



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

LUCAS FUNCHAL DA SILVA

**TRANSFORMAÇÃO DIGITAL EM MICROEMPRESAS DO RAMO
ALIMENTÍCIO POR MEIO DE APP PARA COMPRAS**

Assis/SP
2021

LUCAS FUNCHAL DA SILVA

**TRANSFORMAÇÃO DIGITAL EM MICROEMPRESAS DO RAMO
ALIMENTÍCIO POR MEIO DE APP PARA COMPRAS**

Projeto de trabalho de conclusão de curso de Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando: Lucas Funchal da Silva

Orientador (a): Me. Guilherme de Cleve Farto

Assis/SP
2021

FICHA CATALOGRÁFICA

S586t SILVA, Lucas Funchal da
Transformação digital em microempresas do ramo alimentício por meio de app para compras / Lucas Funchal da Silva.
– Assis, 2021.

49p.

Trabalho de conclusão do curso (Ciência da Computação). –
Fundação Educacional do Município de Assis-FEMA

Orientador: Me. Guilherme de Cleve Farto

1.Serviços-web 2.App-compras

CDD005.12

TRANSFORMAÇÃO DIGITAL EM MICROEMPRESAS DO RAMO ALIMENTÍCIO POR MEIO DE APP PARA COMPRAS

LUCAS FUNCHAL DA SILVA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____

Me. Guilherme de Cleva Farto

Examinador: _____

Dr. Almir Rogério Camolesi

Assis/SP
2021

RESUMO

Em virtude do vírus da COVID-19 que assola o mundo, faz-se necessário meios para que as pessoas possam consumir serviços do cotidiano com segurança, sem se expor ao risco de contaminação. O presente trabalho é uma ação que se objetiva em auxiliar as pessoas a realizarem suas compras e a vender seus produtos na modalidade online, evitando desta maneira aglomeração e diminuindo os riscos de contágio. A implementação consiste no desenvolvimento de dois aplicativos consumindo microsserviços para a realização de compras e vendas em supermercados, mercearias e feiras. O trabalho tem por base e busca alcançar os conceitos da transformação digital, tema este que se reflete nos dias de hoje com os diversos setores da sociedade, assim como a promoção da digitalização de serviços que antes não exploravam tão fortemente a adoção de softwares para serem gerenciados.

Palavras-chave: Transformação Digital, Microsserviços, Serviços Web.

ABSTRACT

Due to the COVID-19 virus that is plaguing the world, means are needed so that people can safely consume daily services, without exposing themselves to the risk of contamination. The present work is an action that aims to help people make their purchases and sell their products online, thus avoiding crowding and reducing the risk of contagion. The implementation consists of the development of two applications consuming microservices to carry out purchases and sales in supermarkets, grocery stores and fairs. The work is based on and seeks to achieve the concepts of digital transformation, a theme that is reflected today with the various sectors of society, as well as the promotion of digitization of services that previously did not exploit so strongly the adoption of software to be managed.

Keywords: Digital Transformation, Microservices, Web Services.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1: Medindo o Progresso da Mudança Digital..... | 10 |
| Figura 2: Lugares e seus níveis de risco da COVID-19..... | 13 |
| Figura 3: Visão geral do ciclo de vida DevOps..... | 19 |
| Figura 4: Gráfico do uso de internet no Brasil..... | 20 |
| Figura 5: Gráfico de dispositivo usados para acessar a internet no Brasil..... | 20 |
| Figura 6: Gráfico de conexão por classe social | 21 |
| Figura 7: Tecnologias para a transformação digital..... | 22 |
| Figura 8: Principais componentes em uma arquitetura orientada a serviços | 24 |
| Figura 9: Diferença entre a arquitetura monolítica e a arquitetura de micro serviços..... | 27 |
| Figura 10: Prototipação de tela para realizar compra antes do desenvolvimento..... | 30 |
| Figura 11: Prototipação de tela para finalizar compra antes do desenvolvimento..... | 30 |
| Figura 12: Application.yml do eureka..... | 32 |
| Figura 13: Application.yml exemplo dos serviços..... | 33 |
| Figura 14: Main exemplo dos serviços para habilitar o eureka..... | 33 |
| Figura 15: Interface exemplo de comunicação dos serviços..... | 34 |
| Figura 16: As portas dos microserviços..... | 34 |
| Figura 17: Diagrama para entendimento de funcionalidade entre produto e estoque | 35 |
| Figura 18: Diagrama para entendimento de funcionalidade entre compra e carrinho..... | 36 |
| Figura 19: Diagrama de entendimento para a funcionalidade da baixa de estoque | 36 |
| Figura 20: ConFigurações para subir imagens no docker..... | 37 |
| Figura 21: Dockerfile exemplo..... | 38 |
| Figura 22: Contêineres das imagens criadas no docker..... | 38 |
| Figura 23: Tela de login e cadastro de estabelecimento..... | 39 |
| Figura 24: Tela de cadastro do produto e seu gerenciamento..... | 40 |
| Figura 25: Tela de estoque..... | 40 |
| Figura 26: Tela inicial do aplicativo do usuário..... | 41 |
| Figura 27: Tela de compra e carrinho..... | 42 |
| Figura 28: Tela de cadastro de endereços..... | 42 |
| Figura 29: Tela compras no aplicativo do cliente..... | 43 |

SUMÁRIO

| | |
|---|-----------|
| 1. INTRODUÇÃO..... | 10 |
| 1.2 OBJETIVOS..... | 12 |
| 1.2 JUSTIFICATIVAS..... | 12 |
| 1.3 MOTIVAÇÃO..... | 13 |
| 1.4 PERSPECTIVAS DE CONTRIBUIÇÃO..... | 14 |
| 1.5 METODOLOGIA DE PESQUISA..... | 14 |
| 1.6 RECURSOS NECESSÁRIOS..... | 14 |
| 1.7 ESTRUTURA DO TRABALHO..... | 15 |
| 2. TRANSFORMAÇÃO DIGITAL..... | 16 |
| 2.1 TENDÊNCIA E VANTAGENS NA TRANSFORMAÇÃO DIGITAL..... | 16 |
| 2.2 IMPRESCÍNDIVEIS DA TRANSFORMAÇÃO DIGITAL..... | 17 |
| 2.2.1 AGILE..... | 17 |
| 2.2.2 DEVOPS..... | 18 |
| 2.3 O GRANDE VOLUME DE DADOS..... | 19 |
| 2.3.1 A ASCENSÃO DOS DISPOSITIVOS MOVEIS..... | 19 |
| 2.4 TECNOLOGIAS DA TRANSFORMAÇÃO DIGITAL..... | 21 |
| 3. SERVIÇOS WEB..... | 24 |
| 3.1 ARQUITETURA ORIENTADA A SERVIÇOS..... | 24 |
| 3.2 REST | 25 |
| 2.3 MICROSERVIÇOS..... | 26 |
| 4. PROPOSTA DO TRABALHO..... | 28 |
| 4.1 TECNOLOGIAS..... | 28 |
| 4.1.1 JAVA..... | 28 |
| 4.1.2 SPRING BOOT..... | 28 |
| 4.1.3 ANGULAR..... | 29 |
| 4.1.4 EUREKA..... | 29 |
| 4.2 OS APLICATIVOS..... | 29 |
| 5. DESENVOLVIMENTO..... | 32 |
| 5.1 MICROSERVIÇOS..... | 32 |

| | |
|----------------------------------|-----------|
| 5.2 CONTÊINERS DOS SERVIÇOS..... | 37 |
| 5.2 OS APLICATIVOS..... | 39 |
| 6. CONCLUSÃO..... | 44 |
| 6.1 TRABALHOS FUTUROS..... | 44 |
| REFÊRENCIAS..... | 45 |

1. INTRODUÇÃO

Diante da situação de pandemia causado pelo vírus da COVID-19, é essencial a colaboração das pessoas para que evitem a propagação do mesmo, comportamento que exige o cumprimento do isolamento social. Entretanto e positivamente, os serviços que dão condições a este cenário cresceram desde o início do surto. Na Figura 1 é apresentado um gráfico que mostra o crescimento do comércio eletrônico nos Estados Unidos. É notado um aumento de 12,3% a 41,9% do seu varejo total entre os meses de março a setembro do ano de 2020.

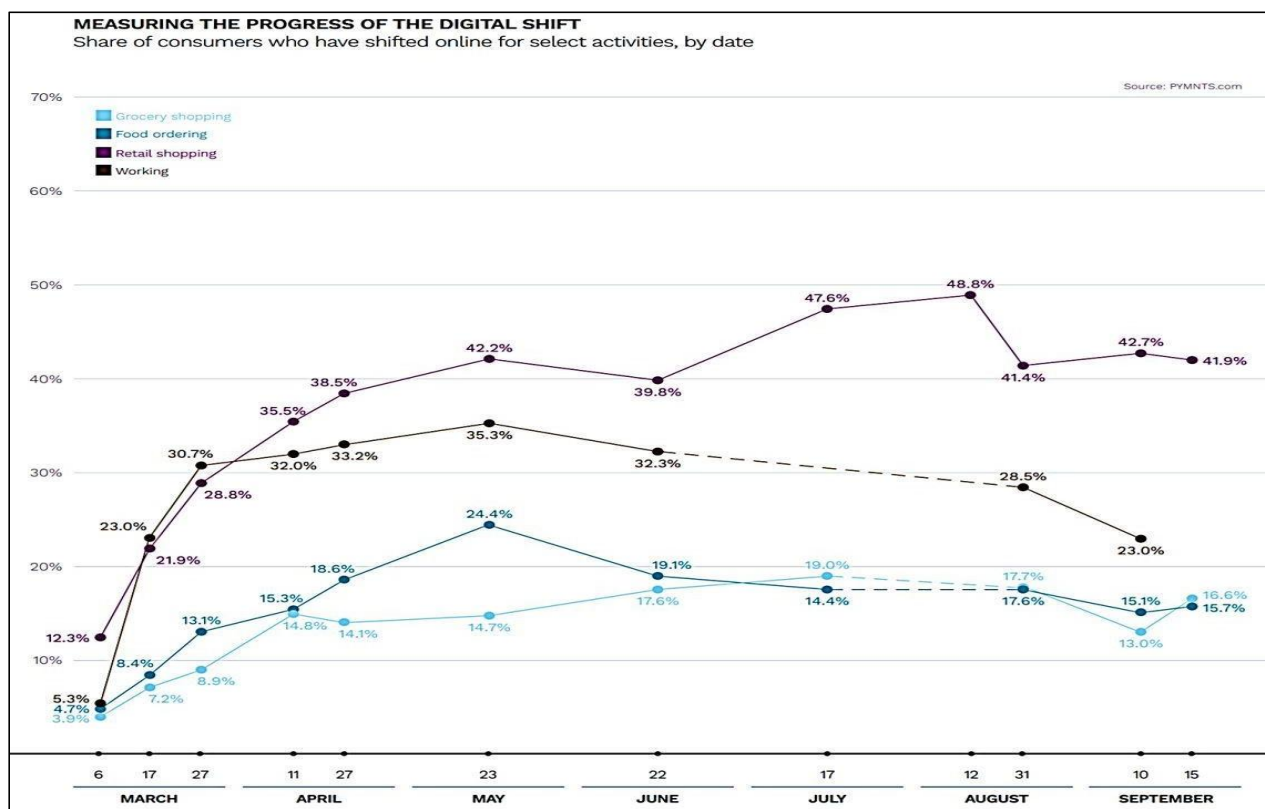


Figura 1: Medindo o Progresso da Mudança Digital

Fonte: PYMNTS, 2020

PYMNTS (2020), apresenta dados que fortalece a importância do desenvolvimento de aplicativos para o compartilhamento de serviços comuns do cotidiano, é destacado que de 100% dos usuários americanos que utilizam serviços online para compras em mercearias, 85,3% desejam manter o uso após o fim da pandemia. Já no serviço de pedido de comida prontas, 80,1%. Cria-se a oportunidade de investir e alimentar a cultura da transformação digital nas demais áreas do cotidiano, tendo como resultado o apoio na luta

contra o vírus.

Neste cenário, segundo IMD (2020), o Brasil avançou 6 posições no índice de competitividade digital, indo da 57ª posição para 51ª em 2020, e segundo FMI (2015) a previsão é de o país ocupar nos próximos anos a 8ª maior economia do mundo. Com base nas informações, temos uma demanda para o surgimento de empresas e start-ups com o objetivo de fomentar o desenvolvimento de soluções digitais que como consequência ajudaria no combate ao vírus da COVID-19.

Entretanto um ponto negativo é a causa do surgimento desse cenário apresentado, segundo VIALLI (2020) o Brasil ganhou 600 mil microempreendedores durante a pandemia, número ocasionado por demissões em massa durante a crise. Não obstante, microempreendedores sentem a falta de uma estrutura para compartilhar, divulgar e ter dados em forma de relatório dos seus produtos para realizar vendas com qualidade e segurança na modalidade online.

Há um grande movimento da transformação digital nas demais diversas áreas do nosso cotidiano, diariamente surge aplicativos que contribuem para o compartilhamento de serviços e gera inovação no mercado. Hoje uma grande referência é uma empresa brasileira, o *iFood*, que revolucionou o mercado no ramo de entrega de comida pronta e lidera no setor na América Latina, proporcionou lanchonetes e restaurantes a vender seus produtos. O Uber, uma empresa multinacional, revolucionou no setor do transporte, gerando emprego e ajudando na locomoção de pessoas com um preço mais acessível do que os táxis.

Motivado pela ajuda ao próximo, o presente trabalho tem por objetivo o desenvolvimento de dois aplicativos que consomem microsserviços, sendo um para o cliente cadastrar e vender seus produtos e outro para usuários comprarem os mesmos. Os aplicativos disponibilizam meios para que, durante a pandemia, empreendedores e usuários consigam realizar suas vendas e fazer suas compras com segurança respectivamente.

1.1 OBJETIVOS

O presente trabalho tem o objetivo de contribuir para a transformação digital dos serviços de mercados, feiras e mercearias, através do desenvolvimento de dois aplicativos webs que consomem microsserviços, sendo um que consiste no cadastro e gerenciamento para a venda de produtos e outro na compra dos mesmos. Outro objetivo é contribuir na luta contra o vírus da COVID-19 trazendo soluções para a diminuição de pessoas circulando em estabelecimentos alimentícios. De forma a tornar possível e combinar o desenvolvimento, se faz indispensável os seguintes itens:

- Pesquisar sobre transformação digital e arquitetura de serviços;
- Modelar os microsserviços;
- Desenvolver os microsserviços que respondam como APIs REST;
- Desenvolver os aplicativos para consumir os microsserviços;

1.2 JUSTIFICATIVA

LEAL (2020) aponta dificuldades do Brasil em lidar com dados referente a COVID-19, tal como número de pessoas contaminadas, mortes causadas pelo vírus e diagnósticos de testes.

Tendo em vista o grande e alarmante número de mortes causado pelo vírus da COVID-19, a negativa perspectiva de melhora a curto prazo, e dados imprecisos, é necessário incentivarmos e desenvolver meios digitais para combater o vírus.

Uma das práticas mais comuns e cultural é de ir ao supermercado ou à feira e fazer compras, mas adequado a pandemia, não há essa liberdade. Os aplicativos solucionam este problema, na tentativa de transformar digitalmente essa cultura para que pessoas consigam fazer suas compras e ajudar os microempreendedores a realizarem suas vendas sem perder a produtividade.

1.3 MOTIVAÇÃO

Na Figura 2 é apresentado os diversos locais e seus níveis de contaminação do coronavírus.



Figura 2: Lugares e seus níveis de risco da COVID-19

Fonte: UFMG, 2020

Com base na Figura 2, os ambientes de supermercados e feiras livres tem nível de contaminação médio, mas com o aplicativo podemos reduzir os riscos.

O desenvolvimento deste projeto se faz conhecido por suas bases, mas o tempo a qual vivemos se diferencia dos anteriores, não sabemos o que virá em seu decorrer, mas pensando na prevenção, a motivação é ajudar as pessoas se protegerem.

Outra motivação é num futuro não muito distante o aplicativo poder se tornar referência neste mercado, já que estamos passando pela transformação digital, práticas como o ato de fazer compras será realizado por meio de aplicativos.

1.4 PERSPECTIVA DE CONTRIBUIÇÃO

Ao final deste trabalho será disponibilizado os códigos em repositórios para que pessoas interessadas possam contribuir com melhorias, tanto em *code review* e ideias. Disponibilizar esse serviço a microempreendedores que não tem condições em ter uma solução para gerenciarem e venderem seus produtos.

Pretende-se publicar em forma de artigo para pessoas com o interesse nesta área e queiram dar continuidade em suas respectivas pesquisas.

1.5 METODOLOGIA DE PESQUISA

Os planos desse trabalho acadêmico foram alcançados por meio de pesquisas teóricas, para ter os fundamentos, foi necessária leitura de artigos científicos, livros, monografias, dissertações, teses, guias práticos e técnicos, livros e fontes digitais confiáveis.

Primeiramente, foram realizados estudos nas áreas de Transformação Digital e Serviços Web. Em seguida, foram realizados estudos de tecnologias na utilização do desenvolvimento, tal como *Spring* e *Eureka* para os microsserviços e *Angular* para o desenvolvimento dos aplicativos. Foram realizados também estudos de docker como infraestrutura.

1.7 RECURSOS NECESSÁRIOS

Para atender os objetivos propostos neste trabalho, os seguintes recursos de hardware e de software são importantes:

- **Hardware**
 - Notebook Dell

- Processador Intel Core i7 2.9 GHz.
- Disco SSD 214 GB.
- Memória RAM 16,0 GB.
- **Software**
 - **Eclipse ou Inteliij** – Ambiente de desenvolvimento para o *back-end* da plataforma;
 - **Visual Studio** – Ambiente de desenvolvimento para o *front-end* da plataforma;
 - **GIT** – Para versionamento de código;
 - **Angular CLI** – Para trabalhar com o *framework Angular*;
 - **Docker e Kubernetes** – Para conteirnização dos serviços

1.8 ESTRUTURA DO TRABALHO

Este trabalho será estruturado nas seguintes partes:

- **Capítulo 1 – Introdução:** Neste capítulo, será contextualizado a área de estudo e apresentará os objetivos, justificativa, motivação, metodologia de pesquisa, perspectiva de contribuição e recursos necessários para o desenvolvimento desta pesquisa.
- **Capítulo 2 – Transformação Digital:** Neste capítulo, foi abordado conceitos do tema referido e sua importância nos dias de hoje.
- **Capítulo 3 – Serviços Web:** Neste capítulo, foi abordado conceitos existentes sobre o tema, explanando o motivo e estudos da arquitetura a qual o presente trabalho se apoiará.
- **Capítulo 4 – Proposta de Trabalho:** Neste capítulo, foi apresentado a ideia dos dois aplicativos e suas tecnologias.
- **Capítulo 5 – Desenvolvimento do Trabalho:** Neste capítulo, foi realizado o desenvolvimento dos aplicativos e dos microsserviços.
- **Capítulo 6 – Conclusões:** Resultados e trabalhos futuros
- **Referências**

TRANSFORMAÇÃO DIGITAL

Este capítulo será apresentado o tema da transformação digital, identificando as tendências, vantagens, discorrer resumidamente dos imprescindíveis e crescentes temas que compõem o assunto, mencionar sobre o aumento do volume de dados e dos dispositivos móveis.

Segundo FERREIRA (2018) as formas de comprar e se relacionar mudaram ao longo do tempo, basta apenas um *smartphone* para ter tudo em mãos, o autor defende que a transformação digital mudou a maneira do ser humano viver e das empresas pensar sobre seus novos produtos. Um ponto importante deve ser ressaltado, KHAN (2016) discorre que a transformação digital não é somente a adoção de novas tecnologias, mas também de novas maneiras de tornar os negócios mais eficientes e competitivos.

1.1 TENDÊNCIAS E VANTAGENS NA TRANSFORMAÇÃO DIGITAL

Para SAUVAGE (2019) transformação digital é definida como o uso das tecnologias disponíveis para obter um aumento geral no padrão de vida da sociedade e gerar aumento no desempenho de negócios. Não se trata apenas de uma ação criativa, mas abrange a inovação, aperfeiçoar soluções existentes e de sucesso.

MOSER e JÚNIOR (2020) defende que para a inovação acontecer é importante ter um pensamento crítico sobre o problema, evitando desta maneira o conformismo, na intenção de sempre obter resultados expressivos com qualidade.

A grande tendência desta transformação é termos serviços comuns sendo incluídos em soluções tecnológicas. Ganha-se força para empresas e start-ups implementarem projetos para facilitar o cotidiano da sociedade, para tal afirmação é comprovada a seguir “Tem um aumento considerável no número de projetos inovadores para digitalização dos negócios com foco na melhoria do nível de satisfação dos clientes externos” (COMPUTERWORLD, 2017, p.4).

Outra grande tendência é citada por MOSER e JÚNIOR (2020), as empresas 4.0 tem como forte característica a troca da mão de obra humana por robôs e programas, e que

ao passo do tempo os funcionários devem se profissionalizar para manipular estas inovações. O autor pontua que estes profissionais devem ser capazes de operar sistemas reais, prever situações, lidar com grande e massivo volume de dados. O autor defende uma revolução cultural e educacional para que os futuros profissionais independentes da área de escolha estejam capacitados para executar tais tarefas, tendo em vista que todo o setor tem caminhado para a transformação.

A transformação digital tem o potencial de simplificar e integrar todas as operações para aumentar a eficiência dos negócios, ajuda as organizações a otimizar suas atividades para economizar tempo, esforços e recursos. A crise da COVID-19 tem sido um desafio, mas a transformação digital pode ajudar as empresas a suportarem a dificuldade financeira causada pelo vírus pois é exigido o distanciamento social e isolamento.

2.2 IMPRESCINDÍVEIS DA TRANSFORMAÇÃO DIGITAL

Agile e DevOps são imprescindíveis no desenvolvimento de software nos dias de hoje, assim como mobilidade e nuvem. (COMPUTERWORLD, 2017, p. 5). Os conceitos citados pela revista são de importância para empresas que querem migrar para soluções digitais, são termos que fazem parte de uma cultura ágil, relevante para a transformação digital acontecer.

2.2.1 AGILE

TIMOTEO e LIMA (2009) Define métodos ágeis como um conjunto de técnicas de desenvolvimento para engenharia de *software*. Esta abordagem tem seu espaço por dar valor a princípios como:

- Garantir satisfação do cliente com entregas rápidas e contínuas.
- Simplicidade.
- Cooperação constante entre clientes e equipe técnica.
- Colaboração com clientes mais do que contratos.

O autor frisa que cada funcionalidade desenvolvida utilizando abordagens *Agile* agrega no produto do cliente, trazendo qualidade e maior satisfação. Para afirmar que tais abordagens são de extrema importância, DINGSØYR et al (2012) diz que práticas ágeis de desenvolvimento de *software* têm como característica se adequar às constantes mudanças nos requisitos em qualquer estágio do processo de desenvolvimento. O processo é iterativo, levando em consideração o *feedback* dos clientes, para que suas necessidades sejam melhor atendidas.

Um dos métodos mais conhecidos no *Agile* é o *Scrum* e segundo SCHWABER (2004), a abordagem consiste em projetos interativos, organizado em *sprints* de poucos dias, tem como características times auto-organizados onde cada integrante assume atividades capaz de realizar dentro do período da *sprint*. O desenvolvimento deve ser capaz de se adaptar às mudanças através desses ciclos curtos. No final de cada *sprint*, um protótipo incremental é entregue e comparado com os requisitos. De acordo com *feedback* o produto pode ser ajustado para ficar de acordo com as necessidades do cliente.

2.2.2 DEVOPS

Segundo HUMBLE e FARLEY (2010), DevOps é um movimento, focado em contribuir para uma maior colaboração entre todos comprometidos na entrega de um sistema a fim de implementar aplicações relevantes de forma mais rápida e confiável.

Já para KIM et al (2016) o termo DevOps refere-se ao emergente movimento profissional que defende um envolvimento colaborativo entre os times de desenvolvimento e de operações, tendo como resultado um fluxo rápido, aumentando simultaneamente a confiabilidade, estabilidade e segurança do ambiente de produção.

Na Figura 3 é apresentado o ciclo de vida do DevOps, este contém todas as fases de um projeto.



Figura 3: Visão geral do ciclo de vida DevOps

Fonte: POLETTI, 2021

2.3 O GRANDE VOLUME DE DADOS

FERREIRA (2018) afirma que a era digital está presente para revolucionar e acelerar as mudanças na tecnologia e infere que as próximas duas décadas do século XXI terão mais transformações que nos últimos cem anos. A afirmação gera um desafio confirmado por ROGERS (2017) que diz que o maior desafio é converter uma grande massa quantidade de dados em informações valiosas.

Esse aumento das informações é correspondente a transações de dados *online* através da navegação na internet, interações com redes sociais e etc. O autor cita que neste cenário o *Big Data* é introduzido e quando aplicado seus conceitos já definidos é possível trabalhar com grande volume de dados. O uso dessa manipulação permite a quem utiliza informações de grande importância para os negócios. O próximo tópico guia um dos motivos para realização desse presente trabalho.

2.3.1 ASCENSÃO DOS DISPOSITIVOS MÓVEIS

O uso dos dispositivos móveis cresceu no Brasil nos últimos anos. Nas Figuras 4 e 5 comprova por meio de gráficos esse aumento.

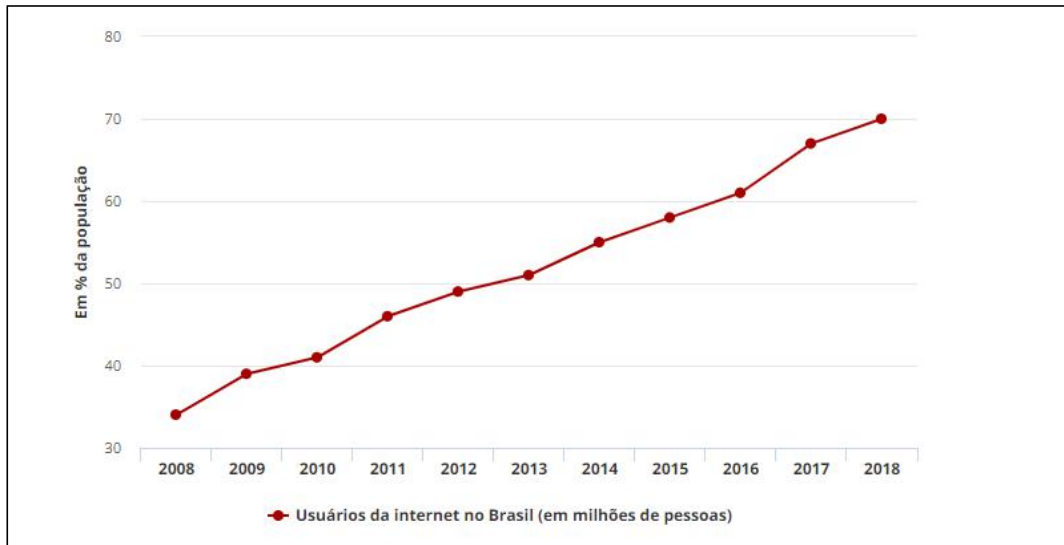


Figura 4: Gráfico do uso de internet no Brasil

Fonte: LAVADO, 2019

Na Figura 4 é notado que em um período de 10 anos o Brasil teve um crescimento de 36% de usuários na internet, correspondente a percentagem da população, é um grande aumento se considerado que outrora poucos tinham acesso a esse recurso e a tendência para os próximos anos é crescer conforme a próxima Figura nos apresenta.

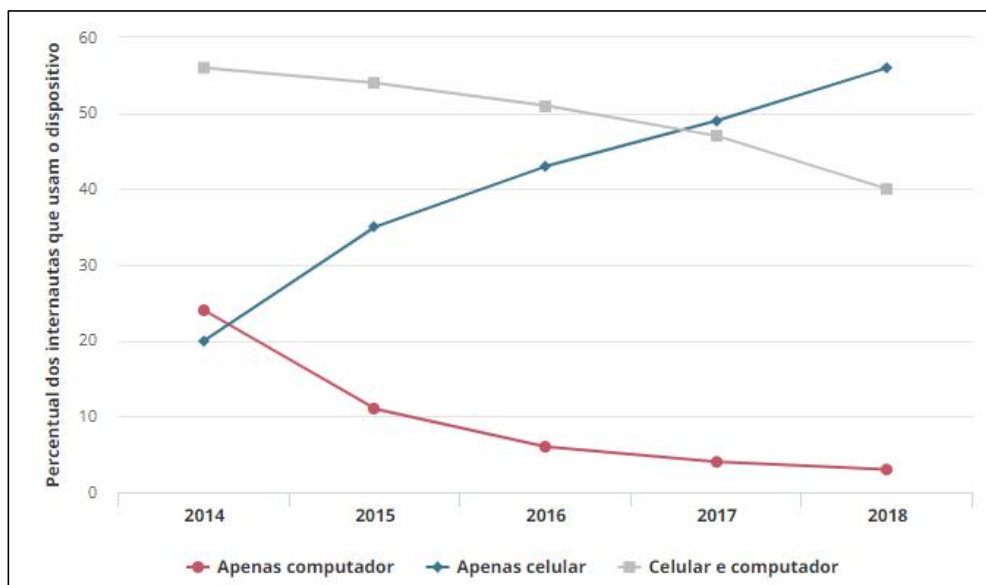


Figura 5: Gráfico de dispositivo usados para acessar a internet no Brasil

Fonte: LAVADO, 2019

Na Figura 5 é observado que no ano de 2014 o uso de celulares era menor que de computadores. No ano de 2015 o celular supera essa marca e em 2017 supera a

utilização dos dois juntos por pessoa. É comprovado que o celular hoje é capaz de realizar muitas atividades e atende nos serviços boa parte dos usuários sem a necessidade de ter um desktop.

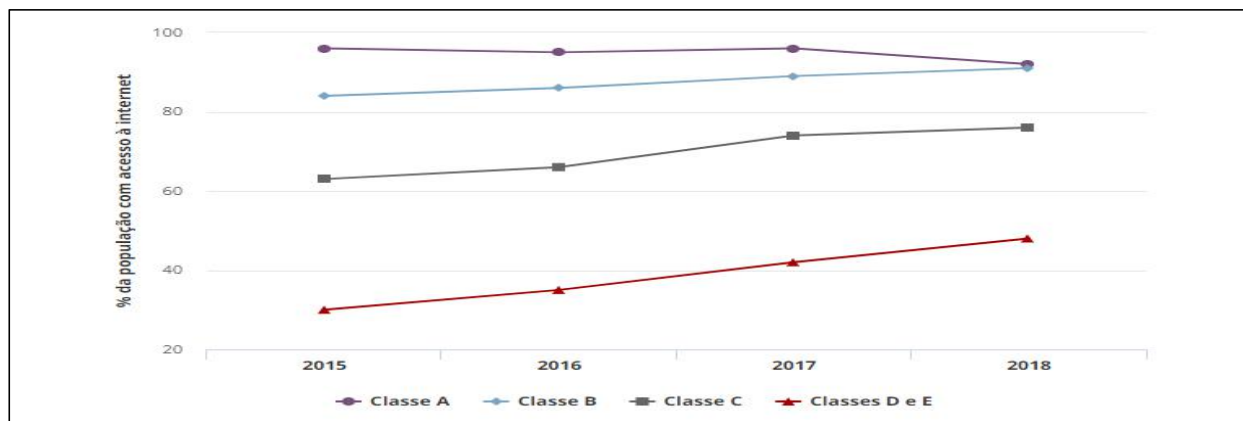


Figura 6: Gráfico de conexão por classe social

Fonte: LAVADO, 2019

Um dos motivos pelo aumento do acesso à internet no Brasil é mostrado na Figura 6 que aponta a inclusão e a acessibilidade desse recurso a classes mais pobres. Isso comprova que as empresas hoje devem pensar e investir em produtos que atendam a todas as classes.

Devido ao aumento de conexão, mais dados são gerados, valores apresentados são apenas do Brasil. Se faz de extrema importância a manipulação desses dados para obtenção de informações, e estes serem aplicados em soluções para que tenhamos uma transformação digital com o objetivo de gerar serviços de qualidade para a sociedade.

2.4 TECNOLOGIAS DA TRANSFORMAÇÃO DIGITAL

Segundo GURRIA (2017) as fases das tecnologias para que tenhamos um cenário da transformação digital são divididas em 3, nomeadas como: tecnologias habilitadoras, tecnologias integradoras e tecnologias de aplicação. A Figura 7 será empregada para organizar essas fases.

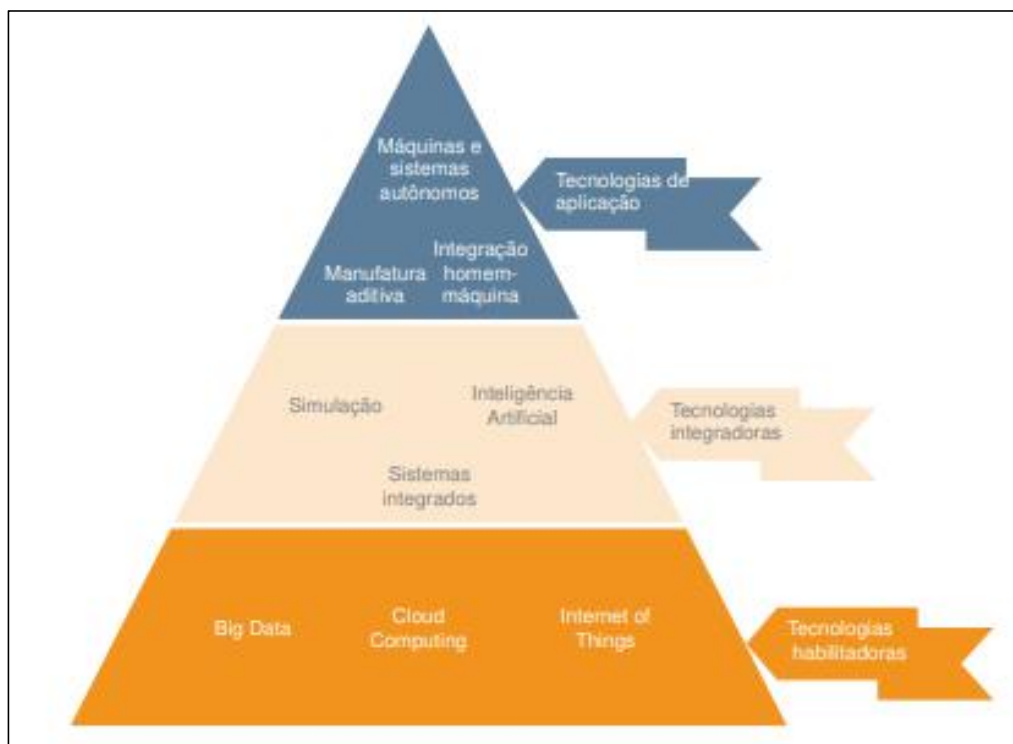


Figura 7: Tecnologias para a transformação digital

Fonte: GURRIA, 2017

Os sistemas *Internet of Things* são feitos por objetos inteligentes que processam informações e interage com o ambiente e com outros dispositivos conectados na internet. Os dados gerados pelo *Internet of Things* geram o *Big Data*, que impulsiona tecnologias como *Machine Learning* e Inteligência Artificial. O *Big Data* é responsável por coletar grande volume de dados e processá-los em alta velocidade e alta variedade. Uma das tecnologias habilitadoras, o *Cloud Computing* é definido como um “modelo para permitir acesso a rede universal e conveniente a um conjunto compartilhado de recursos de computação que podem ser provisionados e liberados com esforço mínimo de gerenciamento ou interação com o provedor de serviços” (MELL; GRANCE, 2011).

Segundo PIOVESAN (2020) as tecnologias integradoras são promovidas pelas tecnologias habilitadoras, é dado como exemplo: inteligência artificial. O autor define que a inteligência artificial é utilizada para automatizar processos, mas para se obter resultados concretos é necessário combiná-la com habilidades humanas para poder aproveitar o melhor que pode surgir desta interação.

Por fim a tecnologia de aplicação é citado por PIOVESAN (2020) como o resultado dos processos anteriores. Dentro dessa tecnologia o autor da como exemplo robôs avançados

colaborativos e veículos autoguiados. Chegamos à conclusão que o desenvolvimento de projetos que contenham a transformação digital é esperado que se utilize dessas tecnologias para geração de resultados e valor expressivos para a sociedade.

3. SERVIÇOS WEB

Um serviço web é uma parte da lógica de negócio e é acessado por meio de protocolos *HyperText Transfer Protocol* (HTTP). CHAPPELL e JEWELL (2002) esclarece que o uso de um serviço web tem níveis diferentes de complexidade, sendo útil para tarefas mais simples até processos difíceis e pesados.

3.1 ARQUITETURA ORIENTADA A SERVIÇOS

RICCI, DENTI e PIUNTI (2010) explica que serviços web na perspectiva da arquitetura orientada a serviços possuem como características principais o encapsulamento e ser fracamente acoplados. O autor ainda pontua que os serviços web é utilizado para a criação de sistemas distribuídos no qual necessita suportar interoperabilidade.

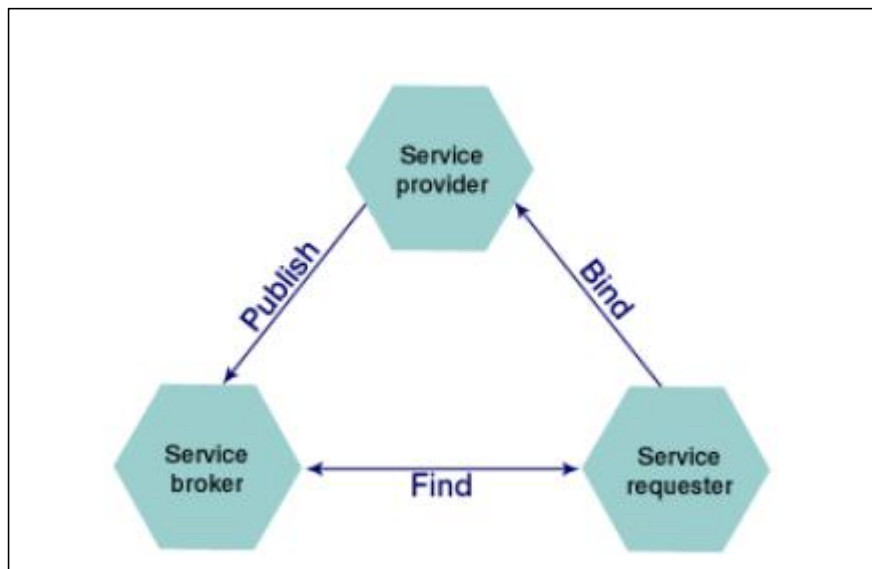


Figura 8: Principais componentes em uma arquitetura orientada a serviços
Fonte: IBM, 2000

Segundo GISOLFI (2001) o modelo da Figura 8 é composto por três tipos de componentes sendo: *Service Provider* o componente que contém a lógica da regra do serviço e disponibiliza os dados para que clientes consumam. O *Service Broker* atua como marca do serviço, um registro para que usuários consiga acessar. O *Service Requester* é aplicação que faz o uso dos serviços, obtém e envia dados.

3.2 REST

Segundo SAUDATE (2014) o estilo de arquitetura REST é movido pelas boas praticas de uso de HTTP, é baseado em recursos que são um conjunto de dados desse protocolo. Para a representação dos dados nessa arquitetura são usados *eXtensible Markup Language* (XML) e *JavaScript Object Notation* (JSON). Cada recurso é representado por *Uniform Resource Identifier* (URI) que corresponde a um endereço próprio. Para SAUDATE (2014) as regras a serem seguidas pelo conceito são:

- Semânticas de recursos: não existem limites quanto à utilizar os recursos, porém o ideal é que seja mantido um padrão em todas as URIs
- Interação por métodos: para interagir com as URIs devem ser utilizados métodos HTTP. Recomenda-se que os recursos sejam substantivos e os métodos sejam verbos. Isso significa que os métodos são responsáveis por provocar alterações nos recursos, que são identificados por URIs. Os métodos HTTP são:
 - GET: recupera dados da entidade, identificado pela URI;
 - POST: cria uma nova entidade;
 - PUT: atualiza uma entidade;
 - DELETE: exclui uma entidade.
- Representações distintas: uso de media types para alterar as representações de um mesmo conteúdo sob visão distintas por meio do cabeçalho Accept do HTTP. Por exemplo, o uso do XML ou JSON para dados, ficaria no cabeçalho como application/xml ou application/json, respectivamente, ou image/* para obter imagens
- Uso correto dos códigos de status: implica em conhecimento dos códigos de status do HTTP e correta aplicação dos mesmos para cada situação. Por exemplo, o código de quando ocorre um erro no servidor, o código quando um recurso é criado, o código de quando um recurso buscado não é encontrado, entre outros.

3.3 MICROSERVIÇOS

Segundo NEWMAN (2015) microserviços podem ser compreendidos como aplicações de pequeno porte, desenvolvidas como serviços e que operam em conjunto. Já para Lewis e Fowler (2014), uma arquitetura de microserviços é uma abordagem de desenvolvimento onde uma única aplicação é composta de um conjunto de pequenos serviços. Cada serviço possui seu próprio processo ou servidor dedicado, deve ser independente, escalável e projetado conforme as regras de negócios. Lewis e Fowler (2014) orientam os engenheiros de *software* e desenvolvedores a projetarem seus micro serviços em:

- Modelo em torno de conceitos de negócios: Os microserviços devem ser modelados em torno de contextos limitados, ou seja, espera-se que de a importância a regras de negócios.
- Ocultar detalhes de implementação internas: Para garantir o desacoplamento dos microserviços, espera-se ser ocultos detalhes de implementação e banco de dados. O diálogo entre os serviços deve ser feito através de APIs. Essa estratégia é um ponto importante para se obter facilidade de novos desenvolvimentos.
- Descentralização: Os microserviços devem ser capazes, coesos e com baixo acoplamento, de modo que alterações não afetem outros serviços.
- Implantação independente: Esse princípio acorda que qualquer microserviço deve ser lançado em produção sem afetar ou ter que implantar outros microserviços. Essa estratégia aumenta a velocidade de lançamento de novos recursos e dá uma maior autonomia para as equipes de desenvolvimento.
- Cultura de automação: Devido à dificuldade que os microserviços podem adicionar ao aumentar o número de serviços independentes do sistema, sugere-se que a empresa se apoie a cultura de automação, utilizando processos e ferramentas de entrega contínua.
- Isolar falhas: Os microserviços devem ser desenvolvidos para tolerar falhas do serviço. Caso ocorra alguma falha, espera-se que a aplicação possa responder da melhor maneira o usuário final.

- Altamente observável: Apesar dos benefícios em quebrar um sistema em serviços acaba gerando uma complexidade alta no monitoramento do comportamento de múltiplos serviços. Com isso, sugere-se utilizar técnicas como monitoramento semântico, que possibilita o monitoramento automático dos microsserviços.

Moreira e Beder (2016) cita as vantagens do uso de microsserviços e estes está relacionado a escalabilidade e facilidade de implementação. A escalabilidade é importante por implementar serviços específicos conforme a demanda do cliente. O autor também cita as desvantagens que são a complexidade do desenvolvimento, chamadas remotas, gerenciamento de bancos e transações que pode vir a ser muito custoso para equipes de desenvolvimento em caso de falhas.

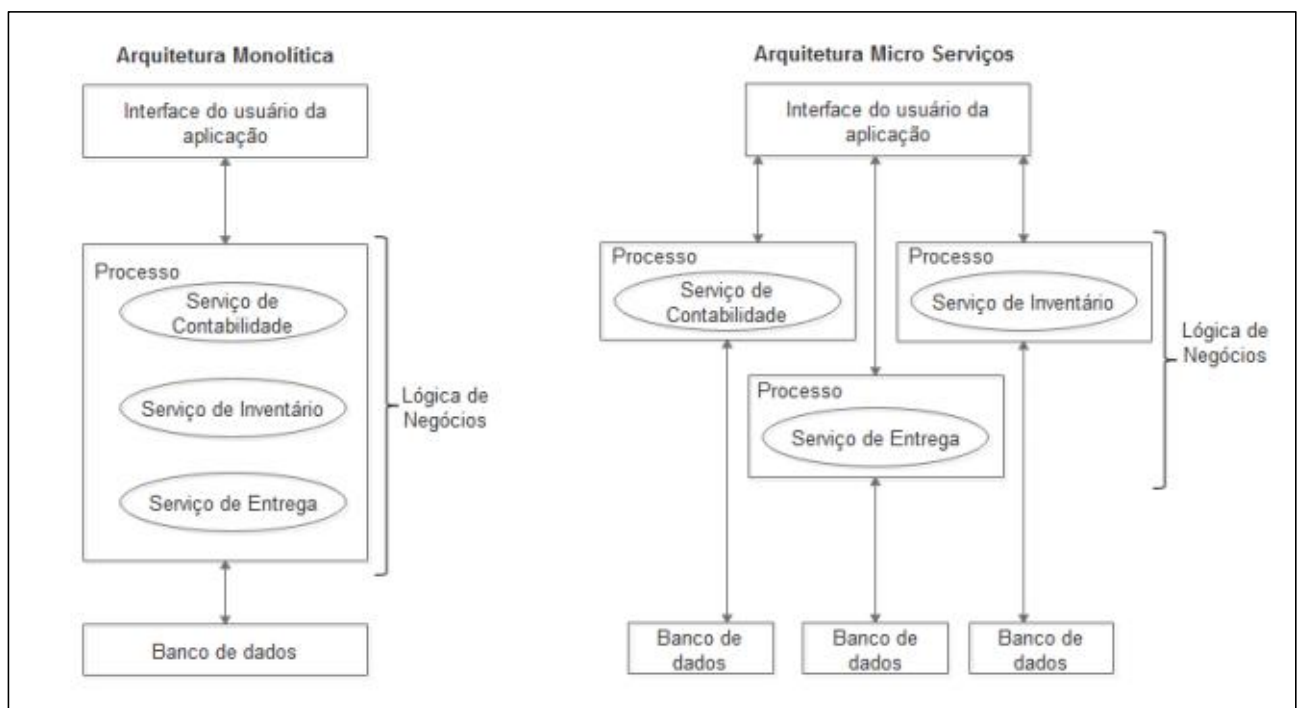


Figura 9: Diferença entre a arquitetura monolítica e a arquitetura de microsserviços
Fonte: Altexsoft, 2017

Na Figura 9 apresenta a diferença de arquitetura monolítica e de microsserviço, tendo como diferencial a separação e unicidade de cada lógica de negócio, comprovando as vantagens e desvantagens citadas por Moreira e Beder (2016).

4. PROPOSTA DO TRABALHO

Neste capítulo apresenta as principais tecnologias que serão utilizadas para o desenvolvimento dos aplicativos. Também será apresentado a proposta de desenvolvimento dos microsserviços e as ideias pensadas para os aplicativos. Ao final deste capítulo se obtém o entendimento das funcionalidades principais pensadas para compor o aplicativo e no seu valor de entrega para o usuário final.

4.1 TECNOLOGIAS

O objetivo desta sessão é dar cobertura as principais tecnologias que foram utilizadas no desenvolvimento dos aplicativos, no intuito de pontua-las e justificar suas escolhas.

4.1.1 JAVA

A linguagem java usa de um conceito denominado máquina virtual, onde, entre o sistema operacional e a aplicação há uma camada que é responsável por interpretar os comandos escritos da aplicação e traduzir para o sistema operacional correspondente por este motivo o java é considerado multiplataforma. (CAELUM, 2016). O java será nossa linguagem para o servidor, devido sua robustez e confiança.

4.1.2 SPRING

Segundo o (SPRING, 2021) o *Spring* tem como objetivo tornar a programação java mais rápida, fácil e segura para todos. Tem como foco a velocidade, simplicidade e produtividade. Nos dias de hoje é o *framework* mais utilizado no mercado java por oferecer soluções de alto nível.

4.1.3 ANGULAR

Na documentação do (ANGULAR, 2021) o *Angular* é uma estrutura de *design* para o desenvolvimento de *front-end* de aplicativos e plataformas eficientes, sofisticados e tendo como principal característica ser página única.

“O aplicativo de página única é composto por componentes individuais que podem ser substituídos e atualizados, tendo a capacidade de recarregar a página inteira, sem a necessidade de ação do usuário.” (JADHAV et al, 2015).

4.1.4 EUREKA

Dentre as soluções fornecidas pelo *Spring*, o Eureka se faz presente. Foi criado pela Netflix para gerenciar status, localizar serviços dentro de uma rede e fazer balanceamento de carga. Devido as funcionalidades fornecidas pelo Eureka e por seu uso em companhias de sucesso como empresas Netflix e Amazon é justificado sua integração neste trabalho para comunicação entre os microsserviços.

4.2 OS APLICATIVOS

O aplicativo do usuário tem como principal função realizar compras em supermercados, feiras e mercearias cadastradas no aplicativo do cliente, esse cadastro requisita informações do estabelecimento que deseja se integrar a rede. Após o cadastro, o cliente recebe no e-mail uma mensagem de boas-vindas. O aplicativo contém um menu para ser navegado entre as opções de cadastro de produto, pedidos de compras e estoque.

No aplicativo do usuário contém uma lista de estabelecimentos cadastrados oferecendo seu catálogo de produtos. Pretendia-se que o usuário pudesse montar sua lista de estabelecimentos favoritos e cadastrar cartões. O aplicativo contém um menu horizontal na parte inferior para navegação. Nas Figuras 10 e 11 é apresentado prototipações de tela do aplicativo do usuário antes do desenvolvimento, um esboço utilizado para inspiração, mas que ilustra bem a ideia do presente trabalho

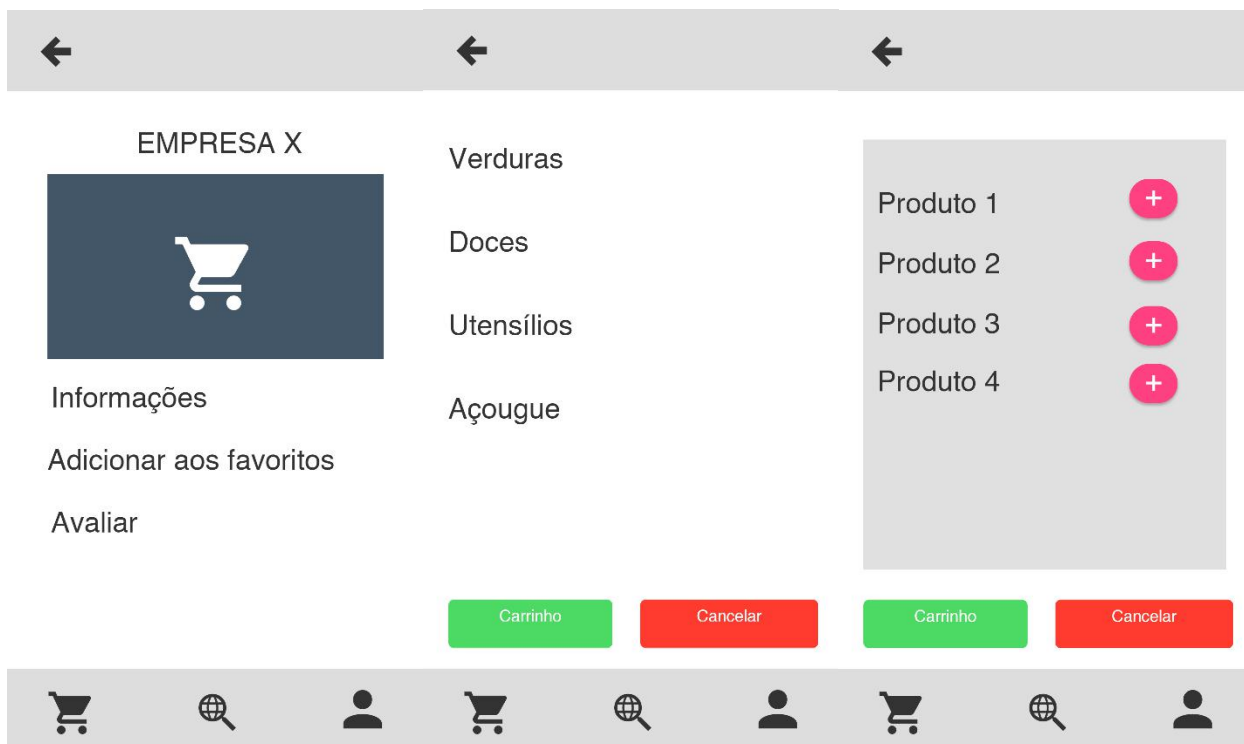


Figura 10: Prototipação de tela para realizar compra antes do desenvolvimento

Na Figura 10 é realizado um processo de compra. O usuário ao selecionar a empresa X é levado para o catálogo de produtos. Pretendia-se que o cliente pudesse cadastrar prateleiras e vincula-as aos produtos para uma melhor organização.

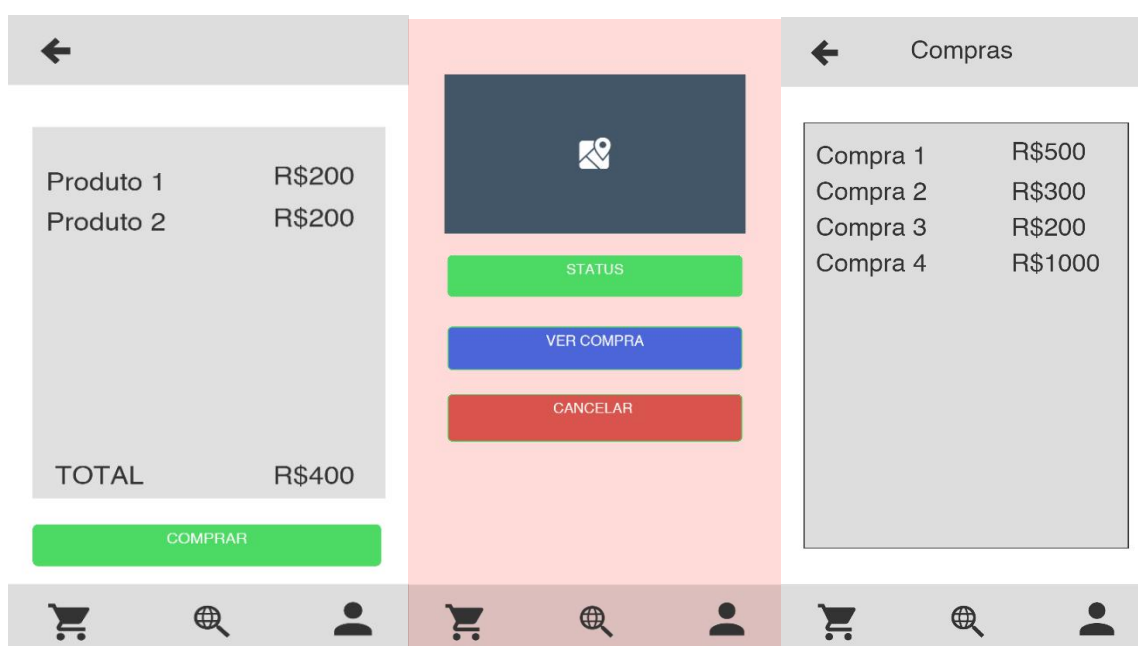


Figura 11: Prototipação de tela para finalizar compra antes do desenvolvimento

Na Figura 11 é uma continuação natural do processo da Figura 10, após selecionar os produtos, o usuário é levado para uma tela para finalizar a compra onde contém informações de valor total e quantidade de cada item. O usuário tem as opções de cancelar a compra e finalizar, no caso de o usuário não concluir a compra, o mesmo tem a funcionalidade de retomar de onde parou.

Ao término da compra, será notificado para o comerciante dentro do aplicativo do cliente um novo pedido de compra, contendo os produtos. Após a separação, o comerciante movimenta a compra disponibilizando para entrega.

Foi utilizado a arquitetura de microsserviços para o desenvolvimento, que tem como característica a divisão de atividades, além de ajudar na manutenção, essa estrutura contribui também para facilidade para implementação de novas funcionalidades, flexibilidade de infraestrutura, escalabilidade, redução de dívida técnica e desenvolvimento de práticas DevOps.

Para atender as demandas dos aplicativos, foram criados 7 serviços, cada um com sua responsabilidade e armazenamento, tais como o serviço para cadastrar produto, o serviço de estoque para gerenciar a quantidade disponível, o de compra que carrega dados do processo como status e data da compra, além de identificar o cliente e usuário e o de carrinho que contém os produtos, este é integrado ao um id de compra para identificação. Pretendia-se criar serviços de pagamento.

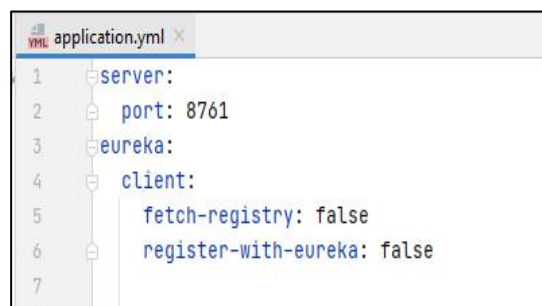
5. DESENVOLVIMENTO

Neste capítulo será descrito e apresentado os microsserviços desenvolvidos tanto como sua infraestrutura para funcionamento e objetivo.

5.1 MICROSSERVIÇOS

Para a comunicação entre os microsserviços foi utilizado uma solução por meio de dependência no *maven*, trata-se do Eureka, um serviço *REST* para localizar serviços. Criado pela Netflix, o Eureka tem como objetivo: gerenciar status, localizar os microsserviços na rede e administrar balanceamento de carga.

A configuração é feita da seguinte maneira, cria-se um projeto java para ser o servidor do Eureka, além das dependências, é necessário inserir informações no arquivo de configuração do projeto, conforme apresenta a Figura 12. Também é necessário inserir na classe *main* do projeto a anotação `@EnableEurekaServer`



```
1 server:
2   port: 8761
3 eureka:
4   client:
5     fetch-registry: false
6     register-with-eureka: false
7
```

Figura 12: Application.yml do Eureka

Para um outro serviço se registrar no servidor é necessário inserir informações no arquivo de configuração do projeto, como apresenta a Figura 13 e inserir a anotação `@EnableFeignClients` no método *main* do serviço que está se registrando.


```

spring:
  application:
    name: 'produto'

eureka:
  client:
    fetch-registry: true
    register-with-eureka: true
    service-url:
      defaultZone: ${EUREKA_SERVER:http://localhost:8761/eureka}

```

Figura 13: Application.yml exemplo dos serviços

O balanceamento de carga ocorre quando o Eureka encontra um mesmo serviço presente em duas portas diferentes. Para que o Eureka consiga fazer o balanceamento, é necessário injetar na classe *main* do serviço registrado, um método fornecido pela dependência do Spring Cloud. A Figura 14 resume em imagem os requisitos necessários, tanto como anotações como método, para habilitar a inteligência de balanceamento de carga sobre o serviço

```

@SpringBootApplication
@EnableFeignClients
public class ProdutoApplication {
    @Bean
    @LoadBalanced
    public RestTemplate getRestTemplate() { return new RestTemplate(); }

    public static void main(String[] args) { SpringApplication.run(ProdutoApplication.class, args); }
}

```

Figura 14: Main exemplo dos serviços para habilitar o Eureka

Estando um serviço registrado no Eureka, para realizar requisições a outros serviços, é criada uma interface que contém a anotação `@FeignClient("nome do serviço que você quer se comunicar")`, nessa interface são criados métodos responsáveis por chamadas REST a outros serviços, conforme é apresentado na Figura 15. Para que tenha sucesso as requisições, é necessário que ambos os serviços estejam registrados no servidor.

```
@FeignClient("estoque")
public interface EstoqueClient {

    @RequestMapping(method = RequestMethod.POST, value = "/estoque")
    void post(List<Integer> idsProdutos);

    @RequestMapping("/estoque/quantidade/{idProduto}")
    EstoqueDTO getEstoqueProduto(@PathVariable Integer idProduto);

    @RequestMapping(method = RequestMethod.DELETE, value = "/estoque/{idProduto}")
    EstoqueDTO deletaEstoque(@PathVariable Integer idProduto);
}
```

Figura 15: Interface exemplo de comunicação dos serviços

Foram desenvolvidos 7 serviços para atender as demandas dos aplicativos, os mesmos são fracamente acoplados e seguindo a recomendação dos estudos de microsserviços, cada um contém sua base no banco de dados. A Figura 16 ilustra a arquitetura dos serviços.

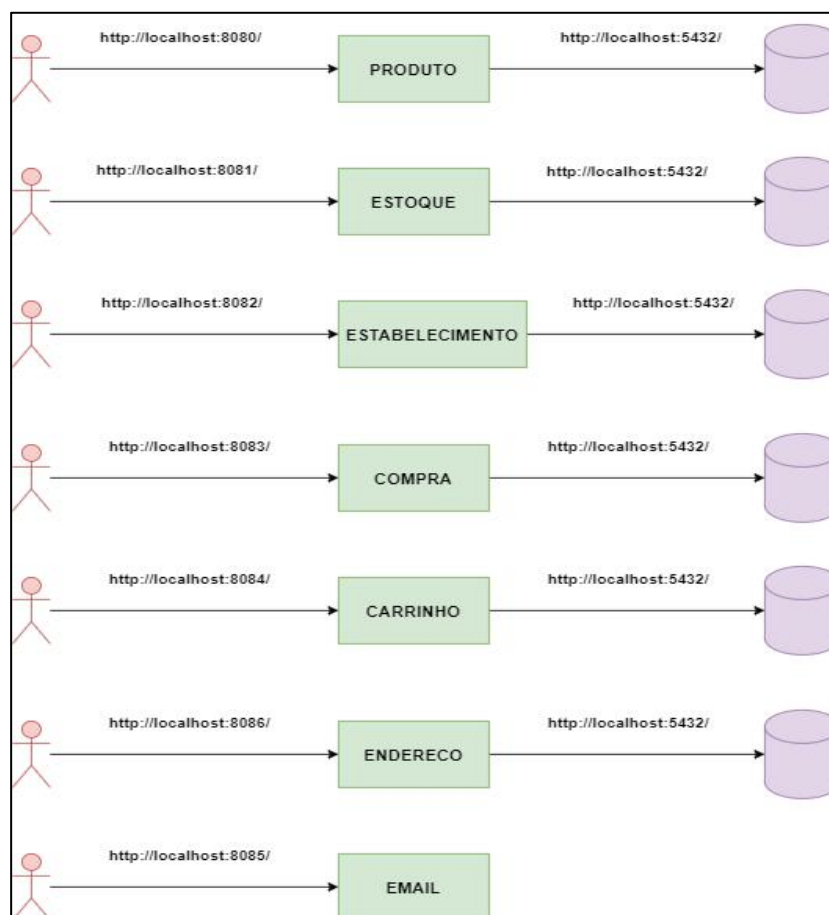


Figura 16: As portas dos microserviços

O serviço de produto tem por função armazenar os dados do mesmo, tais como nome e valor. A entidade persistida no banco guarda o id do cliente no momento do cadastro para que saiba identificar quem é o dono, neste processo descrito, o serviço de estoque é chamado, cadastrando o id do produto com quantidade zero. O serviço de estoque tem como objetivo gerenciar a quantidade dos produtos, sempre que criado um novo produto é também criado um estoque, outra função desse serviço é fazer a baixa de estoque no momento da compra. A imagem 17 ilustra este processo.

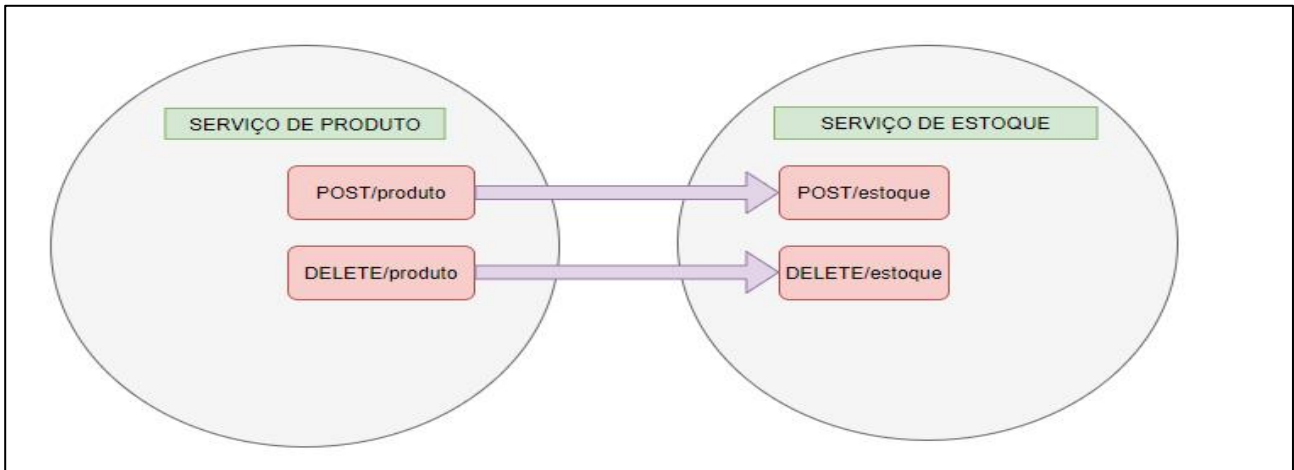


Figura 17: Diagrama para entendimento de funcionalidade entre produto e estoque

O serviço do estabelecimento armazena as informações do negócio, como nome do estabelecimento, cnpj e dados de autenticação como email e senha, contém toda a lógica de autenticação usando o *Spring Security*. Este serviço contém um método que busca por todos os estabelecimentos cadastrados para serem listados no aplicativo do usuário.

O serviço de compra é um dos processos mais importantes, neste contém informações tais como id da compra, id do cliente, id do estabelecimento, data da compra, status e o id do carrinho, este último é gerenciado pelo serviço de carrinho que contém o id do carrinho e os ids dos produtos que o usuário selecionar no aplicativo, na Figura 18 ilustra como estes dois serviços se comportam.

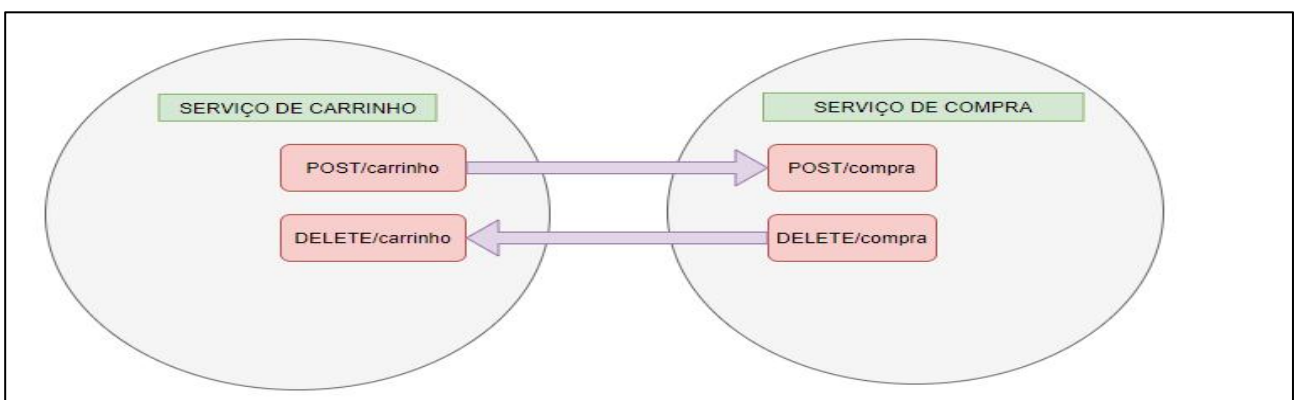


Figura 18: Diagrama para entendimento de funcionalidade entre carrinho e compra

O serviço de carrinho tem métodos para acrescentar e tirar produtos de acordo com o a movimentação do usuário no aplicativo. O serviço de compra contempla métodos de alterar status, buscar compras por status e id do estabelecimento. Quando o usuário

concluir uma compra o status dela vai ser alterado para pedido feito e nesse momento ocorre a baixa no estoque, a Figura 19 ilustra o processo em que o serviço de compra faz uma requisição passando o id do carrinho para o serviço de estoque, o mesmo faz uma outra requisição no serviço de carrinho recuperando as informações necessárias para dar baixa no estoque.

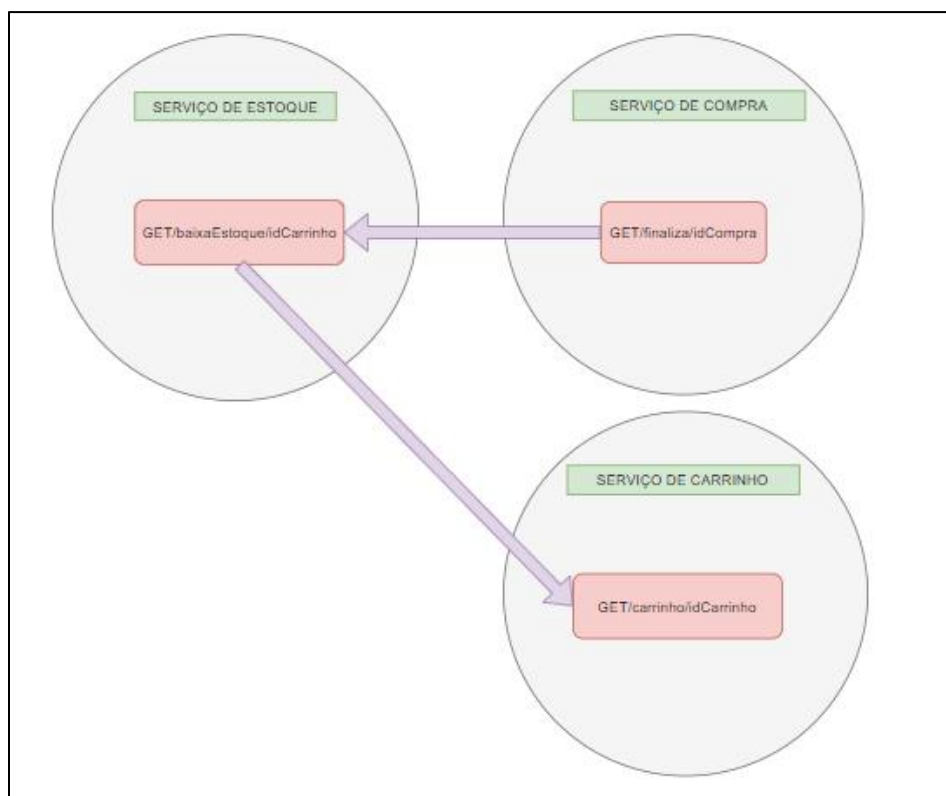


Figura 19: Diagrama de entendimento para a funcionalidade da baixa de estoque

O serviço de email tem por objetivo enviar uma mensagem de boas vindas assim que o cliente se cadastrar no aplicativo. O endereço é um serviço criado para gerenciar os locais de entrega.

5.2 CONTÊINERS DOS SERVIÇOS

Para executar os passos que serão descritos foi necessário configurar o *settings.xml* do *maven* e adicionar plugins no *pom.xml* de todos os projetos como ilustra a Figura 20.

```

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </plugin>
    <plugin>
      <groupId>com.spotify</groupId>
      <artifactId>dockerfile-maven-plugin</artifactId>
      <version>1.4.9</version>
      <configuration>
        <repository>${docker.image.prefix}/${project.artifactId}</repository>
      </configuration>
    </plugin>
  </plugins>
</build>
<?xml version="1.0" encoding="UTF-8"?>
<settings
  xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <settings>
    <pluginGroups>
      <pluginGroup>com.spotify</pluginGroup>
    </pluginGroups>
  </settings>
</settings>

```

Figura 20: ConFigurações para subir imagens no docker

Em cada serviço foi inserido na raiz do projeto um arquivo Dockerfile, o conteúdo são instruções para criar uma imagem dentro do Docker, na Figura 21 é apresentado um exemplo desse arquivo.

```

FROM adoptopenjdk/openjdk11:alpine
VOLUME /tmp
ARG JAR_FILE=target/produto-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]

```

Figura 21: Dockerfile exemplo

O *FROM* indica qual a imagem base será usada, para o presente trabalho é uma máquina *Linux Alpine* com a *Open JDK 11* já instalados. o *VOLUME* cria uma pasta temporária no contêiner com os arquivos do projeto. O *ARG* cria uma variável com o caminho do JAR gerado do projeto. O *COPY* faz uma cópia do arquivo JAR com o nome *app.jar* e o

ENTRYPOINT define os comandos que serão executados dentro do contêiner. Para cada projeto foi necessário executar o comando *npm clean install dockerfile:build* nas raízes do projeto para ser gerado as imagens no Docker.

Com as imagens já dentro do Docker, foi usado o *docker compose* para subir as máquinas, para tal ação é necessário criar um arquivo *docker-compose.yml* e nesse arquivo é especificado a imagem, as portas e a network que serão utilizadas, contém também neste arquivo a imagem do banco de dados. A Figura 22 apresenta o resultado após executar o comando *docker-compose up*. As máquinas estão prontas para serem consumidas pelos dois aplicativos.

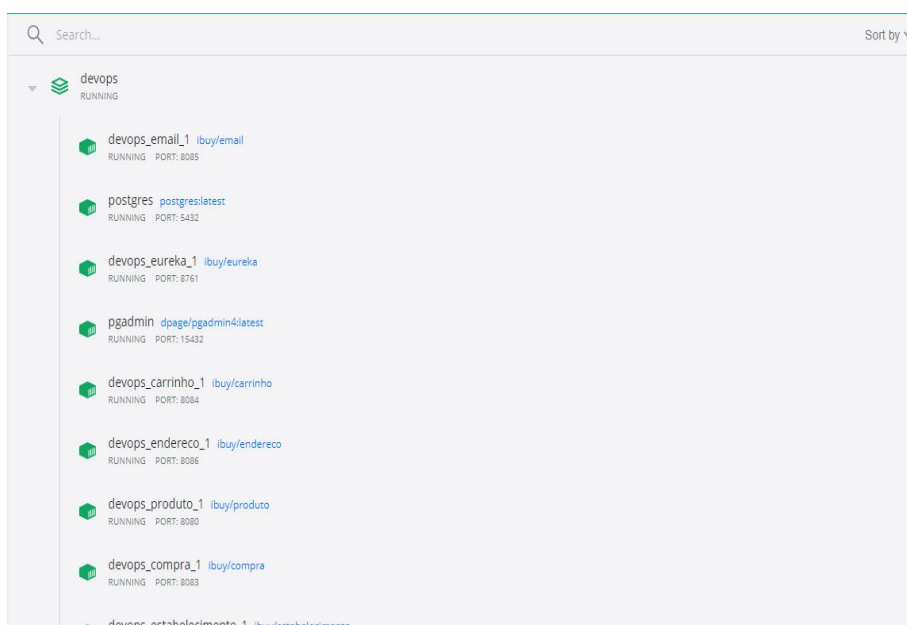


Figura 22: Contêineres das imagens criadas no docker

5.3 OS APLICATIVOS

O objetivo dessa seção é apresentar as principais telas dos dois aplicativos como resultado do desenvolvimento. Na Figura 23 é apresentada a tela de login e cadastro de estabelecimento no aplicativo do cliente. Para o cadastro são necessárias quatro informações, o nome do estabelecimento, este que será apresentado para o usuário no aplicativo de compras, cnpj, e-mail e senha, estes dois últimos são importantes para entrar no aplicativo. Após o cadastro, um e-mail de boas-vindas será enviado.

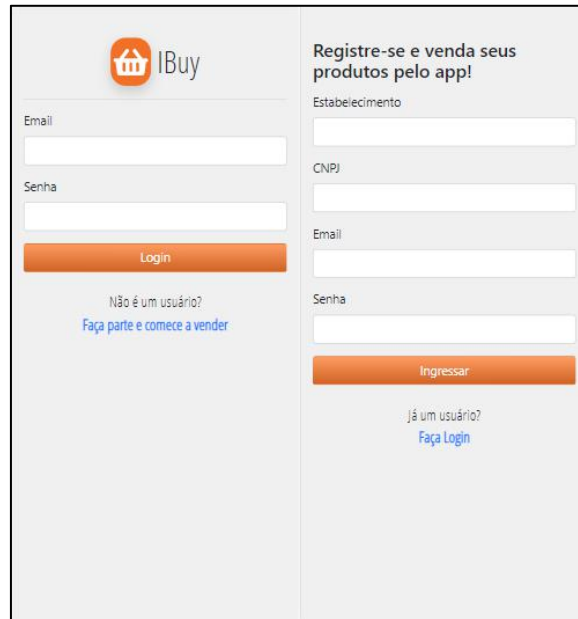


Figura 23: Tela de login e cadastro de estabelecimento

Após o cliente fazer o cadastro do estabelecimento e entrar na aplicação, será direcionado para o cadastro de produtos como é apresentado na Figura 24. Nesta primeira versão, o aplicativo do cliente contém um conjunto de botões posicionado na parte superior para navegar na aplicação. Para o cadastro do produto são necessárias duas informações: descrição e preço. Após a inserção, o usuário será direcionado para a listagem dos produtos, para cada produto dentro da lista, estão disponíveis as opções de editar e excluir produto como também é apresentado na Figura 24.

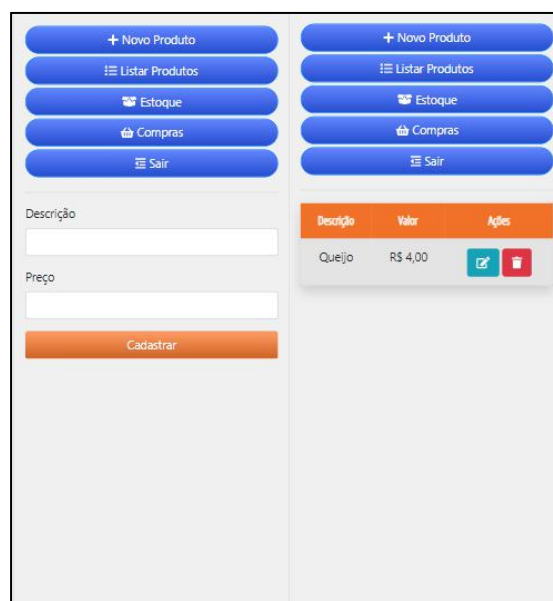


Figura 24: Tela de cadastro do produto e seu gerenciamento

Quando é cadastrado um produto, também é criado um estoque para o mesmo, tendo como quantidade inicial 0. A Figura 25 apresenta a tela de estoque, onde o cliente pode editar a quantidade do produto inserindo o seu valor no campo a frente da descrição, esta tela não contém um botão para confirmar a alteração, foi desenvolvido para que, perdendo o foco do campo, envie uma requisição de alteração com o valor informado pelo cliente.



Figura 25: Tela de estoque

A Figura 26 apresenta a tela inicial do aplicativo do usuário, este contém um menu horizontal na parte inferior para navegação. O ícone do carrinho direciona para a tela de compras não concluídas, esta funcionalidade foi desenvolvida para o caso de o usuário não concluir a compra em um primeiro momento. O ícone da lupa direciona para tela apresentada na Figura 26, onde é listado todos os estabelecimentos cadastrados. O ícone do usuário direciona para cadastros de informações importantes para a realização da compra, tais como endereços e cartões.



Figura 26: Tela inicial do aplicativo do usuário

A Figura 27 apresenta duas telas no aplicativo do usuário: lista de produtos do estabelecimento e carrinho. Após a escolha do estabelecimento para a realização da compra, o usuário é direcionado para a lista de produtos, no qual, para cada item tem-se a opção de adicionar ou retirar uma unidade do carrinho. Está disponível na parte superior do aplicativo um botão de cancelar a compra, caso seja o desejo do usuário. A tela de carrinho apresenta informações da compra, tais como: valor total e quantidade de cada item. Para concluir o processo o usuário deve escolher a opção de finalizar a compra, disponível na parte superior da aplicação.

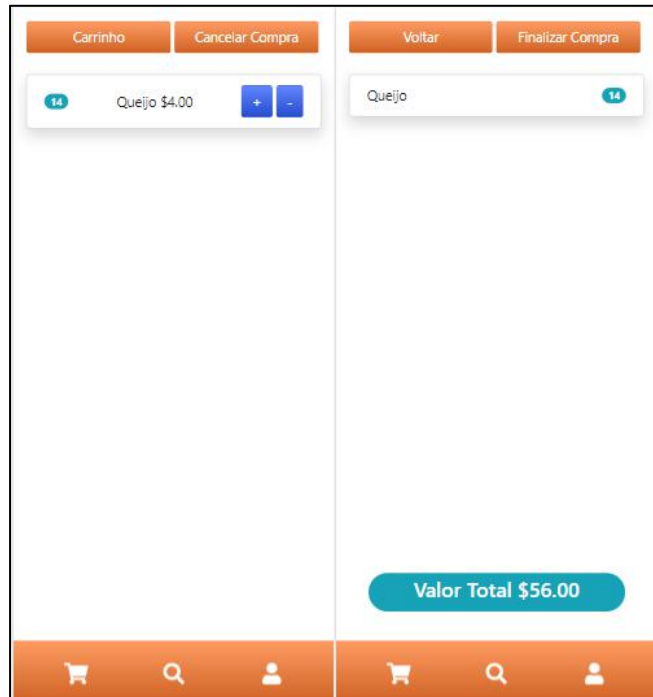


Figura 27: Tela de compra e carrinho

Nos cadastros das informações para compras do usuário, está presente o cadastro de endereços. Após inserir o CEP o aplicativo preenche os campos da rua, bairro, cidade e estado, apenas exigindo do usuário a inserção do numero e um apelido para o endereço.

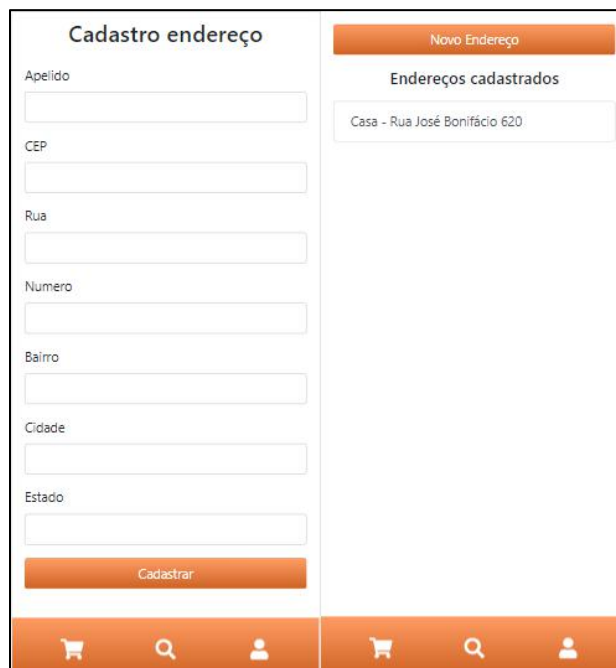


Figura 28: Tela de cadastro de endereços

O estabelecimento tem disponível dentro da sua aplicação uma lista de pedidos de compras feitos por usuários, como apresenta a Figura 29. Após o cliente escolher a opção de separar produtos, será direcionado para a tela que contém lista de produtos e suas respectivas quantidades. Para finalizar a compra, o cliente seleciona a opção de pronto para entrega.

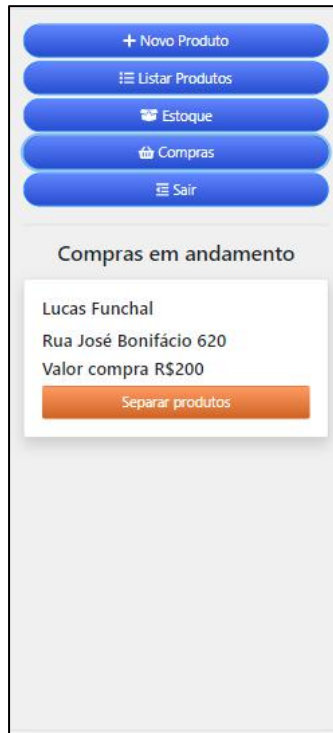


Figura 29: Tela de compras no aplicativo do cliente

6. CONCLUSÃO

O principal objetivo deste projeto de pesquisa foi explorar tanto como estudo como nas etapas de implementação, o uso de microsserviços no desenvolvimento de aplicações que promovam a transformação digital. Algo que superou as expectativas e também a proposta do trabalho foi o estudo de serviços em nuvem por meio da criação de uma instância EC2 na *Amazon Web Services* e também da utilização da ferramenta de integração contínua o Jenkins.

6.1 TRABALHOS FUTUROS

O projeto aplicou um ponto muito forte para o tema da transformação digital que são os microsserviços, porém ainda há muito que progredir para alcançar o cenário ideal. Para trabalhos futuros, pretende-se incrementar novas funcionalidades, inserir no contexto de software em nuvem, implementar pipelines para integração contínua e métricas para monitorar a qualidade do desenvolvimento para um feedback rápido e conseqüentemente ter melhorias contínuas. As aplicações foram direcionadas para o desenvolvimento Web baseadas em microsserviços, pretende-se em trabalhos futuros direcionar os esforços para o desenvolvimento de aplicações mobile.

REFERÊNCIAS

ALTEXSOFT. **Using Microservices for Legacy System Modernization**. 2017. Disponível em: <<https://www.altexsoft.com/blog/engineering/using-microservices-forlegacy-system-modernization/>>. Acesso em: 10 de Janeiro de 2021.

ANGULAR. **Introduction to the Angular Docs**. Disponível em <<https://angular.io/docs>>. 2021. Acesso em: 17 de fevereiro de 2021.

CAELUM. **Java e Orientação a Objetos**. Disponível em <<https://www.caelum.com.br/apostila-java-orientacao-objetos/>>. Acesso em: 16 de fevereiro de 2021.

CHAPPELL, D. A.; JEWELL, T. **Java Web Services**. 1ª edição. Estados Unidos: O'Reilly Media, 2002.

COMPUTERWORLD. **Transformação digital avança e desafia os CIOs**. Perspectiva. São Paulo, dezembro 2017. p. 4-6. Conjunto de Conhecimentos em Gerenciamento de Projetos, 6ª Ed. Pennsylvania: PMI, 2017.

DINGSØYR, Torgeir et al. **A decade of agile methodologies**: Towards explaining agile software development. 2012. Em <<https://revistaoste.com/a-verdade-por-tras-dos-numeros-da-covid-19/>>. Acesso em: 10 de novembro de 2020.

FERREIRA, Líndice Thiengo. **TRANSFORMAÇÃO DIGITAL: Aplicações e limitações de seu uso em Empresas de Seguro no Brasil**. 2018. 155. Dissertação (Mestrado). Escola Brasileira de Administração Pública e de Empresas. Rio de Janeiro, 2018.

FLUTTER. **Feito por Google**. Disponível em: <<https://flutter.dev>>. 2021. Acesso em 17 de fevereiro de 2021.

GISOLFI, Dan. **Web services architect, Part 2: Models for dynamic e-business**. Disponível em:<<https://www.ibm.com/developerworks/library/ws-arc2/>>. Acesso em: 09 de março de 2021

HUMBLE, Jez; FARLEY, David. **Continuous delivery: reliable software releases through build, test, and deployment automation**. Pearson Education, 2010.

IBM. **Web Services architecture overview.** Disponível em:<
<https://www.ibm.com/developerworks/webservices/library/w-ovr/>. Acesso em 09 de março de 2021.

IMD. **World Digital Competitiveness Ranking 2020.** Disponível em:
<<https://www.imd.org/search/searchresults/?Term=World%20Digital%20Competitiveness%20Ranking%202020>>. Acesso em: 06 de novembro de 2020.

JADHAV, Madhuri A et al. **Single Page Application using AngularJS.** International Journal of Computer Science and Information Technologies. 2015. Vol 6, p. 2876.

KHAN, Shahyan. **Leadership in the Digital Age - a study on the effects of digitalization on top management leadership.** Stockholm University, Faculty of Social Sciences, Stockholm Business School, Management & Organisation. Independent thesis Advanced level (degree of Master). 2016

KIM, Gene et al. **The DevOps handbook: how to create world-class agility, reliability, and security in technology organizations.** IT Revolution, 2016.

LAVADO, Thiago. **Uso da internet no Brasil cresce, e 70% da população está conectada.** Disponível em
<<https://g1.globo.com/economia/tecnologia/noticia/2019/08/28/uso-da-internet-no-brasil-cresce-e-70percent-da-populacao-esta-conectada.ghtml>>. Acesso em: 16 de fevereiro de 2021.

LEAL, Paula. **A VERDADE POR TRÁS DOS NÚMEROS DA COVID-19.** Disponível

SAUVAGE, Marc. Définition: **Transformation Digitale en 2019 et ses enjeux + exemple.** Disponível em < <https://www.inboundvalue.com/blog/que-signifie-la-transformation-digitale-en-2000-mots.>> Acesso em: 15 de fevereiro de 2021

MELL, Petter; GRANCE, Timothy. **The NIST Definition of Cloud Computing.** NIST Special Publication, 2011.

MOREIRA, P. F. M.; BEDER, D. M. **Desenvolvimento de aplicações e micro serviços: Um estudo de caso.** Revista TIS, v. 4, n. 3, 2016.

MOSER, A.; JÚNIOR, A. K. **Competências para a transformação digital: Desafios para a educação superior**. Metodologias e Aprendizado, v. 1, p. 115 - 122, 11 de março de 2020.

NEWMAN, Sam. **Building Microservices**. O'Reilly Media, Inc., 2015

PIOVESAN, Lia Denize. **Análise exploratória da transformação digital nos processos das cadeias de suprimento**. 2020. 257. Dissertação (Mestrado). Universidade do Vale do Rio dos Sinos. São Leopoldo, 2020.

PMI - **PROJECT MANAGEMENT INSTITUTE**. Guia PMBOK®: Um Guia para o

POLETTI, Charlie. **DevOps, pequenos passos!** Disponível em <<https://medium.com/@cpoletti/devops-pequenos-passos-80c155cb42dc>>. Acesso em: 16 de fevereiro de 2021.

PYMNTS. **The Emerging Post-COVID-19 Consumer: mapping the evolution of consumer's shifting payment preferences**. Disponível em <<https://www.pymnts.com/study/the-emerging-post-covid-19-consumer-ecommerce-coronavirus/>>. Acesso em: 07 de novembro de 2020.

RICCI, A.; DENTI, E.; PIUNTI, M. **A platform for developing SOA/WS applications as opens and heterogeneous multi-agent systems**. In: Multiagent and Grid Systems. Amsterdã, Holanda: IOS Press, 2010. v.6, n.2, p. 105-132. ISSN: 1574-1702.

ROGERS, David. **Transformação Digital – Repensando o seu negócio para a era digital**. Tradução: Afonso Celso da Cunha Serra. São Paulo: Autêntica Business, 2017.

SCHWABER, Ken. **Agile project management with Scrum**. Microsoft press, 2004.

SPRING. **Why Spring?** Disponível em <<https://spring.io/why-spring>>. 2021. Acesso em: 16 de fevereiro de 2021.

TIMOTEO, Aline L.; LIMA, Manoel Eusébio De. **AgILEs: uma abordagem ágil para desenvolvimento de projetos de hardware e software integrados**. 2009. Dissertação (Mestrado). Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2009.

UFMG – Faculdade de Medicina de Minas Gerais. **Saiba onde o risco de contágio do coronavírus é maior.** Disponível em < <https://www.medicina.ufmg.br/saiba-onde-o-risco-de-contagio-do-coronavirus-e-maior/>>. Acesso em: 10 de novembro de 2020.

VIALLI, Andrea. **Brasil ganha 600 mil microempreendedores durante a quarentena.** Folha de S.Paulo, 2020. Disponível em < <https://www1.folha.uol.com.br/mpme/2020/08/brasil-ganha-600-milmicroempreendedores-durante-a-quarentena.shtml> >. Acesso em: 29 de outubro de 2020.