe uma janela onde é possível digitar um código de uma sessão (Figura 12), a sessão deve ser criada pelo estabelecimento a fim de evitar *spams* e gerar registros inválidos no banco de dados. Utilizando um código inválido, o sistema exibe uma mensagem de erro e permite que o usuário realize uma nova tentativa até que um código válido seja inserido, uma vez que isso ocorra, o sistema retorna à tela de boas-vindas, onde são apresentadas as informações da sessão atual, como é apresentada na Figura 13, a partir desse momento, passam a ser exibidos dois novos botões, um que permite ao usuário sair da sessão e outro que o leva a uma página contendo registros financeiros da sessão (Figura 14), nesta, o sistema recupera o valor total da conta e os valores já quitados e os divide pelo número de consumidores que tenham registrado a sessão em questão. Considerando-se uma pessoa poder estar presente na mesa, mas que não utilize o sistema proposto ou não que não vá realizar o pagamento junto aos outros participantes da sessão, é possível alterar manualmente o número de pessoas utilizado para realizar o cálculo.

Figura 12 - Tela de requerimento do código de sessão.



Fonte: Acervo pessoal.

Figura 13 - Tela de acompanhamento de pedidos.



Fonte: Acervo pessoal.

Figura 14 - Tela de Cálculos.



Fonte: Acervo pessoal.

Caso seja realizado o acesso usando uma conta de prestador de serviços o sistema oferece diferentes possibilidades. Estabelecimentos não podem pular a etapa de cadastro como no caso de pessoas físicas. Assim que é feito o acesso ao sistema

utilizando uma conta cadastrada como prestador de serviços, é possível encontrar no centro da tela inicial (Figura 15) uma lista de mesas ativas, e no canto inferior direito, um botão com o desenho de caixas que direciona o utilizador para a página sua página de produtos (Figura 16) e outro botão com o sinal de adição, que permite ao estabelecimento criar uma nova comanda.

Figura 15 - Tela principal para prestador de serviços.



Fonte: Acervo pessoal.

Figura 16 - Tela de produtos.



Fonte: Acervo pessoal.

Acessando a tela de produtos, é possível cadastrar um novo produto, utilizando o botão com sinal de adição, que redireciona o utilizador ao devido formulário, apresentado na Figura 17, tocando o nome de um produto da lista de produtos cadastrados, o sistema exibe a tela do produto, visível na Figura 18, onde é possível alterar seu preço ou excluir o registro.

Figura 17 - Tela para cadastro de produto.



Fonte: Acervo pessoal.

Figura 18 - Tela para edição de produto.



Fonte: Acervo pessoal.

Pressionando o botão com o sinal de adição na página de boas-vindas, o sistema redireciona o usuário para a criação de uma nova comanda (Figura 19), na tela em questão, é solicitado um apelido opcional para a mesa, caso a informação não

seja fornecida, um registro é criado utilizando a data e hora atuais de acordo com o horário padrão de Brasília, então é exibido um alerta contendo o código da mesa que pode ser compartilhado pelo estabelecimento com seus clientes. Uma vez que uma sessão for criada, o estabelecimento pode acessar os detalhes desta a partir da lista encontrada na tela inicial. Tocando um dos itens na lista, o sistema exibe as respectivas informações de acordo como pode ser visualizado na Figura 20, informações como produtos consumidos e valores financeiros, além de dois botões, um que permite adicionar produtos à comanda e outro que permite realizar quitações.

Figura 19 - Tela para criação de nova comanda.



Fonte: Acervo pessoal.

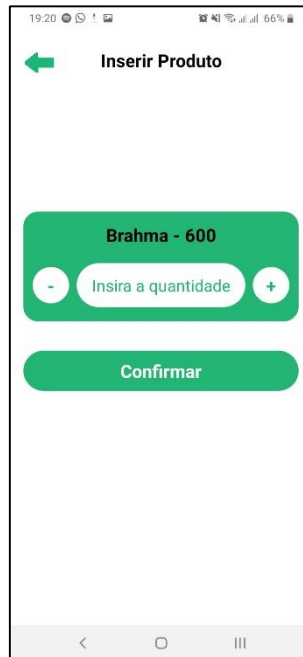
Figura 20 - Tela de detalhes da sessão.



Fonte: Acervo pessoal.

Ao tocar o botão de adição de produtos são exibidos todos os produtos do estabelecimento em ordem alfabética, tocando um dos produtos, é revelada uma tela, seguindo o exemplo mostrado na Figura 21, contendo o nome do produto selecionado e um campo para inserção da quantidade, onde o utilizador pode digitar a quantidade desejada, ou alterar os valores utilizando os botões de subtração e adição presentes que diminuem ou aumentam o valor em uma unidade respectivamente, ao confirmar a ação o sistema retorna à tela de detalhes e os contadores dos produtos têm suas informações atualizadas.

Figura 21 - Tela de inserção de produto.



Fonte: Acervo pessoal.

Caso o estabelecimento deseje realizar um recebimento, é necessário tocar o botão correspondente na tela contendo os detalhes da mesa, assim, o sistema exibe a tela de pagamentos. Nesta tela, o sistema verifica se já existe algum pagamento realizado na mesa, em caso negativo, o sistema oferece ao operador do estabelecimento a possibilidade de realizar um único pagamento no valor total, informando o método de pagamento como pode ser visualizado na Figura 22, o método de pagamento informado pode ser consultado posteriormente no histórico da comanda. Caso seja realizada alguma quitação, o botão então permite a finalização apenas quando o valor total for quitado. Uma vez que a sessão é encerrada, não é mais possível realizar operações nesta.

Figura 22 - Tela de pagamentos.



Fonte: Acervo pessoal.

4. TRABALHOS FUTUROS

Com a conclusão das funcionalidades do sistema poderão ser planejados métodos para remuneração. Dois métodos foram colocados em pauta, o primeiro consiste em exibir anúncios dentro do aplicativo utilizando a ferramenta *AdMob* do Google, que permite a exibição de anúncios de forma remunerada.

Como uma das maiores redes de anúncios do mundo, a *AdMob* trabalha com milhões de anunciantes que competem pelo espaço do seu anúncio. Você mantém controle total do processo, para maximizar os ganhos (Google, 2021).

Ou seja, cada vez que um usuário visualiza uma propaganda, um anunciante realiza um pagamento que é repassado ao responsável pela exibição, esta prática é costumeira em aplicativos para *smartphones* sendo amplamente utilizada para obter remuneração com produtos disponibilizados de forma gratuita. A exibição das propagandas pode ser executada logo após a realização uma ação específica como alterar um dado do perfil, realizar *login* ou ainda após a decorrência de um período específico, mas para a utilização desse método devem ser feitas algumas observações importantes: A exibição de anúncios para estabelecimentos deve ser realizada de forma cautelosa pois pode influenciar negativamente a agilidade das tarefas ou até mesmo inviabilizar a utilização da ferramenta mesmo para as pessoas físicas é necessário ter

cautela quanto a quantidade de anúncios exibidos pois pode se tornar um incômodo (WOEBCKEN, 2019). O segundo método em planejamento procura exibir anúncios aos consumidores dos estabelecimentos vendendo a página principal do sistema como um outdoor digital, negociando os preços diretamente com os anunciantes.

5. CONCLUSÃO

Uma vez que o método de atendimento amplamente utilizado atualmente além de arcaico, fere leis e é considerado abusivo para com os consumidores. A proposição de um sistema prático, gratuito e expansivo aproveita a oportunidade de tentar solucionar de forma simples tal problema, possuindo grande potencial para ser aceito pelo mercado, enfatizado pela crescente adesão dos meios digitais pela sociedade.

Tendo em mente o questionamento realizado na introdução deste trabalho: é possível desenvolver um aplicativo móvel capaz de simplificar o método de atendimento por comandas? Foi desenvolvido um sistema que trabalha com foco nos principais problemas observados durante o atendimento ao cliente, e, portanto, espera-se que seja possível reduzir a ocorrência dos problemas descritos e ainda amenizar seus impactos caso ocorram. Pretende-se, com a utilização do sistema, simplificar a rotina, entregando ao estabelecimento um cadastro de produtos nativo sem a necessidade de outro sistema e também entrega aos consumidores um registro financeiro da sua comanda sendo capaz de entregar vantagens aos usuários devido a execução de tarefas como o arquivamento das comandas, a realização deste por exemplo, requer que o estabelecimento tenha espaço físico disponível sendo um trabalho moroso e não proporciona a devida acessibilidade quando as informações são necessárias, estando ainda sujeito à danificação devido acidentes ou ocorrência de desastres naturais como enchentes. Essa tarefa é ainda mais incômoda para os consumidores dos estabelecimentos e tende ao desuso mesmo que as informações armazenadas sejam importantes, porém, com a utilização do aplicativo esta tarefa seria realizada automaticamente pelo sistema, e não seria necessário ocupar espaço físico.

4. REFERÊNCIAS

Async Storage. Versão 1.14.1. React Native Community, 2021. Disponível em: <<https://react-native-async-storage.github.io/async-storage/docs/usage/>>. Acesso em: 03 de março de 2021.

ABDALA, Daniel. **Diagrama de Estados**. 3f. Universidade Federal de Uberlândia, Minas Gerais. 2011.

BOEHM, Barry. **Spiral Development: Experience, Principles, and Refinements**. 2000. 49 f. University of Southern California, Califórnia, 2000.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **The Unified Modeling Language User Guide**. 2. ed. Addison-Wesley, 2005. 496 p.

BRASIL. Lei nº. 8078 de 11 de setembro de 1990. **Código de Defesa do Consumidor**. Dispõe sobre a proteção do consumidor e dá outras providências. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/l8078compilado.htm

CONVERSE, Tim; PARK, Joyce. **PHP: a bíblia**. Gulf Professional Publishing, 2003.

COSTA, Thomas e OLIVEIRA, Rodrigo. **Planejamento E Controle Das Operações De Produção E Atendimento Ao Cliente: Uma Análise Em Um Quiosque De Praia**. 2018. 20 f. Universidade Federal do Espírito Santo. Espírito Santo. 2018.

CROCKFORD, Douglas. **Introducing JSON**. 1999. Disponível em: <https://www.json.org/json-en.html>. Acesso em 27 de junho de 2021.

CURSO de UML – O que é um Diagrama de Sequência. Produção: Fábio dos Reis. Brasil: Bóson Treinamentos. 2019. Disponível em: https://www.youtube.com/watch?v=UVkj3ed0ZuM&ab_channel=B%C3%B3sonTreinamentos. Acesso em: 04 de agosto de 2021.

Expo™. Versão 4.4.3. Expo™, 2021. Disponível em: <https://expo.io/>. Acesso em: 27 de junho de 2021.

Facebook™. **React: A JavaScript library for building user interfaces**. 2021. Disponível em: <https://reactjs.org/>. Acesso em 27 de junho de 2021.

FacebookTM. **React Native: Learn once write anywhere**. 2021. Disponível em: <<https://reactnative.dev/>>. Acesso em 19 de julho de 2021.

FERNANDES, Diego. **Expo: o que é, para que serve e quando utilizar?**. 2018. Disponível em: <<https://blog.rocketseat.com.br/expo-react-native/>>. Acesso em 02 de julho de 2021.

GoogleTM. **Google AdMob: Ganhe mais com seus apps**. 2021. Disponível em: <https://admob.google.com/intl/pt-BR_br/home/admob-advantage/>. Acesso em 02 de julho de 2021.

HELLER, Eva. **A Psicologia das Cores**. São Paulo: LeLivros, 2007. 541p.

KRUG, Steve. **Não me faça pensar**. 3. ed. Rio de Janeiro: Alta Books, 2014. 212 p.

IBMTM. **Diagramas de Atividades**. [s.d.]. Disponível em: <<https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-activity>>. Acesso em 04 de agosto de 2021.

IBMTM. **Diagramas de Componentes**. [s.d.]. Disponível em: <<https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=services-component-diagrams>>. Acesso em 04 de agosto de 2021.

IBMTM. **Diagramas de Objetos**. [s.d.]. Disponível em: <<https://www.ibm.com/docs/pt-br/rsar/9.5?topic=diagrams-object>>. Acesso em 04 de agosto de 2021.

MAIA, Flávia. **Perdeu a comanda? Não se preocupe: a responsabilidade é da empresa**. 2018. Disponível em: <<https://blogs.correiobraziliense.com.br/consumidor/perdeu-a-comanda-nao-se-preocupe-a-responsabilidade-e-da-empresa/>>. Acesso em: 27 de junho de 2021.

MAGILL, Frank. **The 20th Century Go-N: Dictionary of World Biography, Volume 8**. Routledge, 1999. 2799 p.

PINHO, Diego e ESCUDELARIO, Bruna. **React Native: Desenvolvimento de aplicativos mobile com React**. Brasil, Casa do Código, 2020.

Procon. **Perda da Comanda – Saiba seus Direitos**. Disponível em: <<http://www.procon.pr.gov.br/modules/conteudo/conteudo.php?conteudo=565>>.

Acesso em: 16 de junho de 2021.

React-native-paper. Versão 4.7.2. URBANIAK, Dawid *et al.*, 2021. Disponível em: <<https://callstack.github.io/react-native-paper/>>. Acesso em: 21 de março de 2021.

React-native-svg. Versão 11.0.1. React Native Community, 2021. Disponível em: <<https://github.com/react-native-svg/react-native-svg>>. Acesso em: 03 de março de 2021.

React Navigation. Versão 5.x. Expo™, 2021. Disponível em: <<https://reactnavigation.org/>>. Acesso em: 14 de fevereiro de 2021.

RIBEIRO, Leandro. **O que é UML e diagramas de Caso de Uso: Introdução Prática à UML**. 2012. Disponível em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em: 04 de agosto de 2021.

RIVEST, Ronald. **The MD5 Message-Digest Algorithm**. 1992. Disponível em: <<https://datatracker.ietf.org/doc/html/rfc1321>>. Acesso em: 27 de junho de 2021.

ROTTER, Laila. **LESS IS MORE: Interseções entre o surgimento do design gráfico moderno e o minimalismo de Mies van der Rohe**. 2014. 11f.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript**. 2015. Disponível em: <<https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>>. Acesso em: 27 de junho de 2021.

Styled Components. Versão 5.2.1. React Native Community, 2021. Disponível em: <<https://styled-components.com/>>. Acesso em: 03 de março de 2021.

TYBEL, Douglas. **Orientações básicas na elaboração de um diagrama de classes**. 2016. Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 04 de agosto de 2021

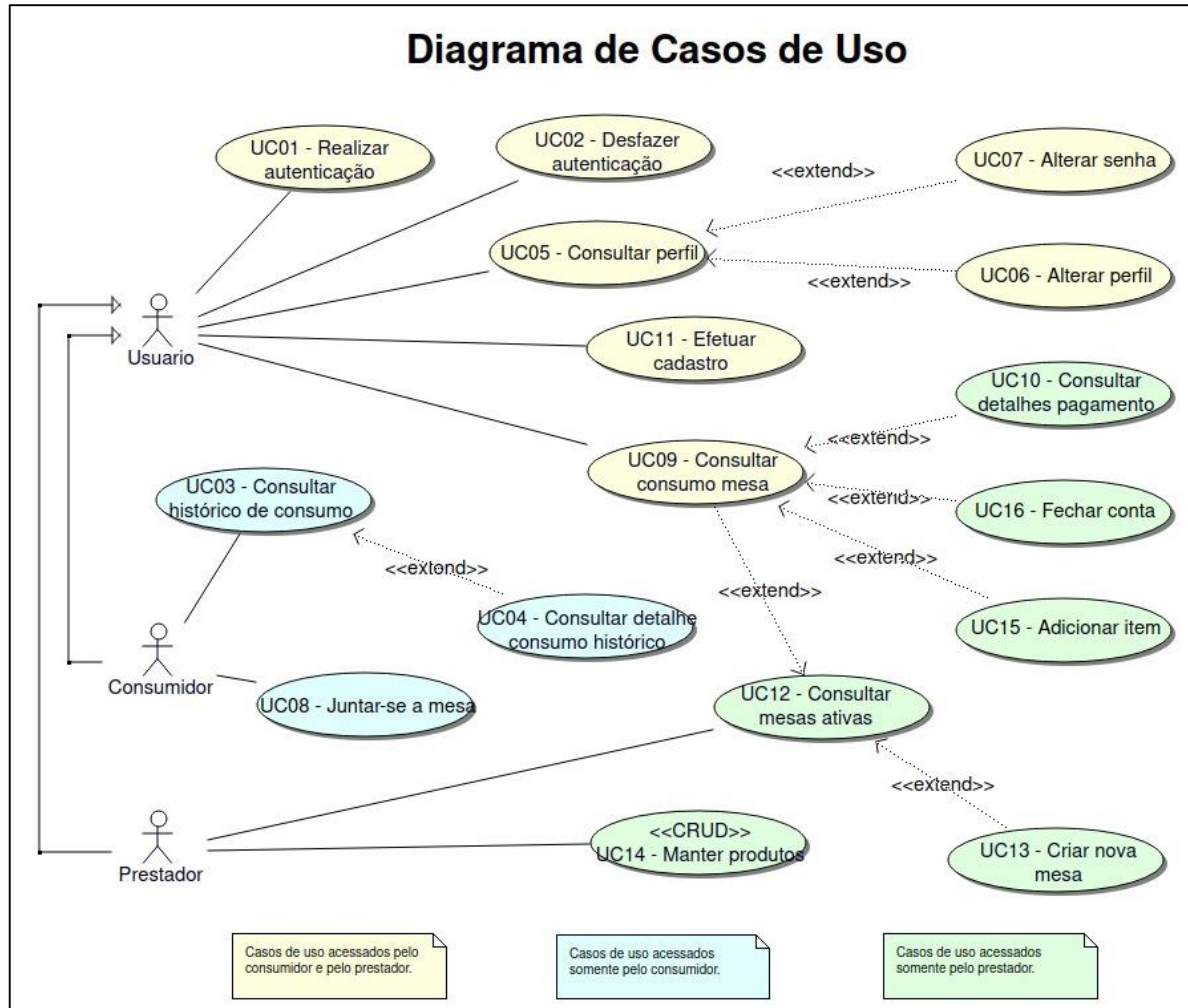
VivaDecoraPro. **O mestre do minimalismo na arquitetura: conheça a vida e obra de Mies van de Rohe**. 2017. Disponível em: <<https://www.vivadecora.com.br/pro/arquitetos/mies-van-der-rohe/>>. Acesso em 30 de julho de 2021

W3Schools™. **SVG Tutorial**. 2021. Disponível em: <https://www.w3schools.com/graphics/svg_intro.asp>. Acesso em 27 de junho de 2021.

WOEBCKEN, Caio. **Por que os usuários odeiam anúncios online e como sua empresa pode contornar essa situação**. 2019. Disponível em: <<https://rockcontent.com/br/blog/anunciar-para-quem-odeia-anuncios/>>. Acesso em: 03 de agosto de 2021.

Apêndice A – Diagramas UML

Figura 23 - Diagrama de casos de uso.



Fonte: Acervo pessoal.

A Figura 23 traz um diagrama de casos de uso apresentado nos padrões da UML (*Unified Modeling Language*), linguagem criada com o objetivo de unificar os vários métodos de se apresentar graficamente um sistema (BOOCH et al, 2005).

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz (RIBEIRO, 2012).

O diagrama vem seguido também de especificações de casos de uso, como a seguir:

Objetivo: Realizar o gerenciamento das comandas sem a necessidade de outras ferramentas.

Atores: Atendente e Consumidor.

Cenário principal:

1. O Consumidor se dirige ao estabelecimento.
2. O Atendente cria uma nova comanda.
3. O sistema insere uma nova comanda no baco de dados.
4. O Atendente informa o código da comanda ao Consumidor.
5. O Consumidor insere o código no sistema.
6. O sistema recupera as informações da comanda no banco de dados.
7. O Consumidor realiza um pedido.
8. O atendente insere o pedido no sistema.
9. O sistema grava o pedido no banco de dados.
10. O Consumidor realiza o pagamento total da comanda.
11. O Atendente registra o pagamento no sistema.
12. O sistema grava o pagamento no banco de dados.
13. O Atendendente encerra a comanda.
14. O sistema cria registros de histórico utilizando as informações da comanda.

Cenários Alternativo

1. O Consumidor realizou o pagamento parcial da comanda.

Caso afirmativo a comanda não pode ser encerrada, aguardando o pagamento do valor restante.
2. O Consumidor não inseriu o código da comanda no sistema.

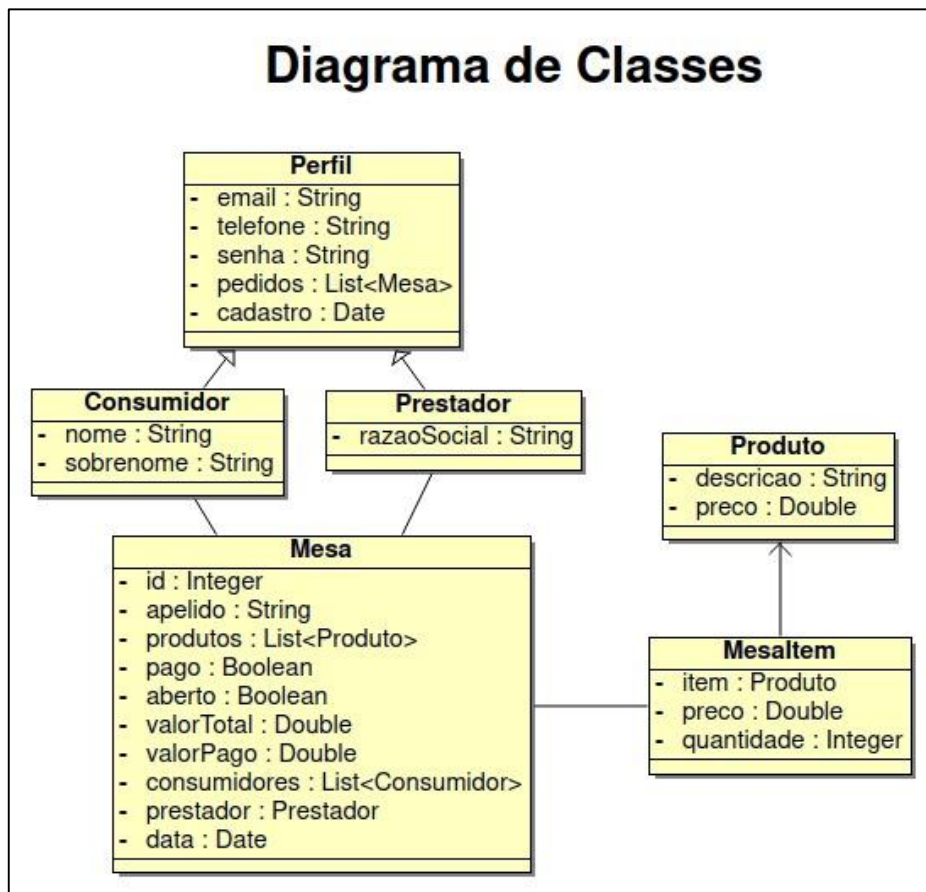
Caso afirmativo, o sistema cria um registro de histórico apenas para o Atendente.
3. O Consumidor inseriu um código diferente do que foi fornecido pelo Atendente.

Caso a comanda exista, o sistema exibe informações da comanda, caso o código seja inválido o sistema exibe um alerta e não registra o código.

Requisitos Especiais

1. Os usuários necessitam de *smartphones* e acesso à internet.

Figura 24 - Diagrama de classes.



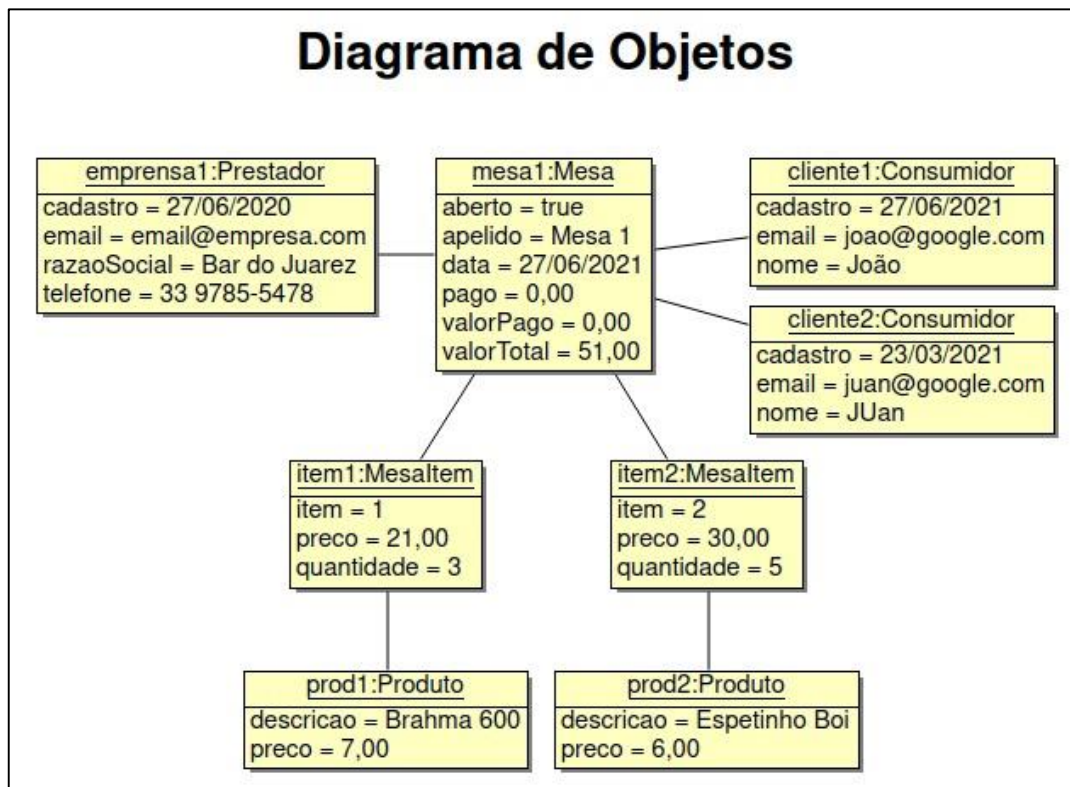
Fonte: Acervo pessoal.

A Figura 24 traz um diagrama de classes apresentado nos padrões da UML.

Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes (TYBEL, 2016).

Sendo assim, foram agrupados os objetos do sistema e catalogados junto de suas propriedades.

Figura 25 - Diagrama de objetos.



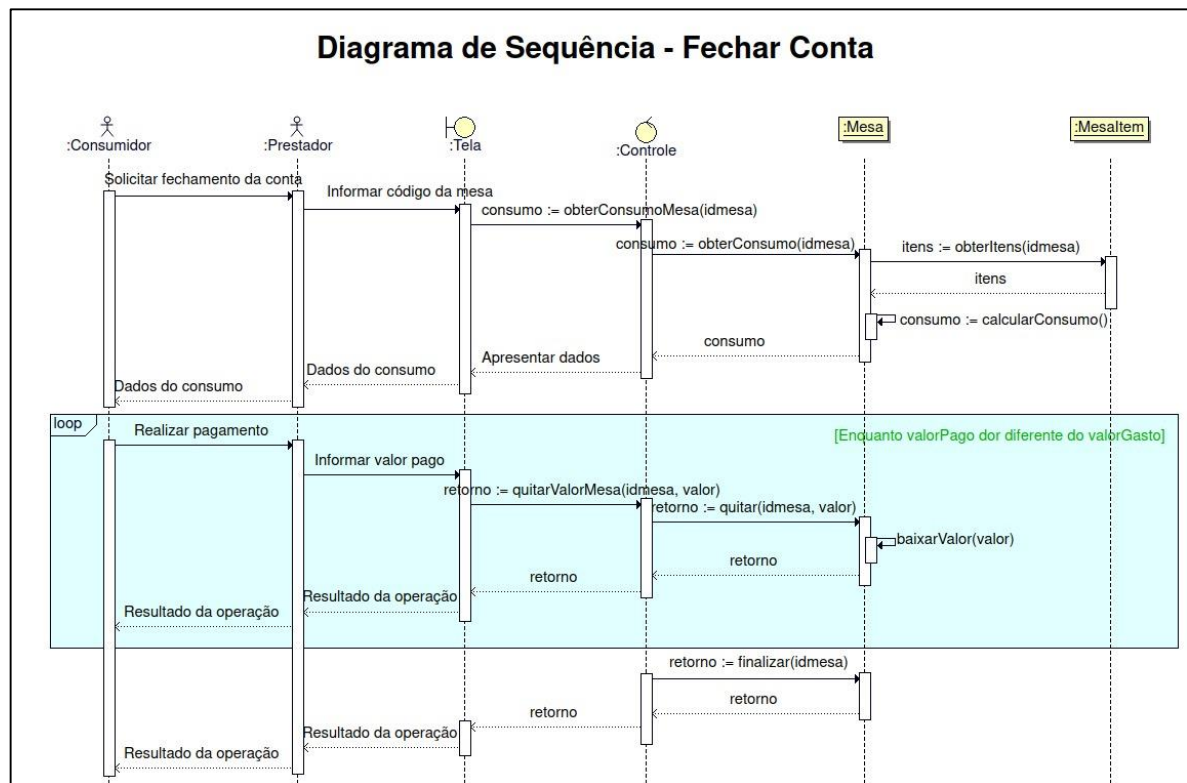
Fonte: Acervo pessoal.

A Figura 25 traz um diagrama de objetos apresentado nos padrões da UML.

Na UML, os diagramas de objetos fornecem uma captura instantânea das instâncias em um sistema e os relacionamentos entre as instâncias. Inicializando os elementos de modelos em um diagrama de classe, você pode explorar o comportamento de um sistema em um determinado momento (IBMTM, [s.d.]).

Seguindo a descrição apresentada, foram apresentadas informações de um determinado momento hipotético da execução do sistema, simulando uma comanda utilizada por dois consumidores que havia pedido ao total 3 unidades do item 1, cuja descrição é “Brahma 600” e 5 unidades do item 2, cuja descrição é “Espetinho Boi”.

Figura 26 - Diagrama de sequência.



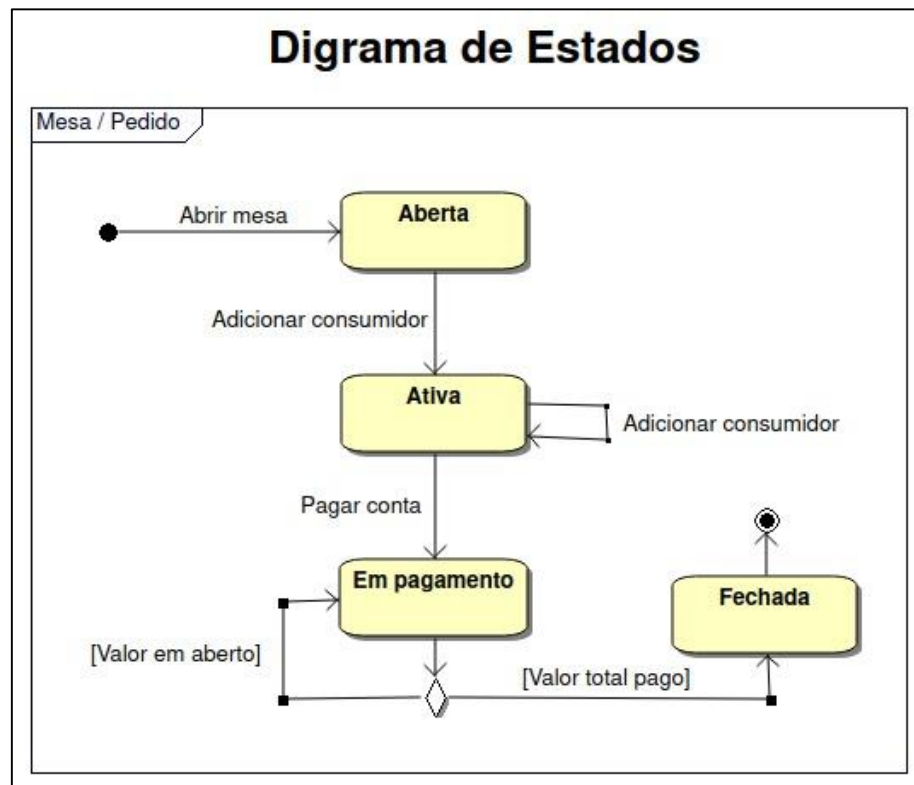
Fonte: Acervo pessoal.

A Figura 26 traz um diagrama de sequência apresentado nos padrões da UML.

Diagrama que enfatiza a ordenação temporal das mensagens. No geral, se baseia em um caso de uso definido pelo diagrama de mesmo nome, e usa o diagrama de classes para definir os objetos envolvidos no processo (REIS, 2019).

No diagrama apresentado é demonstrado um processo a partir do momento que um consumidor deseja realizar o pagamento da comanda, demonstrando que o encerramento desta só ocorre quando o valor total é quitado.

Figura 27 - Diagrama de estados.



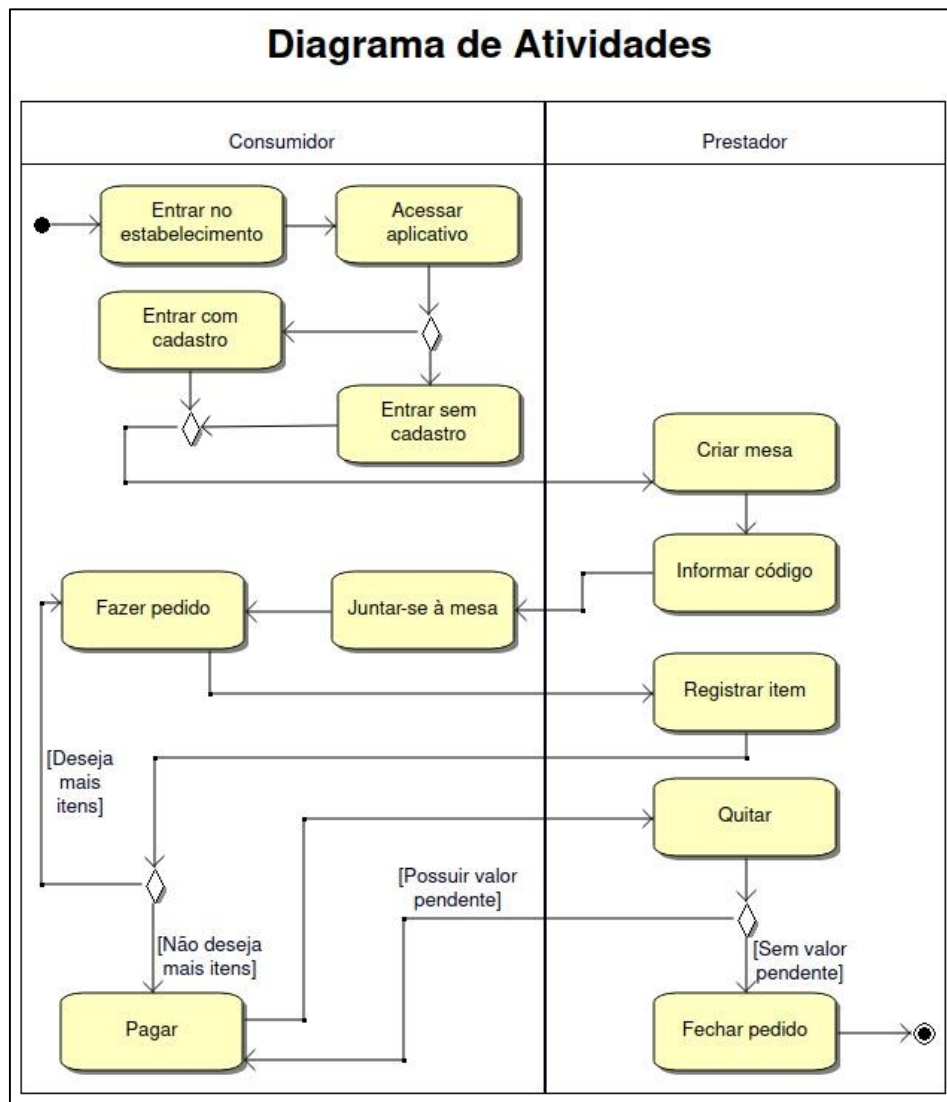
Fonte: Acervo pessoal.

A Figura 27 traz um diagrama de estados apresentado nos padrões da UML.

UML prevê um diagrama específico para modelar os diversos estados de um objeto durante o seu ciclo de vida. Tal diagrama é chamado de diagrama de estados. Ele é muito utilizado na área de eletrônica digital assim como em linguagens formais. Foi importado pela UML por ser uma maneira eficiente e clara de se descrever todos os possíveis estados de um sistema assim como quais eventos levam a transição de um estado para outro (ABDALA, 2011).

No diagrama apresentado, são mostrados os estados de uma comanda no sistema, sendo afetada por uma sequência de eventos que terminam por encerrar a comanda quando é pago o valor total.

Figura 28 - Diagrama de atividades.



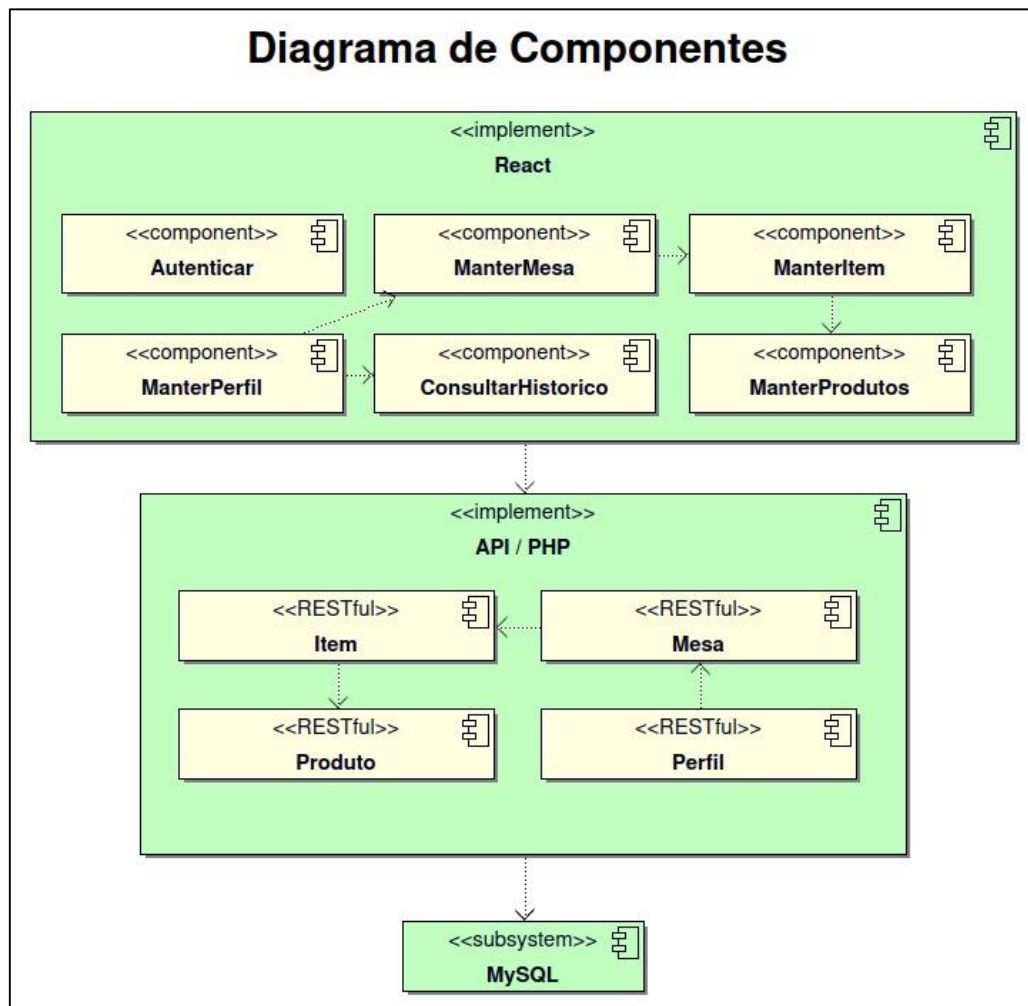
Fonte: Acervo pessoal.

A Figura 28 traz um diagrama de atividades apresentado nos padrões da UML.

Na UML, um diagrama de atividade fornece uma visualização do comportamento de um sistema descrevendo a seqüência de ações em um processo. Os diagramas de atividades são semelhantes a fluxogramas porque mostram o fluxo entre as ações em uma atividade; no entanto, os diagramas de atividades também podem mostrar fluxos paralelos ou simultâneos e fluxos alternativos (IBMTM, [s.d.]).

No diagrama apresentado pode ser visualizado o atendimento de um cliente por um funcionário do estabelecimento desde o momento em que este se locomove até o comércio, até o momento em que a conta é encerrada, o diagrama mostra ainda como suas ações interagem e ainda processos de tomada de decisões.

Figura 29 - Diagrama de componentes.



Fonte: Acervo pessoal.

A Figura 29 traz um diagrama de componentes apresentado nos padrões da UML.

Na UML, os diagramas de componentes mostram a estrutura do sistema de software, que descreve os componentes do software, suas interfaces e suas dependências. É possível utilizar diagramas de componentes para modelar sistemas de software em um alto nível ou para mostrar componentes em um nível de pacote mais baixo (IBMTM, [s.d.]).

São apresentados os arquivos necessários para que o sistema realize suas tarefas, ilustrando a dependência que o aplicativo em React tem de um sistema gerenciador de banco de dados, neste caso o MySQL e os arquivos responsáveis por realizar a comunicação.