**DESENVOLVIMENTO DE UMA APLICAÇÃO**

**MOBILE PARA TROCA DE USADO**

Rafael Lima dos REIS[[1]](#footnote-1)

Centro Universitário Municipal de Franca – rafaellimareis01@gmail.com

Vinicius Marques PERES[[2]](#footnote-2)

Centro Universitário Municipal de Franca – viniciusmarquesperes@gmail.com

**Resumo**: texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto

**Palavras-chave**: texto; texto; texto

**Abstract**: *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*; *text*

**Keywords**: *text*; *text*; *text*;

**1 Introdução**

Uma das bases do capitalismo é o acúmulo interminável de bens onde a publicidade é usada como ferramenta para criar desejos nas pessoas de sempre adquirir coisas novas, gerando assim um acúmulo de coisas desnecessárias para a pessoa ou para qualquer integrante de sua família. Com isso, há um grande desperdício de espaço ocupado por esses itens. Sem saber o que fazer com estes, as pessoas tendem a deixá-los guardados até que estraguem ou se deteriorem, onde consequentemente são jogados no lixo.

Outro fator que alavanca esse acúmulo de bens materiais é a crescente onda de novas tecnologias, com constantes mudanças no mercado, tornando assim os produtos obsoletos com uma grande velocidade. Neste ponto entra a economia compartilhada, que ganhou força nos tempos de crise. Essa prática permitiu aos consumidores terem acesso a produtos de qualidade com um valor bem menor, ou em troca de outras coisas

A economia compartilhada foi Iniciada em meados de 90 nos EUA com o incentivo dos avanços tecnológicos da época, gerando assim uma redução dos custos das transações de “pessoa para pessoa” (SHIRKY, 2008). Silveira (2016, p.3) define a Economia compartilhada como um sistema que possui sua base formada em cima do compartilhamento de recursos humanos, comércio e consumo compartilhado de bens e serviços. Essa ideia sugere negociações de compartilhamento, empréstimo, aluguel, doação e troca de itens.

Mesmo com a existência da economia compartilhada hoje em dia a poucos métodos para realizar essas trocas. Existem algumas plataformas web destinadas a esse propósito, como "TomaLaDaCa", "Troca Fácil", mas que não chegaram a agradar os usuários, porque não apresentam um layout atrativo nem um ambiente organizado, onde, os usuários acabavam se perdendo em busca de produtos, ou que dificultaram alguns processos simples como cadastro de usuário ou produto. O mais utilizado atualmente para tentar alavancar a economia compartilhada são as trocas realizadas através de grupos em redes sociais.

Uma possibilidade ao invés de comprar é fazer uma breve pesquisa e trocar pela coisa que desejava ou por outra que lhe interessa, gerando assim uma economia em seu orçamento.

É quase impossível negar que passamos por uma evolução nos últimos anos para a “era digital” onde todas as mudanças na sociedade e nas organizações são transmitidas para todos em pouco tempo. Com essa evolução a internet vem tomando a maior parte do tempo das pessoas, junto com o celular se torna uma poderosa ferramenta para facilitar o dia-a-dia de todos, nas tarefas de compras, pagamentos e troca de itens. Por isso a maior aposta do mercado são os aplicativos mobile.

Com a economia compartilhada há algumas vantagens em aderir a ideia como: eliminar coisas desnecessárias, espaços extras na casa, novos itens sem pagar nada. Quem já aplica esse sistema utiliza normalmente grupos em suas redes sociais para fazer a troca de seus bens pessoais. Mas quem pratica esse tipo de atividade nem sempre sabe os riscos que corre, alguns desses riscos são a possibilidade de serem enganados em relação a qualidade do produto, sofrerem “trote” dos locais de troca e até receberem produtos estragados.

Foi analisado alguns sistemas encontrados hoje no mercado e em base dessa análise foi levantado alguns itens que poderiam ter sido abordados pelas aplicações. Algumas questões são: localização onde poderia ser apresentado aos usuários os itens mais próximos dele para evitar a necessidade de se utilizar o correio, outro ponto levantado é que quase nenhuma das aplicações possuem aplicativos, segundo pesquisa realizada pelo IBGE em 2015 (IBGE, 43) o acesso à internet era feito em 92% dos casos através do celular, ou seja, é bem mais fácil o usuário realizar suas trocas através de um aplicativo do que em seu computador.

Nesse contexto, o problema de pesquisa deste trabalho é responder se as tecnologias de desenvolvimento de aplicações para dispositivos móveis, bem como as metodologias e técnicas da engenharia de software disponíveis atualmente permitem o desenvolvimento de uma solução móvel para troca de usados.

O objetivo deste trabalho é o desenvolvimento de um software para dispositivo móvel independente de plataforma para apoiar a troca de usados utilizando tecnologias e metodologias da engenharia de software.

Os objetivos específicos são apresentados abaixo:

1. Criar documentação necessária do software

2. Produzir código-fonte de acordo com padrões de projeto de software

3. Testar a solução com potenciais usuários.

Os procedimentos metodológicos estão assim organizados:

1. Revisão bibliográfica dos temas aderentes ao tema do trabalho

2. Levantamento de requisitos e funcionalidades com potenciais usuários

3. Construção do diagrama BPMN

4. Criação de diagramas da UML

5. Projeto e desenvolvimento do software utilizando Angular 4 e Ionic 3 e o paradigmas baseado em componentes

6. Teste e avaliação com potenciais usuários

**2 Referencial Teórico**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

* 1. **E-commerce**

O e-commerce, que em português significa comércio eletrônico, é uma categoria de comercio que realiza transações por meio de dispositivos e plataformas eletrônicas, como computadores e smartphones. Um exemplo comum desta categoria é comprar ou vender produtos em lojas virtuais.

No início, o principal foco do e-commerce era para vender bens com valores modestos, como: livros e CDs. Atualmente, ele é utilizado para fins diversos, desde produtos que custam milhões até roupas e alimentos.

FELIPINI, DAILTON aponta a mudança na forma como as pessoas realizam compras e realizam transações. O autor aponta que essa mudança é uma excelente oportunidade para quem deseja ingressar no mundo do comércio eletrônico, pois é bem mais fácil se estabelecer em um setor que está em crescimento do que outro que está estabilizado. Veja a seguir algumas razões para criar um empreendimento na Internet.

* + 1. **A força da novidade iguala as oportunidades**

O conhecimento relativo a esse ambiente de negócios, o chamado know-how, ainda não está plenamente definido. Na verdade, empresários e estudiosos do e-commerce ainda estão buscando conhecimento e obtendo aprendizado com erros e acertos durante seus caminhos no e-commerce. Este fato tem como função de nivelador, onde a distância de quem já está neste ambiente e quem está iniciando é muito curta, sendo assim pequenos segmentos que, por ventura, possa a vir não interessar às grandes empresas, ou não estejam suficientes maduros, podem representar uma excelente oportunidade de negócios para empreendedores com uma visão mais ampla.

* + 1. **O empreendimento pode ser implantado aos poucos e testado**

Ao contrário de uma empresa tradicional, em que o início das operações ocorre somente com o empreendimento totalmente estruturado, um negócio no comercio eletrônico pode ser implementado em etapas, onde o investimento é diluído caucionando a facilidade nas correções de erros.

* + 1. **O momento é agora**

Com a crescente de dispositivos tecnológicos e recursos fornecidos pelos mesmos, estima-se que as transações eletrônicas só tendem a crescer, ou seja, cada vez mais estará concentrado informações e oportunidades no mundo eletrônico.

* 1. **Aplicações Híbridas**

Aplicações hibridas são aplicações que não são desenvolvidas para um dispositivo específico. Estas aplicações possuem características e funcionalidades que dependem de um conjunto de softwares para que sua implementação ocorra, como a câmera ou agenda, por exemplo. Outra característica das aplicações hibridas está no processo de desenvolvimento do projeto, que utiliza poucas linguagens e frameworks, portanto seu desenvolvimento ocorre de forma ágil e consequentemente possui um custo baixo.

Resumidamente os aplicativos híbridos são aplicativos desenvolvidos de forma responsiva, ou seja, para funcionar para qualquer dispositivo.

* 1. **Angular**

Angular é um framework Java script baseado em TypeScript. Foi criado e é mantido pela Google, lançado em 2016, atualmente está na versão 4. Baseado em componentes que de acordo com a documentação do (Angular, 2018) um componente é uma diretiva com configurações mais simples, que por sua vez são marcadores do DOM (tendo atributo, nome ou CSS) que com o auxílio do compilador do Angular é inserido comportamentos específicos, ou até fazendo transformações no próprio DOM (Modelo de objeto de documento).

(Dan, 2017) aponta alguns benefícios de se utilizar Angular e TypeScript:

* + 1. **Consistência**

Um dos fatos mais importantes no angular é que a estrutura geral é baseada em componentes e serviços. Os componentes por sua vez começam da mesma maneira (com seus "imports", "decorator" e "class"), mesmo tendo coisas adicionais a estrutura geral sempre é a mesma.

Outra área de consistência são os serviços, mesmo existindo várias opções (factories, services e providers) com angular há a possibilidade de "injetar" em seu construtor sem haver diferença e dificuldade entre cada um.

* + 1. **Produtividade**

Com a consistência também temos a produtividade, com as classes dentro dos módulos do ES6 ele se organiza e transforma em um código auto responsável, onde, os dados de entrada passam somente pelas propriedades de entrada e as saídas pelas propriedades de saída

* + 1. **Modularidade**

No Angular existem os módulos que fornecem uma maneira de organizar todas as funcionalidades da aplicação, dividindo assim tudo em recursos que podem ser reutilizados. Há também outros benefícios, como por exemplo o "lazy loading", onde um ou mais recursos são carregados em segundo plano ou de acordo com a demanda da aplicação.

* + 1. **Captura de Erros Precoce**

Como o Angular utiliza o TypeScript ele acaba ganhando algumas vantagens, como possibilidade de conexão com arquivos .tst ou diretamente com .js, sem ter qualquer tipo de perda de desempenho. Outra vantagem do TS é o suporte dos principais recursos do ES2015, ES2016 e ES2017. Também há o suporte para tipos (primitivos, interfaces e outros tipos personalizados).

Além dessas vantagens do TypeScript, o Angular também foi desenvolvido com incentivos aos testes. Tornando assim todo o ambiente com suporte aos testes de unidade e ponta a ponta com Karma e Jasmine.

* 1. **Ionic**

De acordo com a Documentação do Ionic (2018) ele é um *Software Development Kits* (SKD) que permite desenvolver aplicativos para dispositivos móveis com um desempenho quase similar aos aplicativos "nativos" (Java para Android e Objective-c ou swift para iOS) usando apenas linguagens web (HTML, CSS3 e seus derivados e angular).

Em seu artigo Bruno (2017) aponta alguns pontos positivos do Ionic:

● Estrutura do projeto, como cada componente é separado por pastas deixando assim mais fácil de encontrar os itens desejados.

● Organização das dependências através do @ngModule, onde, ao invés de declarar em vários lugares da aplicação só é necessário declarar em um só lugar. Deixando assim o código muito mais fácil de dar manutenção.

● Ionic CLI, um gerador de componentes, com isso pode ser desenvolvido (pages, providers, tabs, pipes, components e diretivas). Além disso todos esses componentes serão organizados de acordo com a estrutura padrão dos projetos Ionic.

● O suporte ao ES6 é outra vantagem do ionic, mesmo que nem os browsers não suporta a sintaxe, no momento em que a aplicação é compilada também é gerada uma versão para o ES5.

* 1. **Cordova**

(Cordova, 2017) é uma estrutura de desenvolvimento móvel. Onde é permitido usar tecnologias da web (HTML5, CSS3 e Java Script para desenvolvimento, onde é utilizado as APIs de ligação compatíveis com cada dispositivo para acessar os recursos de hardware como sensor, GPS, agenda, entre outros.

Em seu artigo (Allan, 2017) com a ajuda do Cordova os desenvolvedores híbridos têm acesso a quase todos os recursos nativos dos aparelhos, como câmera, sensor, lista de contatos, dados dos celulares através das APIs que o Apache Cordova disponibiliza. Quando se utiliza alguma linguagem para aplicações híbridas, é fundamental a utilização de plugins para a comunicação com o hardware dos dispositivos, com isso o desenvolvedor pode instalar só o que é necessário para sua aplicação. Com isso pode-se considerar que o Ionic funciona em cima do Cordova, enquanto o Ionic fica responsável pela interface gráfica e regras de negócio da aplicação, o Cordova fica responsável pela comunicação com o hardware e empacotamento do sistema para vários dispositivos móveis.

1. **Empreendendo a Solução Proposta**

Este capítulo apresenta um breve conceito sobre empreendedorismo, startup e o modelo Canvas, seguindo padrões e uma estrutura, que são usados por empresas multinacionais.

* 1. **Empreendedorismo**

Empreendedorismo é a forma de se fazer algo novo ou diferente de todos os que existem no mercado, onde é o termo referente a busca de outras oportunidades através da criatividade e inovação. Uma das formas mais usadas atualmente relacionado com um empreendedor é o início de uma nova empresa.

A definição mais “formal” para empreendedorismo é onde alguém coloca em prática uma ideia nova, oferecendo produtos ou serviços novos ou melhorados para o mercado.

Em seu artigo Telma (2011 apud Hoskisson 2008) diz que a essência do empreendedorismo é saber encontrar e ter a coragem de explorar as oportunidades, principalmente as que ninguém vê ou às quais ninguém reconhece uma possível chance de se tornar um grande negócio. Podendo assim ligar o empreendedorismo com a satisfação das necessidade e força para enfrentar crises, explorando oportunidades com criatividade.

* 1. **Startup**

Uma startup é uma empresa em fase inicial, geralmente no início da implementação e organização dos processos internos. Pode até ser uma empresa que ainda não iniciaram a venda de seu produto ou serviço, mas que já esteja funcionando, ou pelo menos em fase de instalação (Hugo, 2016).

Polyana (2012 apud Hermanson 2011) apresenta um ponto de vista semelhante onde diz que startups são empresas pequenas, em fase inicial, com sua principal meta o estudo e desenvolvimento de ideias inovadoras, com um baixo custo de manutenção e uma grande possibilidade de gerar lucros rapidamente.

* 1. **Modelo Canvas**

O modelo Canvas (Business Model Generation) é uma metodologia desenvolvida por Alex Osterwalder, durante sua tese de doutorado. Dividido em 9 componentes essenciais para todas as organizações, em formato de um quadro, para permitir que o negócio seja analisado de forma completa, deixando bem mais fácil as possíveis alterações para diversas ideias ou situações (Gleison 2018).

Com essa metodologia qualquer empresa pode criar modelos de negócio de suas empresas ou até recriar de outras. Esses componentes do modelo Canvas cobre as 4 áreas de um negócio, que são: clientes, oferta, infraestrutura e viabilidade financeira. Criado em meados de 2000 o Canvas já possui várias empresas que utilizam essa metodologia (IBM, Ericsson, governo canadense).

Os 9 blocos do modelo Canvas são descritos a seguir:

Segmentação do Cliente: é nesse bloco onde é definido o segmento de mercado onde o projeto irá afetar. Deve ser definido os principais clientes para o projeto.

Oferta de Valor: é onde a empresa define quais serão os principais benefícios que o cliente terá utilizando seu produto. É neste momento em que a empresa pode refletir e descobrir seus diferenciais diante dos concorrentes.

Canais: É definido por onde a empresa vai trabalhar o marketing do produto diante de seus clientes.

Relacionamento: É onde a empresa descreve sua estratégia para evitar que seus clientes troquem seu produto pela concorrência.

Fontes de Renda: O bloco de receita é onde a empresa determina o meio de cobrança do produto. Alguns exemplos são: venda, assinatura, licença, entre outros.

Recursos Chave: É listado os recursos fundamentais para que a empresa tenha sucesso na produção do produto.

Atividade Chave: Junto com os recursos chave, as atividades são as ações que devem ser feitas de forma constante para que o negócio funcione de maneira adequada.

Parcerias Chave: São os itens/atividades terceirizadas, ou seja, qualquer tipo de atividade ou matéria-prima fornecida por outra empresa deve ser listado neste bloco.

Estrutura de Custo: É onde a empresa define todos os principais custos com maior peso financeiro derivado de operações do negócio.

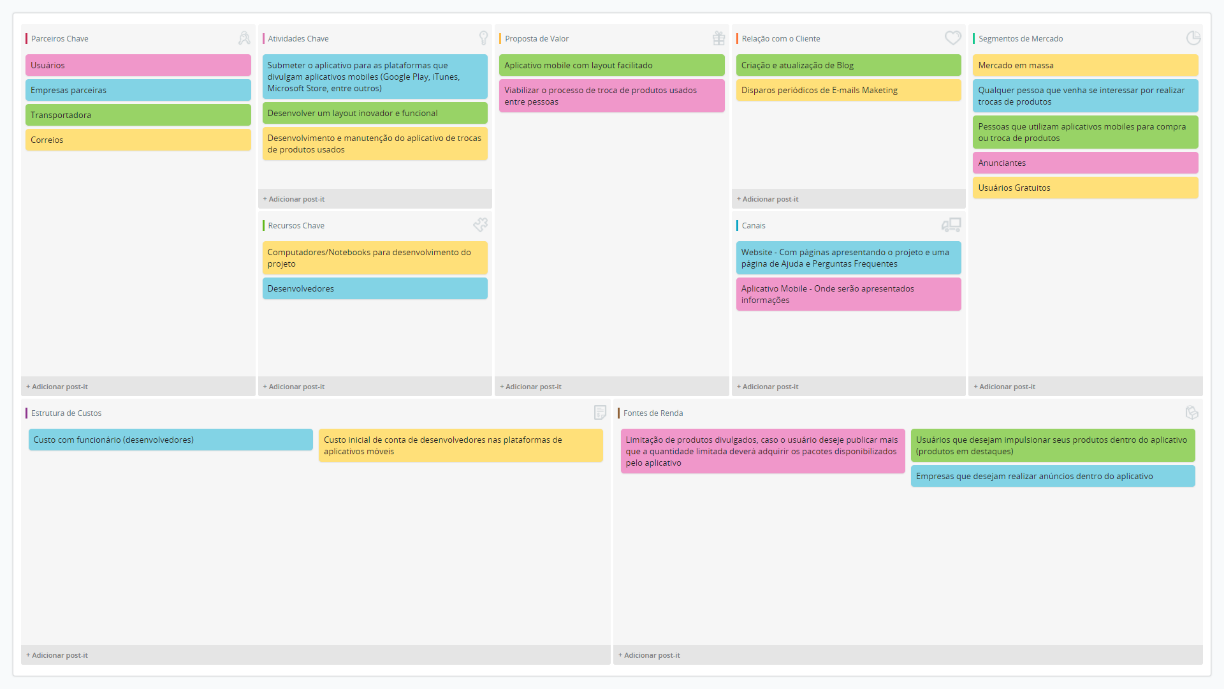


Figura 1 – Modelo Canvas

A Figura 1 do modelo Canvas desenvolvido para o projeto, contendo cada um dos nove componentes é discutido em documento disponível no repositório do projeto (GITHUB, 2018).

1. **Definição dos Processos da Aplicação Mobile**

Nos tópicos a seguir serão tratados as definições de todos os processos de documentação do projeto em desenvolvimento, que visa

O repositório das documentações deste projeto é encontrado no (GITHUB, 2018).

* 1. **BPMN**

O *Business Process Modeling Notation* (BPMN) é uma notação que representa por meio de diagramas os processos de negócio.

A modelagem é importante para esclarecer e tornar os processos visualmente, facilitando tanto em análise de melhorias quanto de automação dos processos (ARANTES, 2014).

O BPMN deste projeto é encontrado no (GITHUB, 2018).

* 1. **UML**

O *Unified Modeling Language* ou Linguagem de Modelagem Unificada (UML) é uma linguagem de modelagem que visa auxiliar os analistas de software a definir os requisitos, dinâmica dos processos, estrutura lógica, o comportamento do software. Essas características são determinadas antes mesmo do software entrar em desenvolvimento (FREITAS, 2008).

* 1. **Diagrama de Caso de Uso**

O diagrama de Caso de Uso é responsável por apresentar os processos do sistema do ponto de vista do usuário, descrevendo as principais funções do sistema (RIBEIRO, 2012).

Esse diagrama é composto por quatro partes, sendo elas:

* O cenário: responsável pelas sequências de eventos;
* Os atores: que representa o tipo de usuários;
* O use case: representando a funcionalidade ou tarefa;
* E a comunicação que liga um ator à um caso de uso.
  1. **Diagrama de atividade**

O diagrama de atividade tem como objetivo principal, utilizando o ponto de vista funcional do sistema, especificar o funcionamento do software. Tem semelhanças a um fluxograma, com isso fornece uma maneira mais lógica e menos abstrata do sistema em seu nível micro ou macro (VENTURA, 2016).

* 1. **Diagrama de estados ou Diagrama de máquina de estados**

O diagrama de estados, também conhecido como diagrama de máquina de estados, tem diversos tipos de usabilidade, sendo as principais ilustrar cenários de caso de uso em um contexto de negócios, descrever como os estados de um objeto e seu tempo de vida, apresentar o comportamento feral de uma máquina de estado, entre outros (Lucidchart, 2018).

* 1. **Diagrama de sequência**

Diagrama de sequência é responsável por descrever como, e em qual ordem, grupos de objetos trabalham, são conhecidos como diagramas de eventos. Esse diagrama ilustra os detalhes de um Caso de Uso, modelando a lógica de um processo, operação ou função (Lucidchart, 2018).

* 1. **UML**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

* 1. **UML**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

1. **Resultados**

Nesta seção iremos detalhar o desenvolvimento de toda a aplicação, desde o front-end que será o aplicativo mobile para os usuários, até a parte do back-end que será utilizado pela aplicação para se comunicar com o banco de dados.

* 1. **Back-end**

Para evitar ao máximo que a aplicação execute tarefas pesadas como consultas em um banco de dados e processamento de regras de negócio, decidimos desenvolver uma Api Rest (Representational State Transfer), onde o aplicativo terá apenas que passar alguns parâmetros e acessar as urls com os métodos de requisição HTTP (GET, UPDATE, DELETE, POST).

Inicialmente para a parte do backend foi usado a linguagem node, que é uma plataforma open source que roda códigos em javascript do lado do servidor e conta com vários pacotes auxiliares para se desenvolver uma API. Dois pacotes utilizados para desenvolver a estrutura inicial da aplicação foram:

Express: Utilizado para montar a base da api, como as rotas, middlewares de segurança e erros, e porta do servidor.

Consign: Foi utilizado para realizar todo o mapeamento dos arquivos da api, para que fosse criado uma ordem de carregamento, evitando assim qualquer erro de sincronização dos arquivos.

Na figura 1 pode ser observado como foi utilizado esses dois pacotes, no método “load” (linhas 14 a 19 e 33 a 35) onde carregamos todos os arquivos necessários em uma determinada ordem com o consign e em “app.listen” (linha 24 a 26) onde é definido a porta onde a api estará disponível com o express.

Figura 1 - Utilização de consign e express em node.js e estrutura do projeto

Outra funcionalidade do pacote express é a possibilidade de se criar os endpoints que a api necessita, dentro dessas rotas pode se obter parâmetros passados e direcioná los para outra parte do código, com isso temos uma aplicação mais organizada e de fácil manutenção.

Esse exemplo de roteamento pode ser observado na figura 2, onde pode ser definido a url “/products” (linha 9 a 17) e todos os métodos que essa url disponibiliza. Nessas rotas também pode ser definido funções auxiliares conhecidas como “middlewares” onde todas as rotas deverão passar, na imagem é utilizado a função de “auth.authenticate()” (linha 10) onde fica obrigatório que todos que acessarem essa url deverão passar o parâmetro “Authorization” que é uma chave única que cada usuário tem.

Figura 2 - Exemplo de rotas utilizando o express

A segunda parte do backend é sobre a comunicação da api com o banco de dados (PostgreSQL) escolhido pelo time de desenvolvimento. A biblioteca utilizada para fazer essa comunicação foi o sequelize um ORM (Object Relational Mapper) feito para backend desenvolvido em js.

Um ORM é uma técnica de mapeamento objeto relacional, onde é feita a relação dos objetos com os dados salvos em banco. Essa técnica vem se tornando bastante conhecida, pois, além de aumentar a produtividade ele deixa bem abstrato as consultas sql.

Figura 3 - Exemplo de Models para o sequelize

Na figura 3 é possível observar a definição da tabela de produtos, onde é definido o campo “id” como chave primária e auto incremento (linha 3 a 6), e os campos “nome” e “id\_user” como obrigatórios (linha 8 a 21), caso a biblioteca não receba um desses campos obrigatórios ele automaticamente retorna uma mensagem de aviso e não permite a inserção da tupla na tabela. Também é possível observar todos os relacionamentos com outras tabelas, definindo o tipo de relacionamento e suas respectivas chaves estrangeiras.

Por fim para facilitar o uso da api foi desenvolvido uma documentação de todos os endpoints disponíveis e com exemplos de como deve ser utilizado, essa documentação está disponível em um servidor, é disponibilizado menu com todas as requisições e exemplos de uso como na figura 4.

Figura 4 - Imagem da documentação da API

4.2 FrontEnd

A segunda parte do sistema é a aplicação mobile desenvolvida no framework ionic que utiliza a linguagem TypeScript para que seja desenvolvido o aplicativo em uma linguagem e compilado para 3 sistemas operacionais diferentes (Android, iOS e Windows Phone).

O framework vêm com várias funcionalidades para facilitar o desenvolvimento da aplicação e a divisão entre todas as tarefas, deixando assim um código muito mais organizado. Um desses exemplos é toda a comunicação que o aplicativo mobile faz com a API, na figura 5 pode ser observado da linha 25 a 27 onde é definido um método get para buscar informações dos produtos a serem trocados.

Figura 5 - Imagem de um provider da aplicação mobile

Como foi exibido na figura 5, há arquivos separados por funcionalidades. Onde os responsáveis por acessar essas providers que consultam a API, são as pages, onde cada página é dividida em 3 arquivos principais (.ts, .html, .module.ts, .css) onde o .html é responsável por exibir a página para o usuário, .css é onde fica todo o estilo da página e efeitos, e o .ts é onde a lógica fica, todo o acesso ao hardware do dispositivo e aos provider. Por ser construído em módulos o framework ionic necessita apenas da injeção das dependências que a página vai precisar em seu funcionamento, na figura 6 da linha 21 a 24 são definidas todos os componentes que a página vai usar, na linha 29 a 36 é definida a função que utiliza o provider para buscar as informações que serão exibidas para os usuários.

Figura 6 - Page com injeção de dependências e consulta aos providers

Outra parte que foi trabalhada no projeto foi a reutilização de código onde com a ajuda do ionic foi possível criar componentes que podem ser inseridos em várias páginas diferentes, evitando assim que o código seja duplicado em várias páginas. Na figura a seguir (figura 7) é definido um simples componente que têm como objetivo criar um slider de imagens (será mostrado mais a frente a tela em que esse slider é utilizado), com isso qualquer página que precisar de um slider, deve somente injetar a dependência e utilizar o componente, sem ter que reescrever o mesmo novamente.

Figura 7 - Definição de um componente

Após todas as fases de planejamento e desenvolvimento do projeto, terminamos a primeira versão da aplicação, a seguir será apresentado algumas telas desenvolvidas no decorrer do projeto. A tela inicial da aplicação no primeiro acesso é a tela de login que pode ser vista na figura 8, onde o usuário deve colocar seu “login” e senha que foi cadastrado, também é possível acessar o aplicativo com o facebook, ou até criar um novo cadastro.

COLOCAR FIGURA 8 DA TELA DE LOGIN

Após realizar o login o usuário deve acessar menu e selecionar uma das opções disponíveis, entre elas uma lista de produtos disponíveis para troca em nossa plataforma, outra opção é visualizar todas as ofertas que fez a outros usuários ou que recebeu, também é possível visualizar chats, um pequeno relatório sobre as trocas já realizadas na plataforma, por último é possível selecionar um submenu chamado “cadastros” onde pode visualizar seu perfil ou cadastrar um novo produto ou editar um produto já cadastrado para troca. Essas opções são apresentadas na figura 9.

COLOCAR FIGURA 9 DO MENU

Uma das opções dita acima é a área dos seus produtos, onde o usuário pode editar um produto que ele já havia criado ou cadastrar um novo em alguns passos, para criar um novo produto primeiro ele deve colocar algumas imagens do produto novo, e preencher algumas informações para o novo produto ser criado (figura 10), com isso já vai estar liberado para outras pessoas fazerem ofertas.

COLOCAR FIGURA 10 DA TELA DE CRIAR UM PRODUTO

A principal função da aplicação é as ofertas de troca que pode ser realizadas por qualquer usuário cadastrado, basta apenas escolher um produto que deseja e ir para a tela de finalizar oferta (imagem 11), onde ele vai escolher um ou mais de seus produtos para oferecer para o dono do produto que ele deseja.

COLOCAR FIGURA 11 DA TELA DE FINALIZAR OFERTA

* 1. Assunto 2

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

1. **Conclusão**

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

**Referências**

ARANTES, Rhaíssa Nogueira. Artigo Introdução ao Business Process Modeling Notation (BPMN). 2014. Disponível em: <https://www.devmedia.com.br/introducao-ao-business-process-modeling-notation-bpmn/29892>. Acesso em: 12 set. 2018.

FREITAS, Márcio Nogueira. Artigo UML 2008. Disponível em: <https://www.devmedia.com.br/uml/8579>. Acesso em: 12 set. 2018.

RIBEIRO, Leandro. Artigo 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: 12 set. 2018.

VENTURA, Plínio. Artigo Entendendo o Diagrama de Atividades da UML 2016. Disponível em: <https://www.ateomomento.com.br/uml-diagrama-de-atividades>. Acesso em: 12 set. 2018.

Lucidchart. Artigo Diagrama de máquina de estados 2018. Disponível em: < https://www.lucidchart.com/pages/pt/diagrama-de-máquina-de-estados>. Acesso em: 12 set. 2018.

Lucidchart. Artigo Diagrama de máquina de estados 2018. Disponível em: < https://www.lucidchart.com/pages/pt/diagrama-de-sequência>. Acesso em: 12 set. 2018.

1. Discente do curso de Bacharelado em Sistemas de Informação [↑](#footnote-ref-1)
2. Docente do curso de Bacharelado em Sistemas de Informação [↑](#footnote-ref-2)