

GETEASY:

Prototipação de Aplicativo auxiliar em representação comercial

Alexandre Badoco Júnior
Graduando em Sistemas de Informação – Uni-FACEF
Alexandre.badoco@hotmail.com

Ely Fernando do Prado
Mestre em Computação – Uni-FACEF
elyfprado@gmail.com

Resumo

O trabalho de representação comercial tem uma demanda organizacional difícil de se manter e o autor deste trabalho vive o problema e percebe a grande dificuldade e a demanda de serviço manual para finalização dessa tarefa. Com isso foi realizado o levantamento e prototipação de uma solução utilizando das tecnologias atuais em aplicativos móvel, que possibilitará a criação de uma ferramenta para cadastrar e gerenciar pedidos, clientes e produtos. O software beneficiará as empresas que utilizam de representantes para o setor de vendas, agilizando os processos de cadastro e distribuição dos pedidos. A motivação técnica é prototipar tecnologias que possam apoiar uma solução computacional baseada na engenharia de software para resolver o problema exposto.

Palavras-chave: Sistemas. Diagrama. Aplicativos. Prototipação.

Abstract

The work of commercial representation has an organizational demand that is difficult to maintain and the author of this work lives the problem and realizes the great difficulty and demand for manual service to complete this task. With this, a survey and prototyping of a solution was carried out using current technologies in mobile applications, which will enable the creation of a tool to register and manage orders, customers and products. The software will benefit companies that use representatives for the sales sector, streamlining the registration and distribution of orders. The technical motivation is to develop technologies that can support a computational solution based on software engineering to solve the exposed problem.

Keywords: Information system. Sales. Applications. Prototyping.

1 INTRODUÇÃO

É perceptível que nos tempos atuais ter uma gestão dos dados é sinônimo de sucesso e crescimento, com isso a utilização de softwares cresceu exponencialmente. Todos os dias, pequenos e grandes comerciantes perdem vendas por falta de gerenciamento ou por falta de uma solução simples que possa possibilitar o registro e gerenciamentos de suas vendas.

O objetivo deste trabalho é fazer o levantamento dos requisitos e prototipação de uma solução leve e simples para dispositivos móveis que será usadas para cadastrar clientes e seus pedidos e organizar e distribuir os pedidos para os setores correspondentes.

Para o desenvolvimento deste projeto foram feitas entrevistas com potenciais usuários para levantar os requisitos mínimos para projetar o *Business Process Model and Notation* (BPMN) e *Unified Modeling Language* (UML), projetar o modelo de negócio CANVAS e prototipar a interface utilizando técnicas de rabisco frame.

Para a elaboração da prototipação do aplicativo foram utilizadas as ferramentas de edição de imagens Adobe Photoshop e Corel Draw, e a ferramenta de planilhas Microsoft Office Excel e para a prototipação de diagramas foram utilizados a plataforma online *Lucidchart* e o aplicativo *DBDesigner*.

2 Referencial Teórico

Foi utilizado como referencial teórico, artigos e livros sobre interação homem-computador (IHC) que possibilitou um melhor aprendizado durante o desenvolvimento desse artigo. Por se tratar de tecnologias atuais foram utilizados informações e documentos dos sites de seus desenvolvedores.

2.1 Dispositivo móvel

A evolução do hardware dos dispositivos móveis permitiu aos usuários a possibilidade de muitas outras funções em seus aparelhos móveis do que apenas o envio de mensagens e ligações. Com a melhoria do hardware foi possível utilizar sistemas operacionais mais complexos, e isso permitiu o desenvolvimento de aplicações com maior complexidade permitindo a utilização de sensores e outros *gadgets*. Devido a esta evolução um aparelho celular se transformou em uma oportunidade de entretenimento e fonte de informações em tempo real, que podem ser utilizados como ferramentas para solução de problemas (SARTORELI. KUCHAUSKI. 2013).

O mercado de aplicativos atual é disputado por diversas plataformas de desenvolvimento incluindo sistemas operacionais e fábricas de softwares, gerando uma variedade de soluções no mercado. Desenvolver aplicativos em um ambiente fragmentado e em rápido crescimento e selecionar qual plataforma que serão desenvolvidos os aplicativos são alguns dos maiores desafios para o desenvolvedor.

Sabendo que as maiores plataformas são Android e IOS que são criadas de formas diferentes e com funcionalidades diferentes (MARTINS, 2013).

2.1 Sistema Android

O sistema operacional em dispositivos móveis é considerado o programa principal que administra todas as funções de maneira eficiente e sem interrupções de forma que o usuário possa manter conectado os recursos do hardware (POLANCO; TAIBO, 2011). O sistema Android é um sistema operacional baseado em Linux que foi adquirido pelo Google e *Open Handset Alliance*, com a finalidade de satisfazer as necessidades de operadores e fabricantes de *smartphones*, além de promover o desenvolvimento de aplicativos com qualidade diferente de outros sistemas operacionais (GOOGLE, 2021).

Mesmo sendo construído com base Linux o sistema Android não tem os componentes essenciais como o *Windows system* e não contém conjuntos de padrões apresentados em algumas versões do Linux (PEREIRA, 2009). O nível de segurança do sistema operacional levanta dúvidas por ser *open source*, entretanto ele tem padrões de segurança que impedem que *hackers* possam criar versões maliciosas e implantá-los nos celulares. Tendo como base o Linux, toda versão do Android instalada é criado um usuário diferente assim impedindo que um usuário acesse o outro. A estrutura do Android para construção de aplicativos requer que todo o processo que exija componentes externos, sendo físicos ou digitais, solicite uma verificação de segurança que o próprio usuário deve aceitar.

Para iniciar o desenvolvimento com Android é necessário instalar o *Software Development Kit* (SDK) que contém as bibliotecas, e o simulador de dispositivo móvel que inicia uma máquina virtual Android com suas ferramentas utilitárias e *Application Programming Interface* (API) completa da linguagem JAVA. O Eclipse é um software de bloco de texto que é mais bem recomendado para a codificação em Android pois ele se integra com o ambiente de desenvolvimento SDK. É possível testar os apps diretamente no smartphone apenas plugando-o via USB e habilitando a depuração via *Universal Serial Bus* USB, assim exige menos desempenho do computador, porém pode consumir muita memória de seu *smartphones*. (LECHETA, 2010)

2.2 Sistema IOS

O *iPhone OS* (iOS) é o sistema operacional móvel da Apple, criado diretamente para dispositivos da Apple como o iPhone, iPod, iPad, e Apple Tv. Em 2007 Steve Jobs apresentou a todos o primeiro dispositivo móvel com o sistema iOS, e teve como uma de suas principais características a impossibilidade de utilizar o SO em aparelhos de terceiros, isto é, ele é encontrado somente em aparelhos da própria marca. Lançado com tela *multitouch* e inúmeras outras funções (SARTORELI. KUCHAUSKI. 2013). Em seu lançamento Steve Jobs criador da marca disse: “As pessoas não sabem o que querem, até mostrarmos a elas”.

A exclusividade do IOS, aumenta a dificuldade da criação de aplicativos mal-intencionados junto à sua constante atualizações que é atualizados os padrões de segurança. O desenvolvimento para IOS tem suas dificuldades pois as bibliotecas só funcionam em aparelhos que utiliza IOS e frequentemente é preciso criar códigos exclusivos. Para publicar na *AppleStore* a empresa deve emitir um certificado de segurança, com isso todos os aplicativos desenvolvidos tem que passar por uma vistoria e seus requerimentos de segurança, a empresa também possui uma cadeia

de inicialização segura garantindo que apenas os códigos permitidos possam ser executados.

Jailbreaks é a maior de suas preocupações pois consistem em um vírus que invade por meio dos e-mails, que pode afetar dependendo da versão do OS até o *Kernal* do dispositivo, todas as atualizações lançadas pela Apple têm a descrição dizendo que houve melhoras na segurança contra *Jailbreaks* (Matos, ET Al. 2020).

2.3 React Native

React Native é um *framework* mobile nativo com base em *Java Script*, que possui uma estrutura multiplataforma que permite o desenvolvimento para *Android* e *IOS* de forma simples e rápida. Por ser uma linguagem nativa e multiplataforma, é possível escrever códigos que são compilados para as duas plataformas, dessa forma não é necessário desenvolver dois códigos diferentes. Os testes de desempenho feitos segundo (DANIELSSON, 2016), foram surpreendentemente positivos para o *React Native*, pois alguns usuários não conseguiram diferenciar entre os dois aplicativos criados com *Android* ou *React Native*.

Um aplicativo híbrido é desenvolvido usando um componente chamado *webview* e contem acesso a APIs nativas. Isso resultará em um aplicativo móvel que tem que ser instalado no dispositivo. O aplicativo terá restrições iguais a web, porém, a maior parte é construída usando uma *webview*, que é essencialmente um componente nativo que simula um navegador. O aplicativo é construído usando técnicas da web, com a capacidade de chamar APIs nativas usando *Java Script*, que utiliza das APIs para obter acesso à câmera, GPS e outros componentes de *hardware* (HANSSON; VIDHALL, 2016).

Existem comunidades de desenvolvedores para essa linguagem, como a *EXPO* ou *Rocketseat* que são as comunidades mais utilizadas pelos desenvolvedores, e ambas têm cursos de especializações. A *EXPO* desenvolveu um ambiente de emulação para desenvolvimento em *React Native* tendo uma seção para codificar e outra para emular um dispositivo móvel, além de possuir um aplicativo que permite emular o app sem precisar de instalação direto no smartphone (EXPO, 2021). A Figura 1 mostra a plataforma EXPO.IO

Figura 1 -Montagem a partir de imagens coletadas no site expo.io, via Print Screen do Windows 10.



Fonte: O Autor.

3 Empreendedorismo

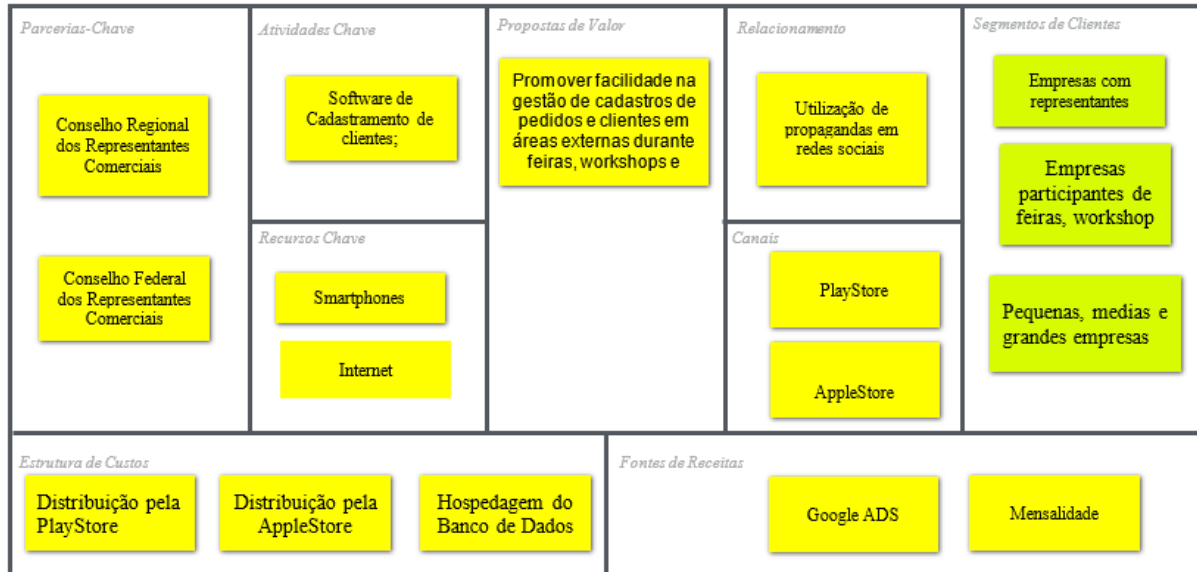
Antes de analisar o projeto em si, é interessante que se pense sobre o negócio e a viabilidade do projeto em um contexto mercadológico. Portanto, pode-se analisar o planejando do projeto como negócio através de um Modelo *Canvas Bussiness*.

3.1 Canvas Bussiness

O modelo *Canvas Business* junta todos os critérios para ser considerado um projeto de uma maneira simples e fácil para começar a trabalhar. Não é necessário ser um profissional de engenharia de software ou um gerente para entendê-lo e ser capaz de usá-lo, com o conceito simples é possível por em prática seus metodos (OSTERWALDER, PIGNEUR. 2011).

Na figura 2 podemos observar o modelo CANVAS desenvolvido para o protótipo descrito neste artigo.

Figura 2 - Prototipação do esquema Canvas.



Fonte: O Autor.

É mostrado na figura 2 que o projeto tem o apoio indireto do conselho regional dos representantes comerciais e o conselho Federal dos representantes comerciais, que fiscalizam as ferramentas e condições de trabalhos dos representantes comerciais.

4 DESENVOLVIMENTO

Nesta seção são apresentados os processos de análise de requisitos e desenvolvimento do protótipo do projeto, apresentando três tipos de diagramas BPMN, Caso de Uso e UML.

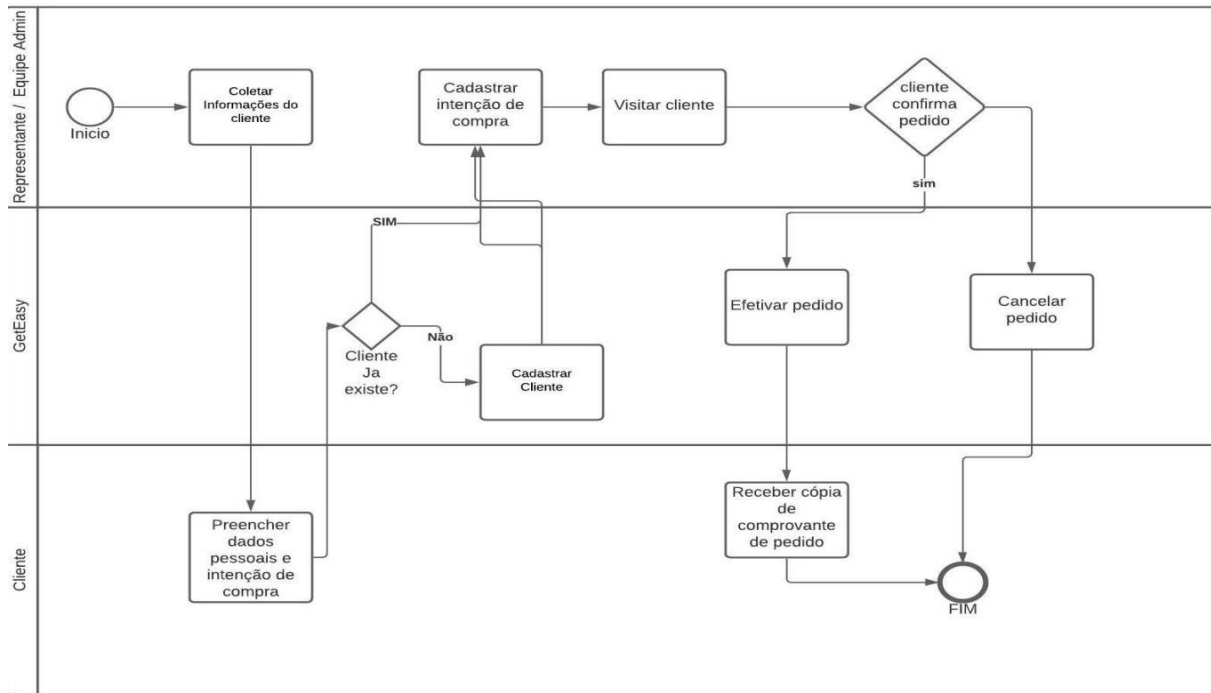
4.1 BPMN

A iniciativa de Gerenciamento de Processos de Negócios (BPMN) criou um processo de negócios padronizados. Como o objetivo principal do BPMN é fornecer um padrão de estrutura que fosse facilmente compreensível por todos usuários de negócios, desde os analistas de negócios que criam os rascunhos iniciais dos processos, até os desenvolvedores responsáveis pela implementação da tecnologia.

BPMN define um Diagrama de Processo de Negócios (BPD), que é baseado em uma técnica de fluxograma adaptado para a criação de modelos gráficos de operações do modelo de negócios (WHITE, 2004).

A figura 3 é o BPMN desenvolvido para o protótipo descrito neste artigo, mostrando o processo referente ao cadastro do cliente e do pedido. É possível observar que o usuário coleta os dados do cliente e é feita a verificação automatizada, indicando se o cliente já existe, caso exista o processo continua para tela de cadastro do pedido, nesse ponto os produtos devem já estar cadastrados pelo usuário ou de forma automatizada por meio de APIs. Após o envio do pedido o cliente recebe um e-mail com um link de confirmação e com o resumo da venda.

Figura 3 – BPMN criado pelo autor deste trabalho.



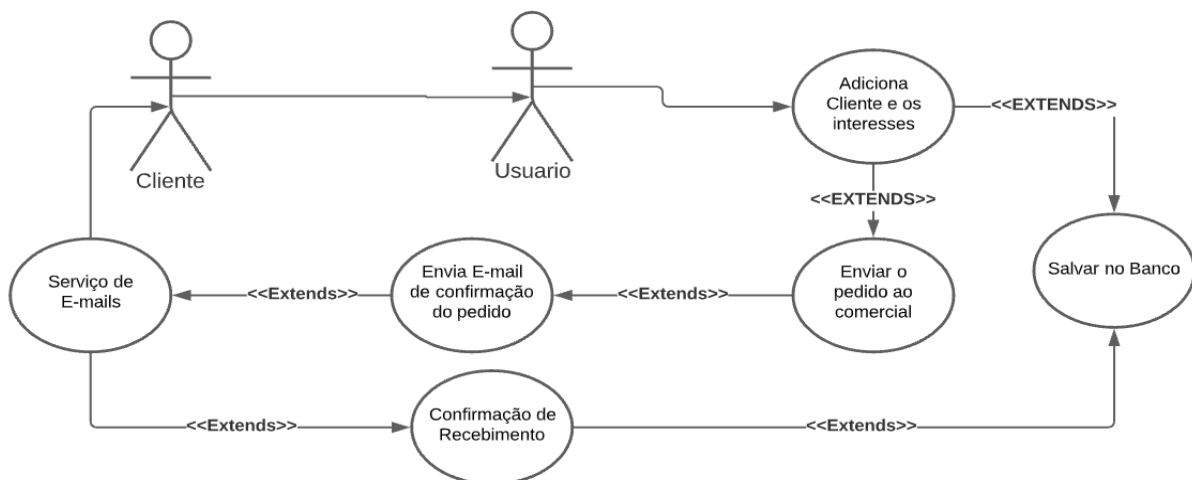
Fonte: O Autor .

4.2 Diagrama de casos de uso

O diagrama de caso de uso (*Use Case*) é um mapa de processos exclusivo, usado para modelar e demonstrar a forma de que o processos irão funcionar. O diagrama apresenta os atores relevantes e os serviços exigidos pelo sistema para ter o funcionamento eficaz (PENDER, 2004).

A Figura 4 apresenta o diagrama de caso de uso de forma a entender o processo mostrado na figura 3, identificando quando o processo deve ser executado pelo usuário ou pelo software.

Figura 4 – Diagrama de caso de uso;



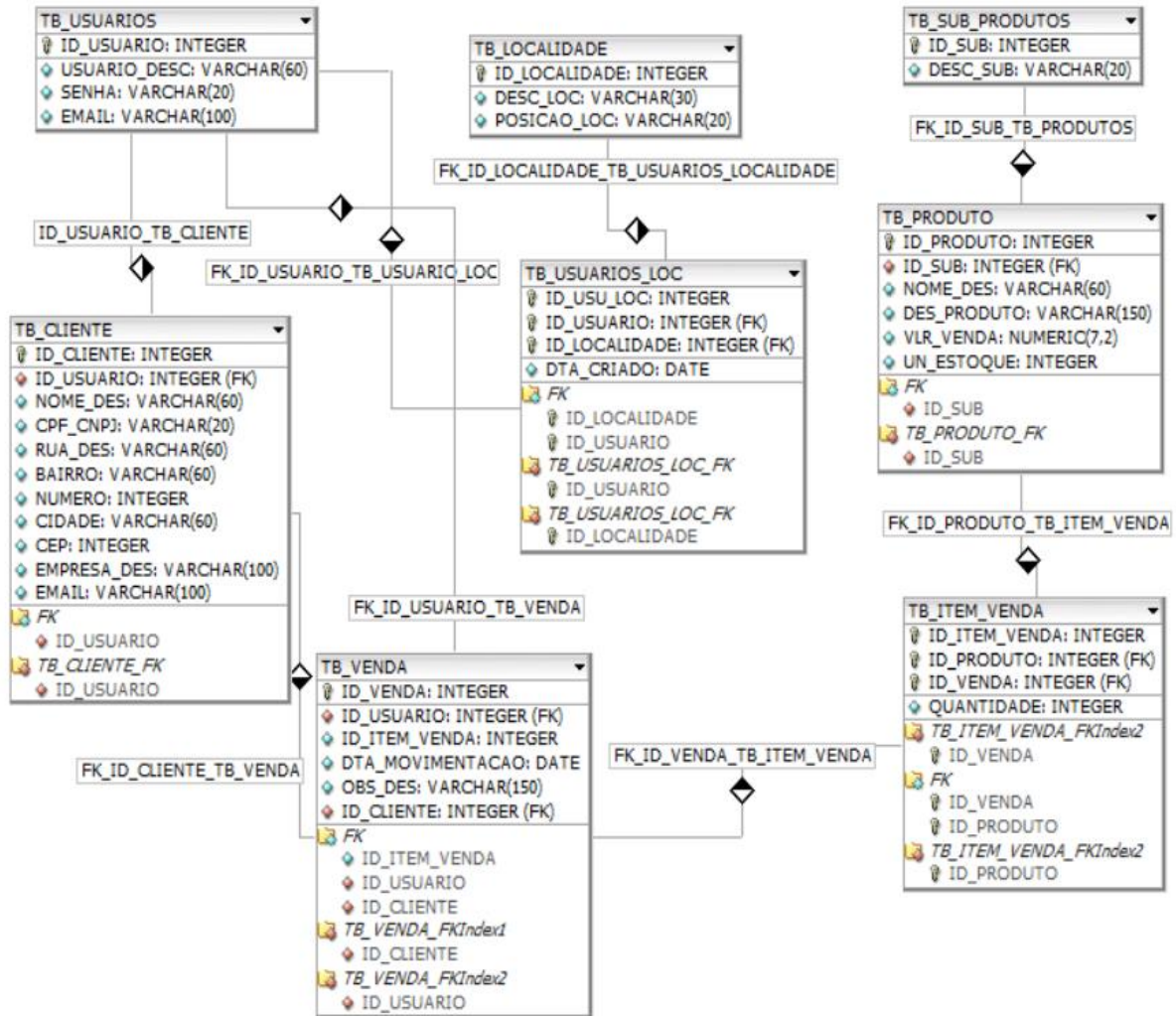
Fonte: O Autor.

4.3 Diagrama de classes

A *Unified Modeling Language* (UML) é uma linguagem universal para a elaboração da estrutura de projetos de software. Entre os diagramas da UML mais importantes, podemos citar o diagrama de classes, o qual é possível ter uma fácil visualização das tabelas de um banco de dados e documentar os artefatos que utilizam de sistemas complexos (BOOCH, 2005).

É possível observar na Figura 5 que os registros possuem rastreabilidade de cadastros por usuário através da tupla ID_USUARIO possibilitando qual usuário realizou a transação. O diagrama foi ajustado para que possa funcionar na terceira forma normal (3FN). Para que as tabelas estejam na 3FN há algumas regras, todos os atributos dessa tabela devem estar na 1FN, 2FN e ser funcionalmente independentes uns dos outros, ao mesmo tempo que devem ser dependentes exclusivamente da chave primária da tabela.

Figura 5 – diagrama de classes



Fonte: O Autor.

4.4 Prototipação

O processo de desenvolvimento de um software envolve várias camadas, não tendo a importância do estado que esteja o seu ciclo de vida adotado. Para todo software desenvolvido é importante afirmar que houve os processos de validação e gerenciamento de requisitos prontos, que são peças chaves para o sucesso. A prototipação é um dos métodos de validação que mais se utiliza no mercado atual, pois é possível mostrar ao usuário como será o software e pode ser utilizado como documento de validação (FERREIRA, 2003).

Segundo (SANCHES DA SILVA, 2010), a prototipação em projetos com contato constante do cliente, se torna inviável o desenvolvimento com isso a prototipação consegue amenizar os custos de produção, Permitindo que o cliente consiga aprovar ou modificar o projeto antes de elevar os custos com desenvolvedores e ferramentas.

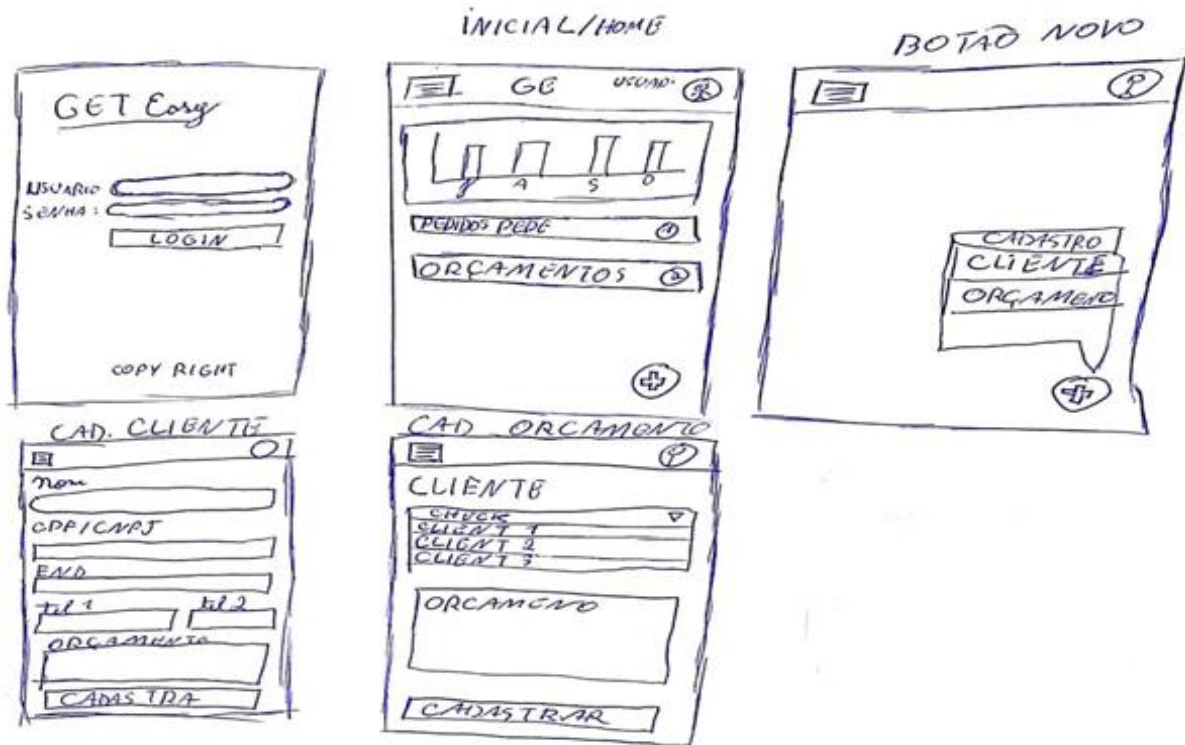
Segundo (SOMMERVILLE, 2003) a definição de prototipação é como uma versão inicial de um software mostrando seus conceitos e funcionalidades e soluções. Por ser uma atividade ampla há diversas maneiras de prototipar a interface de um software. Em seguida será mostrado algumas formas de prototipação de interface.

4.4.1 Rabisco Frame

Segundo (SANCHES DA SILVA, 2010) rabisco frames ou prototipação em papel é criação de forma rudimentar da interface do software, que facilita o entendimento do cliente e a fácil mudança nas primeiras fases do desenvolvimento. Sua forma de aplicação é a utilização de papel, lápis, borrachas, fotocópias e recortes. Abaixo a demonstração da interface feita utilizando as técnicas de rabisco frames referente ao aplicativo descrito nesse artigo.

Foram utilizados papel e caneta como ferramentas para produção do modelo de interface representado na figura 6. De forma simples e com desenhos do autor desse artigo, é possível entender que a interface foi dividida em cinco telas principais, mas ao decorrer da codificação será visto a necessidade de criar outras telas para suprir os requisitos do aplicativo.

Figura 6 -Montagem da interface através de técnicas de prototipação em papel criado pelo autor deste trabalho.



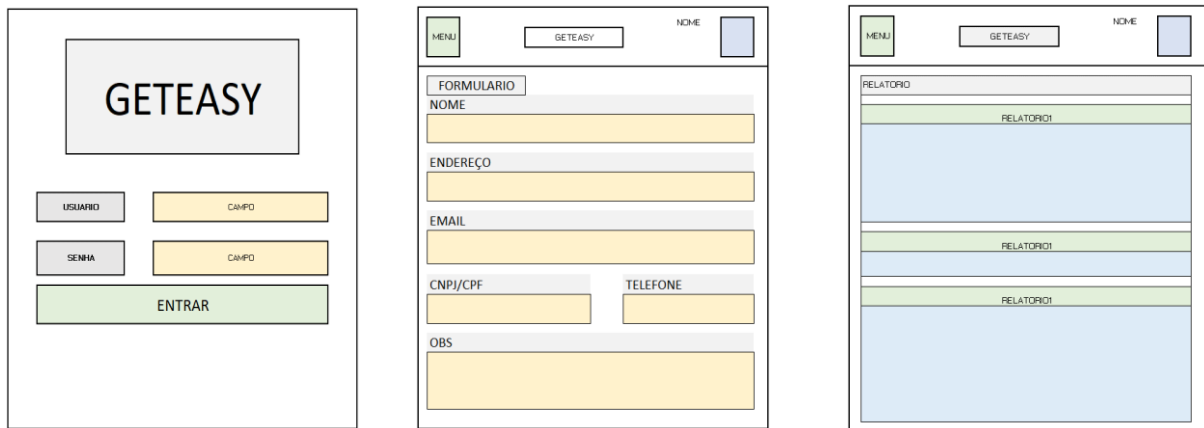
Fonte: O Autor.

4.4.2 Wireframes

Segundo (SANCHES DA SILVA, 2010) wireframes é feito de forma a documentar a estrutura e o conteúdo da interface, mostrando a relevância dos elementos do layout e as junções feitas entre eles. Durante o desenvolvimento é mostrado de forma estruturada os elementos como imagens, textos, caixas de seleção e animações, utilizando diferentes cores, formatos, fontes e estilos de letra para diferenciá-los, mantendo sempre o mesmo padrão para todos wireframes.

A Figura 7 apresenta a interface desenvolvida utilizando a técnicas de wireframes com ajuda da ferramenta Excel que possibilitou trabalhar com *grids*. Agora é possível identificar os locais da funcionalidade por meio das cores; a cor verde foi utilizada para apresentar botões e áreas clicáveis, a azul para demonstrar áreas com imagens ou gráficos, áreas amarelas identificamos campos dos formulários e a parte cinza são as representações dos textos e títulos. Nessa amostragem foi desenvolvido apenas as telas de *Login*, *Home* e cadastro simplificado.

Figura 7 - Interface através de técnicas de prototipação wireframes criado pelo autor deste trabalho



Fonte: O Autor.

5 Resultado

As Figuras 8, 9, 10 e 11 a seguir, mostram de forma estilizada a interface pensada para o aplicativo descrito neste artigo, de forma a apresentar com clareza a ideia para um futuro desenvolvimento. Para a criação dessas telas foi utilizada a ferramenta de edição Adobe Photoshop, com a paleta de cores do *Materialize*.

Observando a Figura 8, é possível visualizar com maior clareza o formato do aplicativo descrito neste artigo. As cores utilizadas foram observadas em desenvolvimentos de aplicativos similares e junto com o estudo da teoria das cores, segundo (PENDRASSOLLI; NERIS, 2014), foi possível chegar ao consenso que a utilização da cor vermelha nos botões de ação passa a sensação de confiança e intensidade da decisão e as cores acinzentadas são excelentes como base para destacar qualquer cor não neutra além de promover a sensação de profissionalismo e estabilidade.

Na primeira tela vemos o nome do aplicativo GetEASY o qual passa a ideia de ter uma fácil utilização. Abaixo do título temos dois campos, Usuário e Senha, e o botão de entrar para efetuar o login. Em seu rodapé é visível o botão de configuração que não teve sua interface desenvolvida pois há decisões a serem tomadas para os futuros desenvolvimentos.

Figura 8 – Interface da tela de Login

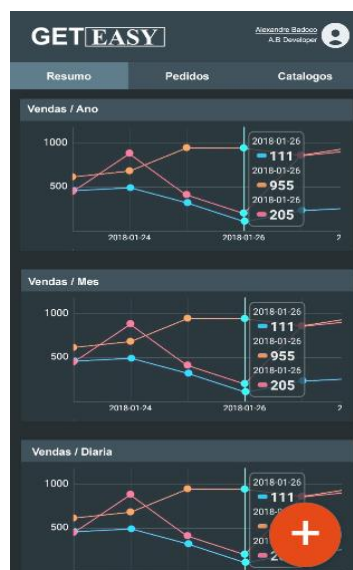


Fonte: O Autor.

Como citado anteriormente, há a opção de configurar o aplicativo por meio do botão de configuração, e nele pode ser acrescentado uma forma de adaptação para usuários com necessidades especiais, opções de alterar as cores para alguns tipos de daltonismo e a possibilidade de aumentar o tamanho da letra para facilitar a leitura.

Na Figura 9 podemos observar a barra de navegação com três itens principais sendo eles: resumo, pedidos e catálogo. O resumo mostra os gráficos de vendas por ano, por mês, as comparações de saída por produto ou por cliente

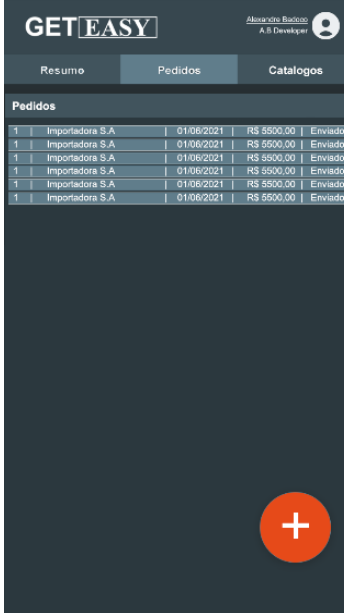
Figura 9 – Interface da tela de resumo



Fonte: O Autor.

É possível ver na figura 10 a apresentação da tela de pedidos exibindo os pedidos efetuados, cancelados ou pendentes por meio de uma lista com as informações básicas de cada pedido possibilitando a verificação detalhada com apenas um clique em cima do registro.

Figura 10 – Interface da tela de Pedidos



The screenshot shows the 'GET EASY' mobile application interface. At the top, there is a header with the logo 'GET EASY' and the user's name 'Alexandre Bastos' with a profile icon. Below the header, there are three tabs: 'Resumo', 'Pedidos' (which is selected), and 'Catalogos'. The main content area is titled 'Pedidos' and displays a list of six orders. Each order is represented by a row with the following columns: a small icon, the company name 'Importadora S.A.', the date '01/06/2021', the amount 'R\$ 5500,00', and the status 'Enviado'. At the bottom right of the screen, there is a large red circular button with a white plus sign, likely used to add new orders.

Ícone	Importadora S.A.	01/06/2021	R\$ 5500,00	Enviado
Ícone	Importadora S.A.	01/06/2021	R\$ 5500,00	Enviado
Ícone	Importadora S.A.	01/06/2021	R\$ 5500,00	Enviado
Ícone	Importadora S.A.	01/06/2021	R\$ 5500,00	Enviado
Ícone	Importadora S.A.	01/06/2021	R\$ 5500,00	Enviado
Ícone	Importadora S.A.	01/06/2021	R\$ 5500,00	Enviado

Fonte: O Autor.

A figura 11 apresenta a interface de uma tela de formulário onde todas as telas serão feitas neste padrão de desenvolvimento. Campos que podem utilizar muitos dados grandes são deixados sozinhos em uma única linha e os que utilizam poucos dados pequenos podem dividir a tela em até três colunas de forma que não aconteça confusão ao ser selecionados os campos do formulário.

Figura 11 – Interface de representação de um formulário



The image shows a mobile application interface for 'GET EASY'. At the top, there is a navigation bar with the app name 'GET EASY' on the left and a user profile icon on the right labeled 'Alexandre Bastos A.B. Developer'. Below the navigation bar are three tabs: 'Resumo', 'Pedidos', and 'Catalogos'. The main content area is titled 'Cadastro Cliente' and contains a registration form with the following fields: 'Nome:*', 'Razão Social:', 'CPF/CNPJ:*', 'Telefone:*', 'Rua:', 'Numero:', 'Bairro:', 'Cidade:', 'UF:', 'CEP:', and 'Pais:'. Each field is represented by a white input box. At the bottom of the form is a large orange button labeled 'ENTRAR'.

Fonte: O Autor.

Como citado anteriormente, por meio do uso do Adobe Photoshop e da paleta de cores do Materialize para a criação das telas, e com o auxílio das Figuras apresentadas anteriormente, é possível idealizar como ficaria algumas partes da interface do aplicativo.

6. Acessibilidade ao Daltonismo

Conhecido como daltonismo, a doença discromatopsia pode ser genética e hereditária. Infelizmente não há um grande foco em pesquisas e dificilmente o diagnóstico ocorre precocemente. Esta doença se dá quando há anomalia no gene responsável em traduzir estímulos elétricos produzidos pelas fotopsinas que são as proteínas fotorreceptoras encontradas nos cones da retina que são a base da visão a cores (MORIJO, MARCELINO, MANSANO. 2017).

Anomalia nas proteínas dos cones podem criar diferentes tipos de daltonismo, sendo alguns como a Discromatopsia congênita, o Tricomatismo anômalo e Tritanomalia. A anomalia impede que certos cones da visão não consigam converter os sinais elétricos, assim pessoas com problemas no cone M (verde) passam a apresentar picos de onda bem próximas ao cone do tipo L (Vermelho), assim dificulta a visão das cores verde e vermelho.

Há pesquisas para desenvolvimento de tecnologias que podem resolver a distorção das cores, mas nada conclusivo segundo (LEE, SANTOS 2008), elas trazem a análise de uma ferramenta que simula uma leve correção usando filtros em imagens. Utilizando certos esquemas de cores é possível minimizar o desconforto do usuário daltônico ao utilizar o aplicativo.

7. Considerações Finais

Como descrito no início deste artigo, o aplicativo, após a finalização do seu desenvolvimento, poderá suprir as necessidades de empresas que possuem representantes comerciais e a necessidade de automatizar o processo de vendas. Por meio de uma observação realizada pelo autor dentro de indústria e lojas, foi identificado essa carência nesse gerenciamento. O aplicativo tem a intenção de promover um aumento nas representações de pequenos comerciantes assim gerando um melhor rendimento econômico.

Por se tratar de uma prototipação não foi mostrado todas as telas e suas funções, foram mostradas apenas as ideias principais e suas diferenciações. Futuramente poderemos ver este aplicativo funcionando nos smartphones com suas funções e interfaces que não foram descritas neste artigo.

Após a conclusão do desenvolvimento deste aplicativo a distribuição do aplicativo pode ser feita através das plataformas *PlayStore* e *AppleStore* que possibilita a fácil instalação e atualização. Essas plataformas através do Google ADS permite a divulgação de forma simplificada e abrangente.

Referências

BOOCH, Grady; **RUMBAUGH**, James; **JACOBSON**, Ivar. “UML – Guia do Usuário”, Rio de Janeiro, Campus, 12ª Edição, 2005.

DANIELSSON, William. React Native application development, 2016. 10 13-28 Disponível em: <https://www.diva-portal.org/smash/get/diva2:998793/FULLTEXT02>. Acesso em: 03 de Abril 2021.

EXPO. Feature Preview 2021. Disponível em: <https://expo.io/eas>. Acesso em: 04 Abril de 2021.

FERREIRA, Maria. Estudo da Prototipação na Engenharia de Requisitos para Desenvolvimento de Softwares Interativos em ciclo de vida Acelerado, 2003 Disponível em: http://cassiopea.ipt.br/teses/2003_EC_Maria_Rosilene_Ferreira_Lopez.pdf. Acesso em: 24 de Setembro de 2021

GOOGLE. The Developer's Guide. APIs do Google, 2021. Disponível em: <https://developers.google.com/android>. Acesso em: 23 de Março de 2021.

HANSSON, Nicolas; **VIDHALL**, Tomas. Effects on performance and usability for cross-platform application development using React Native, 2016. Institutionen for datavetenskap: Department of Computer and Information Science. Disponível em: <https://www.diva-portal.org/smash/get/diva2:946127/FULLTEXT01.pdf>. Acesso em: 04 de Abril de 2021.

LECHETA, Ricardo R. Google Android: Aprenda a criar aplicações para dispositivos moveis com Android SDK, São Paulo: Novatec, 2010. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=NrVUAwAAQBAJ&oi=fnd&pg=PA21&dq=android+jdk&ots=QcMEGaNj0p&sig=cFXAdA6GH1NFwXck5AO3cZHXwsk#v=onepage&q&f=false>. Acesso em: 25 de Março de 2021.

LEE, Jinmi. **SANTOS**, Wellington Pinheiro. Uma ferramenta adaptativa para facilitar a visualização de imagens para pessoas portadoras de daltonismo. 2008. Disponível em: https://www.researchgate.net/profile/Wellington-Dos-Santos/publication/323166359_Uma_ferramenta_adaptativa_para_facilitar_a_visualizacao_de_imagens_para_pessoas_portadoras_de_daltonismo/links/5a83bb51a6fdc6f3eb2a678/Uma-ferramenta-adaptativa-para-facilitar-a-visualizacao-de-imagens-para-pessoas-portadoras-de-daltonismo.pdf. Acesso em 22 de Outubro de 2021.

MARTINS, Celly de Siqueira; **ANTONIO**, André Lara Temple de; **OLIVEIRA**, Carlos Antonio de. Os desafios para a mobilização de aplicações baseadas em plataforma Web. X Encontro Anual de Computação – EnAComp, 2013. Disponível em:

<https://www.enacomp.com.br/2013/anais/pdf/39.pdf>. Acesso em: 23 de Março de 2021.

MATOS, Laura Rocha de; at. all. Vulnerabilidades do IOS, 2020. Porto Alegre: Seminário de Tecnologia Gestão e Educação. Disponível em: <http://raam.alcidesmaya.edu.br/index.php/SGTE/article/view/281/269>. Acesso em: 25 de Março de 2021

MORIJO, Daniel Kleber Santos. **MARCELINO**, Vitória de Oliveira. **MANSANO**, Naira da Silva. Daltonismo e as diferentes percepções de cores. REGRAD, UNIVEM/Marília-SP, v. 10, n.1, p. 433 - 439 outubro de 2017. Acesso em 22 de Outubro de 2021.

OSTERWALDER, Alexander; **PIGNEUR**, Yves. BUSINESS MODEL CANVAS. 2011. Disponível em: <http://www.cenda.cl/images/descargas/sanvicente.pdf>. Acesso em: 13 de Abril de 2021.

PENDRASSOLLI, Leandro Carlos; **NERIS**, Paula de Almeida. O Uso de Cores em Aplicações Web: um estudo dos projetos desenvolvidos no curso lato sensu de desenvolvimento de software para a Web. 2014. Disponível em: <https://silo.tips/download/o-uso-de-cores-em-aplicacoes-web-um-estudo-dos-projetos-desenvolvidos-no-curso-la> . Acesso em 22 de Outubro de 2021.

PENDER, Tom. "UML, A Bíblia", Rio de Janeiro: Elsevier, 2004.

PEREIRA, Lúcio Camilo Olivia. Silva, Michel Lourenço da. Android: Para desenvolvedores São Paulo: Brasport, 2009. Disponível em: <https://books.google.com.br/books?hl=ptBR&lr=&id=8u9wJowXfdUC&oi=fnd&pg=PA1&dq=android&ots=LVdm2b2kp&sig=dj78IEGJsYNAirjfTlw73vTIq9Q#v=onepage&q&f=false>. Acesso em: 25 de Março de 2021.

POLANCO, Kristel Malave; **TAIBO**, José Luis Beauperthuy. "ANDROID" GOOGLE'S OPERATING SYSTEM FOR MOBILE DEVICES. 2011. Disponível em: <https://dialnet.unirioja.es/descarga/articulo/7165367.pdf>. Acesso em: 23 Março de 2021.

SANCHES, Silvio da Silva. Análise comparativa de ferramentas computacionais para prototipação de interfaces. REVISTA CIENTÍFICA DO ITPAC Vol 3 pag. 3 Julho 2010. Acesso em: 09 de Outubro de 2021.

SARTORELI, Carlos Eduardo; **KUCHAUSKI**, Alves Uno. Comparativo entre IOS, android e Windows Phone, 2013. ETIC – Encontro de iniciação científica. Disponível em: <http://intertemas.toledoprudente.edu.br/index.php/ETIC/article/viewFile/3169/2919>. Acesso em: 25 de Março 2021.

SOMMERVILLE, Ian. Engenharia de Software. Addison Wesley, 2003. Disponível em: <http://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>. Acesso em: 09 de Outubro de 2021.

WHITE, Stephen A. Introduction to BPMN, 2004. Disponível em: http://yoann.nogues.free.fr/IMG/pdf/07-04_WP_Intro_to_BPMN_-_White-2.pdf. Acesso em: 31 de Agosto de 2021.