



UNISUL

**UNIVERSIDADE DO SUL DE SANTA CATARINA
ANDRÉ VITOR RAMPANELLI
JEFFERSON DAMACENO BRUCHADO**

**SELLVICE:
PROTÓTIPO DE APLICAÇÃO DE OFERTA DE SERVIÇOS**

Palhoça
2020

**ANDRÉ VITOR RAMPANELLI
JEFFERSON DAMACENO BRUCHADO**

**SELLVICE:
PROTÓTIPO DE APLICAÇÃO DE OFERTA DE SERVIÇOS**

Trabalho de Conclusão de Curso apresentado ao Curso de graduação em sistemas de informação da Universidade do Sul de Santa Catarina como requisito parcial à obtenção do título de Bacharel.

Orientador: Prof. Saulo Popov Zambiasi, Dr.

Palhoça
2020

**ANDRÉ VITOR RAMPANELLI
JEFFERSON DAMACENO BRUCHADO**

**SELLVICE:
PROTÓTIPO DE APLICAÇÃO DE OFERTA DE SERVIÇOS**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel e aprovado em sua forma final pelo Curso de graduação em sistemas de informação da Universidade do Sul de Santa Catarina.

Palhoça, 02 de Dezembro de 2020.

Professor e orientador Saulo Popov Zambiasi, Dr.
Universidade do Sul de Santa Catarina

Prof. Theo A. Luz, Me.
Universidade do Sul de Santa Catarina

Prof. M. Inés Castiñeira, Dra.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Agradecemos a nossa família pelo suporte e total apoio até o término de nosso curso, ao professor orientador pela oportunidade e aos colegas que nos ensinaram com excelência e dedicação.

“O homem não teria alcançado o possível se, repetidas vezes, não tivesse tentado o impossível.” (MAX WEBER).

RESUMO

A partir de 1990, o avanço tecnológico, as flexibilizações, as desregulações e mudanças econômicas dentro da sociedade moderna trouxe um nível de disponibilidade tecnológica gigantesco, propondo inúmeras oportunidades não exploradas para criação de aplicações e ferramentas. Uma grande quantidade de pessoas que são infelizes com os seus trabalhos ou a renda recebida. Assim, abrem-se oportunidades não exploradas para a criação de aplicações, tais como Uber, 99pop, Rappi e muitas outras. Uma dessas oportunidades é explorada nessa monografia, pois junto com o aumento da informalidade no trabalho as pessoas buscam um complemento de renda. A aplicação desenvolvida neste trabalho é uma plataforma onde é possível ofertar e consumir serviços de outras pessoas, aproveitando as oportunidades do mercado, com foco em auxiliar pessoas a terem uma renda ou auxiliar. A ideia aqui proposta ganhou ainda mais notoriedade com o período de pandemia da covid-19, pois diversas pessoas perderam seus empregos, aumentando o índice de desempregados. Com a vasta quantidade de tecnologias para desenvolvimento, foram escolhidos e abordados Vue.JS, um framework com alta porcentagem de aceitação entre os desenvolvedores e Firebase Realtime Database, ótimo para criação de aplicações MVP e rápido desenvolvimento.

Palavras-chave: Informalidade. Ofertas. Serviços. Aplicativo. VueJS.

ABSTRACT

From 1990 onwards, the advancement of technology, the flexibility, deregulations, and economic changes within modern society has brought a gigantic level of technological availability that proposes countless unexplored opportunities for creating applications and tools. A large amount of people is unhappy with their jobs and the income received. So many opportunities for creating applications open up, as Uber, 99pop, Rappi and many other. One of those opportunities is explored in this monograph, along with the increase in informality at work, people are looking for an income supplement. The idea proposed here grew up with the covid-19 outbreak, because many people lost their jobs, increasing the unemployment rate. With the vast amount of technologies for development, Vue.JS was chosen, a framework with a high percentage of acceptance among developers and Firebase Realtime Database, great for creating MVP applications and fast development.

Keywords: Informality. Offers. Services. App. VueJS.

LISTA DE ILUSTRAÇÕES

Figura 1 – Variação de pessoas ocupadas em relação ao local de trabalho no setor privado entre 2016-2017 e 2017-2018	13
Figura 2 – Site da plataforma MercadoLivre.	16
Figura 3 – Site da plataforma Enjoei.	17
Figura 4 – Site da plataforma GetNinjas.....	18
Figura 5 – Camadas da engenharia de software.	21
Figura 6 – Processo de Software genérico.	23
Figura 7 – Gráfico de Produtividade x Tempo.....	25
Figura 8 – Desenho do funcionamento do sistema.....	33
Figura 9 – Ator do sistema	36
Figura 10 – Diagrama de caso de uso (fluxo de busca de serviço)	51
Figura 11 – Diagrama de Caso de uso (fluxo de contratação de serviço)	52
Figura 12 – Diagrama de Caso de Uso (fluxo de venda de serviço).....	53
Figura 13 – Protótipo da tela de Autenticação	54
Figura 14 – Protótipo da tela de Cadastro.....	55
Figura 15 – Protótipo da tela de listagem de serviços	56
Figura 16 – Protótipo da tela de detalhes de serviço.....	57
Figura 17 – Protótipo da tela de criação.....	58
Figura 18 – Protótipo da tela de edição de serviço	59
Figura 19 – Tecnologias utilizadas	60
Figura 20 – Tela de Login	64
Figura 21 – Tela de “Esqueci a senha”.....	65
Figura 22 – Tela de Cadastro	66
Figura 23 – Tela de listagem de serviços	67
Figura 24 – Tela de detalhes do serviço.....	68
Figura 25 – Tela redirecionada para o <i>whatsapp</i>	69
Figura 26 – Tela de detalhes do serviço.....	70
Figura 27 – Questão número um.....	73
Figura 28 – Questão número dois	73
Figura 29 – Questão número três.....	73
Figura 30 – Questão número quatro.....	74

LISTA DE QUADROS

Quadro I – Comparativo entre sistemas	19
Quadro II – Comparativo completo entre sistemas	35
Quadro II – Definições do ator usuário	36
Quadro IV – Quadro de requisitos funcionais.....	37
Quadro V – Requisitos não funcionais	38
Quadro VI – Requisitos não funcionais (desempenho).....	38
Quadro VII – Requisitos não funcionais (desenvolvimento)	39
Quadro VIII – Requisitos não funcionais (sistemas operacionais)	39
Quadro IX - Requisitos não funcionais (operabilidade).....	39
Quadro X – Requisitos não funcionais (confiabilidade)	40
Quadro XI – Regras de negócio.	40
Quadro XII – Caso de Uso (cadastro)	41
Quadro XIII – Caso de Uso (autenticação).....	42
Quadro XIV – Caso de Uso (visualizar serviços)	43
Quadro XV – Caso de Uso (detalhes do serviço)	44
Quadro XVI – Caso de Uso (contatar o provedor do serviço)	45
Quadro XVI – Caso de Uso (contratar o serviço).....	45
Quadro XVIII – Caso de Uso (visualizar histórico de pedidos).....	46
Quadro XIX – Caso de Uso (visualizar pedidos em abertos)	47
Quadro XX – Caso de Uso (responder solicitação do serviço)	48
Quadro XXI – Caso de Uso (aprovação mútua).....	49
Quadro XXII – Quadro das perguntas do questionário	72
Quadro XXIII – Quadro com melhores sugestões.....	75
Quadro XXIX – Análise dos resultados	85

SUMÁRIO

1	INTRODUÇÃO	12
1.1	PROBLEMÁTICA	13
1.2	OBJETIVOS	13
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	14
1.3	JUSTIFICATIVA	14
1.4	ESTRUTURA DO TRABALHO	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	ESTADO DA ARTE	16
2.1.1	MercadoLivre	16
2.1.2	Enjoei	17
2.1.3	OLX	17
2.1.4	GetNinjas	18
2.1.5	Comparativo entre plataformas	19
2.2	FUNDAMENTOS UTILIZADOS PARA O DESENVOLVIMENTO	19
2.3	SOFTWARE	20
2.4	ENGENHARIA DE SOFTWARE	21
2.4.1	Processos de Software	22
2.5	DESENVOLVIMENTO DE SOFTWARE	24
2.5.1	Análise de Requisitos de Software	24
2.5.2	CLEAN CODE (CÓDIGO LIMPO)	24
2.5.3	Design Pattern (padrão de projeto)	26
2.5.4	Plataforma RESTful	27
2.6	INTERFACE	27
2.6.1	Usabilidade	28
2.6.2	Interface Humano-Computador (IHC)	28
2.7	CODIFICAÇÃO	28
2.8	TESTES	29
2.9	CASOS DE USOS	29
3	MÉTODO	31
3.1	TIPO DE PESQUISA	31
3.2	ETAPAS METODOLÓGICAS	31
3.3	SOLUÇÃO PROPOSTA	32
3.4	DELIMITAÇÕES	33
4	MODELAGEM DO SISTEMA	35
4.1	ANÁLISE DO ESTADO DA ARTE	35
4.2	ATORES	36
4.3	REQUISITOS	37
4.3.1	Requisitos funcionais	37
4.3.2	Requisitos não funcionais	38
4.4	REGRAS DE NEGÓCIO	40
4.5	CASOS DE USO	41
4.6	PROTOTIPAÇÃO DE TELAS	53
4.6.1	Protótipo da tela de Autenticação	54
4.6.2	Protótipo da tela de Cadastro	55
4.6.3	Protótipo da tela de listagem de serviços	56
4.6.4	Protótipo da tela de detalhes de serviço	56
4.6.5	Protótipo da tela de criação e edição de um serviço	57

5 DESENVOLVIMENTO	60
5.1 TECNOLOGIAS E FERRAMENTAS	60
5.1.1 Firebase	61
5.1.1.1 Firebase Realtime Database.....	61
5.1.2 JSON	62
5.1.3 GitHub	62
5.1.4 Visual Studio Code	62
5.1.5 Vue.JS	63
5.2 HISTÓRICO DO DESENVOLVIMENTO	63
5.3 APLICATIVO DESENVOLVIDO.....	64
5.3.1 Tela de Login	64
5.3.2 Tela de Cadastro	65
5.3.3 Tela de listagem de serviços	66
5.3.4 Tela de detalhes do serviço	67
5.3.5 Tela de Criação e Edição de serviço	69
5.4 AVALIAÇÃO DO APLICATIVO	70
5.4.1 Cenário do questionário	70
5.4.2 Elaboração do questionário	71
5.4.3 Aplicação do questionário	72
5.4.4 Análise dos resultados	72
5.4.5 Considerações da avaliação	72
6 CONCLUSÕES E TRABALHOS FUTUROS	76
6.1 TRABALHOS FUTUROS	76
REFERÊNCIAS	78
ANEXOS	81
ANEXO A – QUESTIONÁRIO	82
ANEXO B – RESULTADO DA PESQUISA	84

1 INTRODUÇÃO

Os anos de 1990 foram marcados por grandes mudanças no que diz respeito a economia e ao trabalho. Entretanto, este crescimento trouxe informalidade em demasia. Houve consequências referentes ao número de assalariados, pessoas com trabalhos sem carteira assinada ou trabalhando por conta própria.

Ocorreu um aumento de contingente de pessoas que iniciaram uma atividade laboral, porém, em condição de informalidade, atingindo recorde nacional. O número de pessoas que trabalham para empresas privadas sem carteira assinada aumentou, assim como o número de trabalhadores por conta própria. (NITAHARA, 2019).

Esse aumento surgiu devido à insatisfação no trabalho, na remuneração e até mesmo pela falta de oportunidades. (LIMA, 2017).

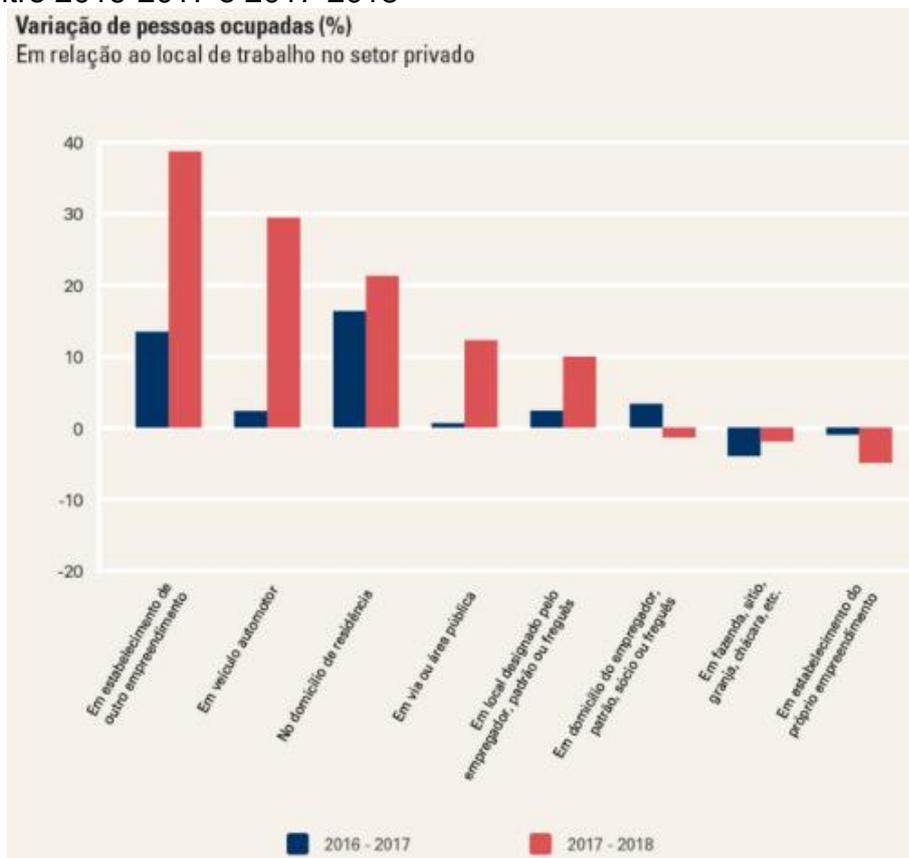
A taxa de desemprego no Brasil está cada vez menor, porém, em análise aos dados nota-se uma estabilidade entre o 1º semestre de 2020 com o último trimestre do ano de 2019, em pessoas que tem a idade para trabalhar.

Todavia, o número de pessoas que oferecem serviços em aplicações semelhantes cresceu demasiadamente. Serviços onde pessoas trabalham com veículos cresceu 29,2 por cento, um aumento de 12,1 por cento de brasileiros que trabalharam em vias públicas e 38,3 por cento de pessoas que trabalham em estabelecimentos de outro empreendimento entre o período de 2017 para 2018. (IBGE, 2012).

As recentes altas podem estar relacionadas ao crescimento dos serviços de transportes de passageiros e de entregas por aplicativos de celular, refletindo as mudanças na economia atual. (IBGE, 2012).

Em análise ao gráfico 1, vê-se a alta mencionada anteriormente, em pessoas que trabalham em estabelecimentos de outros empreendimentos, em veículo automotor, em via pública:

Figura 1 – Variação de pessoas ocupadas em relação ao local de trabalho no setor privado entre 2016-2017 e 2017-2018



Fonte: IBGE. PNAD Contínua. Características Adicionais do Mercado de Trabalho 2012-2018.

1.1 PROBLEMÁTICA

Observa-se atualmente o crescimento de aplicativos para oferta de serviços, com destaque aos automobilísticos, tais como Uber, Rappi e 99 Taxi, está desenfreado por conta do aumento da insatisfação no trabalho, na remuneração e falta de oportunidades.

Porém, é notável que tais aplicativos não atendem pessoas que possam oferecer serviços mais básicos, tais como encanador, diarista, entre outros.

Essas pessoas também podem ser afetadas pelos problemas que desencadearam o aumento dos aplicativos automobilísticos e, assim, necessitam de uma plataforma para englobá-los.

1.2 OBJETIVOS

A seguir são apresentados os objetivos gerais e específicos.

1.2.1 Objetivo Geral

A monografia tem como objetivo o desenvolvimento de um sistema *WEB* onde é possível realizar ofertar e solicitar serviços, tais como encanador, electricista etc. Seguindo padrões de projetos avançados e metodologias ágeis.

1.2.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- Realizar uma pesquisa bibliográfica sobre engenharia de software, padrões de projeto, desenvolvimento, codificação e usabilidade;
- Modelar o Sistema;
- Desenvolver o protótipo do *software*;
- Aprontar testes e avaliação da implementação.

1.3 JUSTIFICATIVA

Entende-se que a flexibilização é um fenômeno que acontece por conta de uma legislação rígida e que essa deve ser maleável, de forma a trazer benefícios aos afetados pela legislação, sejam eles empregadores ou empregados. (OLIVEIRA, 1996 apud SIQUEIRA NETO, 2012).

Entretanto, cabe ressaltar que, apesar de relativamente próximos, a flexibilização e a desregulamentação são diferentes:

A flexibilização pressupõe a intervenção estatal, ainda que básica, com normas gerais abaixo das quais não se pode conceber a vida do trabalhador com dignidade. Precisamente porque há leis em que determinados preceitos devem ser flexíveis ou estabelecer fórmulas alternativas para sua aplicação. (SÜSSEKIND, 2007, p. 14).

Desregulamentação é “a progressiva supressão de regras imperativas, com o correspondente alargamento da liberdade de estipulação” (DE PAIVA, 2000 apud SIQUEIRA NETO 2012; ACIOLLY, 2007).

Logo, compreende-se que a desregulamentação é o afastamento do Estado como responsável por defender determinada matéria, permitindo, desse modo,

a auto regulação dos afazeres entre as partes. (DE PAIVA, 2000 apud SIQUEIRA; ACIOLLY, 2007).

Nos dias atuais observa-se a diminuição da classe trabalhadora tradicional, expansão do trabalho assalariado e do setor de serviços, heterogeneização do trabalho, o desemprego estrutural em função da automação, robótica e microeletrônica.

Por conta destes atos vale o desenvolvimento do software para ter uma ferramenta de suporte para as pessoas que queiram oferecer serviços, independentemente de seu estado trabalhista, para que possa gerar renda.

1.4 ESTRUTURA DO TRABALHO

Esse trabalho está dividido em 5 capítulos. A seguir é apresentada a estrutura do trabalho:

- **Capítulo 1** – Traz aspectos introdutórios, para compreender problemas relacionados e os objetivos, seguindo para uma solução que é detalhada nos capítulos posteriores.
- **Capítulo 2** – Revisão bibliográfica com a apresentação dos conceitos, teorias e técnicas utilizadas no desenvolvimento do trabalho.
- **Capítulo 3** – Método.
- **Capítulo 4** – Modelagem do sistema.
- **Capítulo 5** – Apresentação do sistema e questionário avaliativo da solução desenvolvida.
- **Capítulo 6** - Apresentação da conclusão do trabalho baseando-se nos resultados conquistados ao longo da pesquisa e desenvolvimento.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta a fundamentação teórica utilizada pelo presente trabalho, abordando o estado da arte sobre serviço de buscas e solicitação de serviços, bem como conceitos básicos do desenvolvimento, *design pattern*, código limpo, tecnologia API, usabilidade de interfaces *WEB* e conceitos da engenharia de software.

2.1 ESTADO DA ARTE

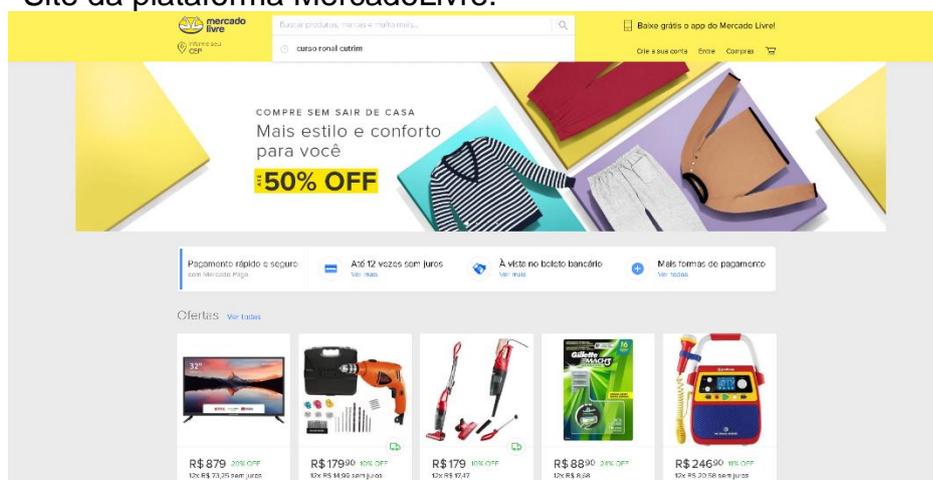
Esta seção tem como objetivo apresentar as principais soluções tecnológicas presentes no mercado¹ que visam oferecer serviços e funcionalidades semelhantes ao serviço proposto neste documento.

2.1.1 MercadoLivre

Uma empresa fundada na Argentina que oferece soluções no comércio eletrônico onde pessoas e empresas podem comprar, vender, pagar, anunciar e enviar produtos *online*.

Na figura 1 demonstra-se a página inicial da plataforma em português:

Figura 2 – Site da plataforma MercadoLivre.



Fonte: *Print* nosso da plataforma *WEB* do MercadoLivre. Plataforma disponível em: <<https://www.mercadolivre.com.br/>>. Acesso em: 18 de maio de 2020.

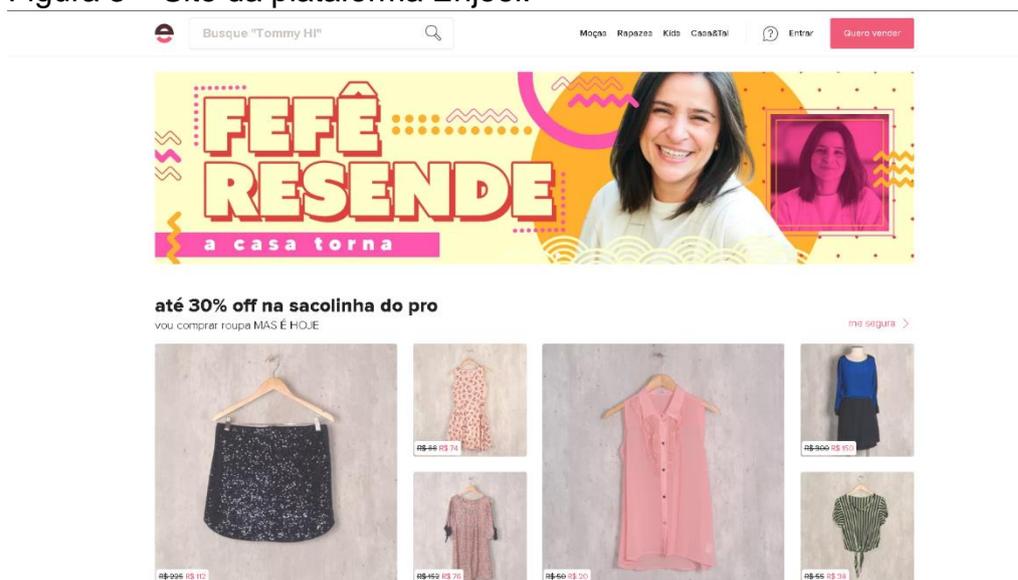
1 Na data em que esta pesquisa foi elaborada.

2.1.2 Enjoei

Empresa brasileira no ramo de comércio eletrônico, com foco em roupas. Inicialmente como um blog, transformou-se em um comércio de roupas, onde pessoas cadastradas podem anunciar e comprar roupas uns dos outros.

Na próxima figura vê-se a página principal da plataforma:

Figura 3 – Site da plataforma Enjoei.



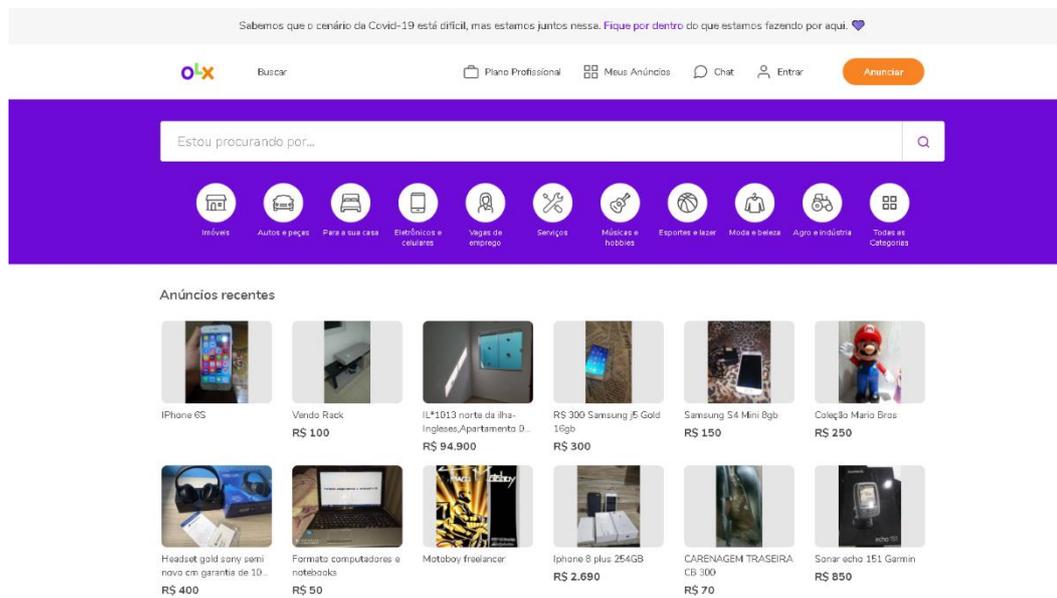
Fonte: *Print* nosso da plataforma *WEB* do Enjoei. Plataforma disponível em: <<https://www.enjoei.com.br/>>. Acesso em: 18 de maio de 2020.

2.1.3 OLX

OLX é uma empresa de nível global, com sede em Amsterdam, focada em comércio eletrônico. Pessoas cadastradas podem anunciar e comprar produtos.

Segue página principal da plataforma na próxima figura:

Figura III – Site da plataforma OLX.

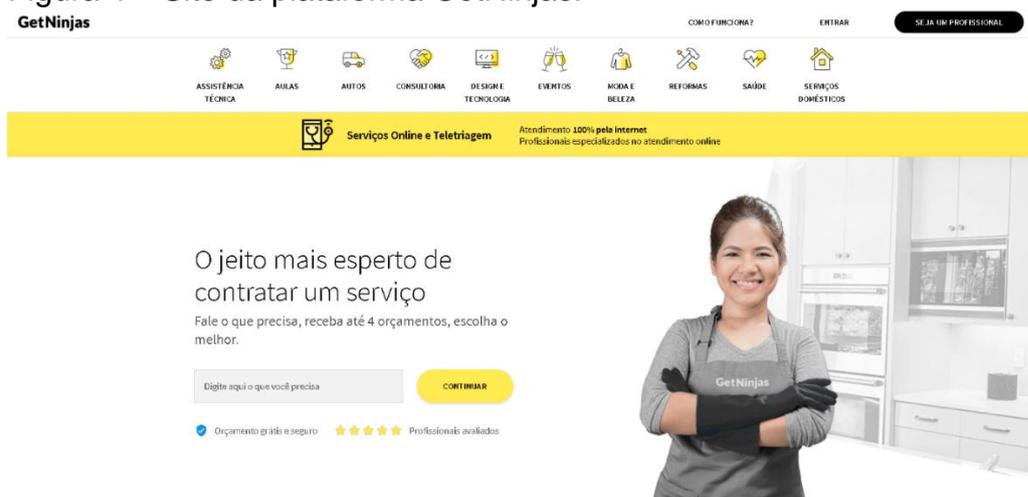


Fonte: *Print* nosso da plataforma *WEB* da OLX. Plataforma disponível em:<<https://www.olx.com.br/>>. Acesso em: 18 de maio de 2020.

2.1.4 GetNinjas

Empresa brasileira voltada ao comércio eletrônico de serviços.
Segue figura da página principal da plataforma:

Figura 4 – Site da plataforma GetNinjas.



Fonte: *Print* nosso da plataforma *WEB* do GetNinjas. Plataforma disponível em:<<https://www.getninjas.com.br/>>. Acesso em: 18 de maio de 2020.

2.1.5 Comparativo entre plataformas

Foram avaliados e comparados os sistemas semelhantes a partir de suas ofertas de produtos e/ou serviços conforme apresentados no quadro 1 a seguir.

Os autores buscaram sistemas grandes e conhecidos pela maioria da sociedade que possuem objetivo parecido ao do sistema aqui proposto e desenvolvido. Tais sistemas que também garantam ganhos de renda, tanto por oferta de serviços ou venda de produtos e um mercado amplo.

Quadro I – Comparativo entre sistemas

Produto	Oferta produtos	Oferta serviços
MercadoLivre ²	X	X
Enjoei ³	X	
OLX ⁴	X	
GetNinjas ⁵		X

Fonte: elaborado pelos autores, 2020.

O produto aqui desenvolvido pode ser tratado como um aplicativo de trabalho sob demanda, parecido como o Uber, porém com um mercado amplo, ou seja, além de apenas transporte de pessoas.

2.2 FUNDAMENTOS UTILIZADOS PARA O DESENVOLVIMENTO

Fundamentos e embasamentos teóricos ao desenvolvimento, algumas definições e conceitos tanto de programação quanto modelagem e tecnologias utilizadas para o desenvolvimento do sistema foram abordados nesse capítulo.

A programação era vista como uma tarefa sofisticada e difícil e, para facilitar, foram criadas notações formais e que continuam em evolução até os dias de hoje:

² Pode ser acessado via: <https://www.mercadolivre.com.br/>

³ Pode ser acessado via: <https://www.enjoei.com.br/>

⁴ Pode ser acessado via: <https://www.olx.com.br/>

⁵ Pode ser acessado via: <https://www.getninjas.com.br/>

As tarefas se tornaram cada vez mais complexas. Reconheceu-se lentamente que a programação era uma tarefa difícil e que dominar problemas complexos não era trivial, mesmo quando - ou porque - os computadores eram muito poderosos. A salvação foi buscada em "melhores" linguagens de programação, em mais "ferramentas", mesmo em automação. Uma linguagem melhor deve ser útil em uma área mais ampla de aplicação, mais como uma linguagem "natural", oferecer mais facilidades. (WIRTH, p. 2, 2008, tradução nossa)

Portanto, com tarefas mais complexas e um vasto número de opções disponíveis para a criação de um sistema, geralmente, prioriza-se encontrar um conjunto de tecnologias de plataforma e padrões de projetos que os desenvolvedores se familiarizam ou que estão bem alocados no momento para o serviço que irão realizar, bem como seu gerenciamento.

Enquanto o sistema ainda estiver funcionando passará por mudanças inevitáveis de requisitos, aumento da carga de trabalho, provocando problemas de desempenho e solicitações de modernização. (LATTE, HENNING e WOJCIESZAK, 2019. p. 96)

Atualmente, o desenvolvimento possui algumas regras de bom senso para facilitar o entendimento do código por outros desenvolvedores na hora de ler e entender a lógica utilizada, como técnicas de *clean code* e *design pattern*.

2.3 SOFTWARE

Software é considerado um sistema não físico, porém lógico com características diferentes do *hardware* e com um padrão de resultados que não são os previstos e destinados por uma ação intencional, tornando essencial o entendimento do *software* para obtenção dos resultados positivos. Também é considerado algo intangível. (PRESSMAN, 2016)

Os métodos propostos pela engenharia de software mostram as práticas e o modo para construir um software. Estes métodos são constituídos de vários conjuntos de tarefas que vão desde o planejamento até pós lançamento do software, como teste e manutenção.

Segundo Pressman, podemos dizer que um software se constitui em:

- Instruções que são executadas, fornecendo um resultado esperado;

- Estruturas de dados para manipular dados e informações; e,
- Informação descritiva, quais descrevem as operações e o uso dos programas.

Pode-se dizer que os principais objetivos são a produção, recuperação, evolução e manutenção dos softwares, tudo isso baseado em processos e metodologias. (REZENDE, 2005)

Contudo, o software possui uma característica diferente da maioria dos produtos, pois ele não deprecia, isto é, não sofre alterações físicas pelas condições de ambiente e tempo. Porém, ele sofre mudanças e manutenções, o que o deteriora com o tempo, aumentando a probabilidade de ocorrer erros, trazendo a necessidade de sempre manter o sistema atualizado. Isso aumenta a sua complexidade e manutenção. (PRESSMAN, 2016).

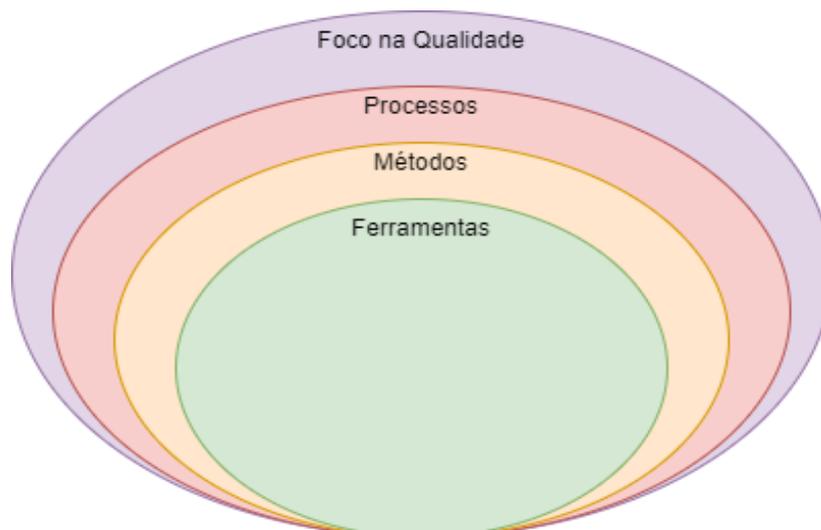
2.4 ENGENHARIA DE SOFTWARE

Engenharia de software pode ser considerada uma disciplina que tem como objetivo a preocupação completa com todos os aspectos da produção e manutenção de um software. (SOMMERVILLE, 2011).

Entretanto, pode ser também considerada como o emprego de princípios para se obter um software de maneira econômica, confiável e com funcionalidade eficiente. (PRESSMAN, 2016).

É considerada uma tecnologia em camadas, onde deve haver o foco em qualidade, pois esta camada é a base das outras. (PRESSMAN, 2016).

Figura 5 – Camadas da engenharia de software.



Fonte: Adaptada de PRESSMAN (2016, p.39).

2.4.1 Processos de Software

Com o avanço da tecnologia, o desenvolvimento de sistemas se aprimorou e se aperfeiçoou com as técnicas desenvolvidas para garantir maior segurança e qualidade, facilitando o desenvolvimento. (PRESSMAN, 2016).

Equivale-se as metodologias de desenvolvimento e manutenção do software um roteiro que contempla ferramentas, pessoas e métodos. (REZENDE, 2005).

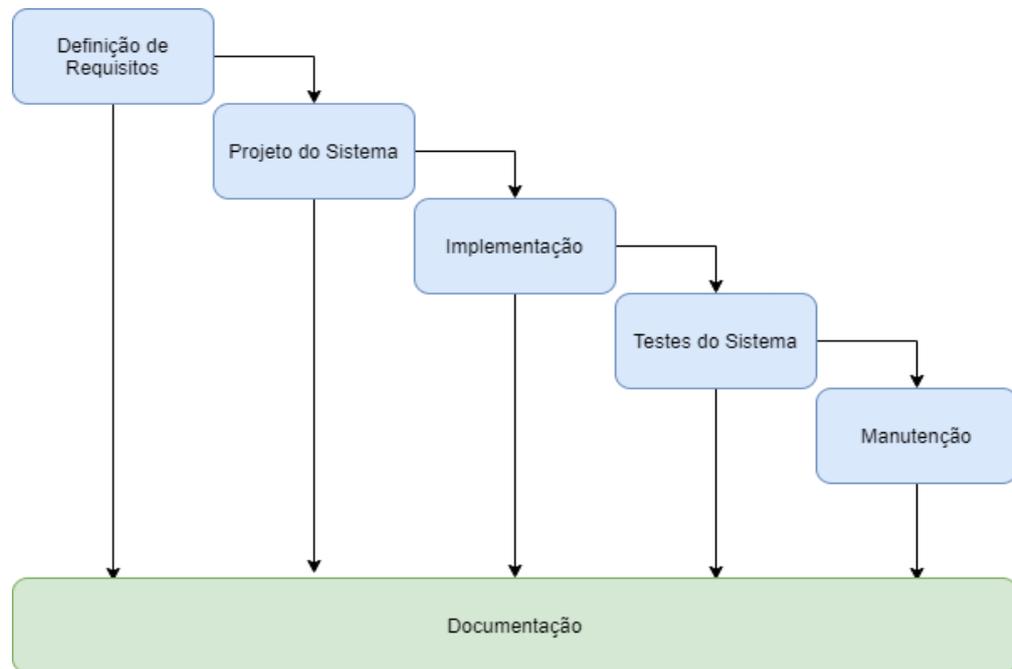
Um processo deve conter respectivas documentações formais descrevendo o que será feito (requisitos funcionais), quando será elaborado (passos), quem desenvolverá (equipe multidisciplinar ou agentes), os requerimentos de necessidades (insumos) e os efetivos resultados (produtos). (REZENDE, 2005, p. 130).

Os processos são utilizados no desenvolvimento, manutenção, aquisição e contratação de softwares, processos estes que possuem seus subprocessos. (REZENDE, 2005).

Modelos de software é a representação dos processos de software, cada qual representando um processo partindo de uma perspectiva particular que proporcionam informações parciais sobre o processo. (SOMMERVILLE, 2011).

Os modelos de processo de software possuem algumas atividades, essas que possuem ações que contemplam uma série de tarefas. A seguir segue um modelo genérico de processo de software:

Figura 6 – Processo de Software genérico.



Fonte: Elaborado pelos autores, 2020.

Nos processos existem algumas atividades que são padrões, como planejamento, modelagem e desenvolvimento, assim como atividades de apoio que perduram ao longo do processo, como administração de riscos e a garantia de qualidade. (PRESSMAN, 2016).

Existem vários modelos de processos de software cada um com suas vantagens, desvantagens e métodos. Sendo assim, cabe a cada equipe ou organização escolher o melhor modelo que se encaixa ao desenvolvimento dos sistemas.

Modelos estes como:

- Modelo Cascata;
- Modelo Espiral;
- Modelo Incremental;
- Metodologias ágeis; e,
- Scrum

2.5 DESENVOLVIMENTO DE SOFTWARE

Este capítulo traz uma abordagem das principais atividades relacionadas ao processo de desenvolvimento do sistema.

2.5.1 Análise de Requisitos de Software

Os requisitos de software devem ser elaborados no início de um projeto de sistemas de software. (REZENDE, 2005).

O engenheiro de software tem como principal tarefa filtrar todos os requisitos funcionais e não funcionais de um sistema de software, sempre prezando pela qualidade e pelo desejo do cliente. Esta etapa é uma das mais importantes do desenvolvimento, pois é ela que moldará as funcionalidades e funcionamento do sistema. Muitas vezes a falta de informações e comunicação entre cliente e o engenheiro ou o mal entendimento entre as partes podem trazer grandes malefícios ao prazo de entrega, podendo ocorrer mudanças de prazos constantes e discussões entre as partes envolvidas. (PRESSMAN, 2016).

Os requisitos são classificados como funcionais e não funcionais.

2.5.2 *CLEAN CODE* (CÓDIGO LIMPO)

O movimento de *clean code* cresceu a partir de más experiências de desenvolvedores com código legado, onde apresentaram conjuntos de práticas e princípios que facilitaram a escrita do código limpo.

Escrever código limpo não é fácil, requer conhecimento de princípios e padrões. Para conquistar a excelência em escrever código limpo é necessário praticar e falhar, errar e refazer seus passos, ver as decisões e o custo por tomar decisões erradas, porém, não escrever um código limpo, gera um código desorganizado e difícil de compreender. (MARTIN, 2009).

À medida que a bagunça aumenta, a produtividade da equipe continua a diminuir, aproximando-se de zero. À medida que a produtividade diminui, a gerência faz a única coisa que pode; eles adicionam mais funcionários ao projeto na esperança de aumentar a produtividade, mas essa nova equipe não é experiente no design do sistema. Eles não sabem a diferença entre uma alteração que corresponde à intenção do design e uma alteração que frustra a intenção do design. Além disso, eles e todos os demais membros da

equipe estão sob uma pressão terrível para aumentar a produtividade. Então, todos eles fazem mais e mais bagunças, levando a produtividade ainda mais para zero. (MARTIN, p. 4, 2009, tradução nossa).

Escrever um código limpo requer habilidades, experiência e técnicas aplicadas. Portanto, um bom programador consegue reconhecer a desorganização e ver as opções e variações para corrigi-la, traçando uma sequência de comportamento e preservando as transformações para ir do caos a um código limpo. (MARTIN, 2009).

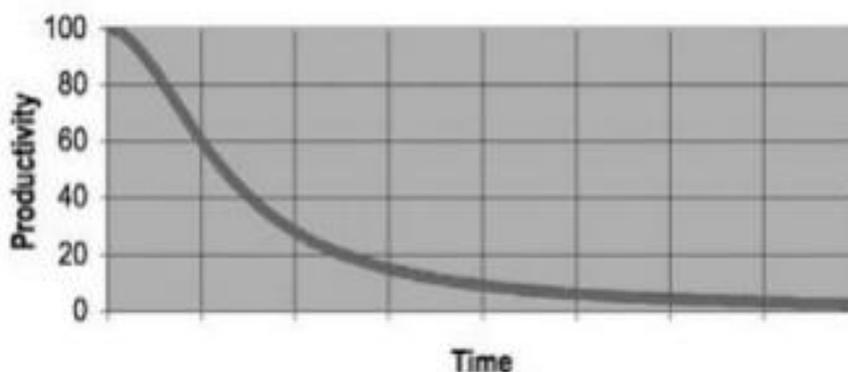
Considerando que os problemas técnicos são difíceis de lidar posteriormente, a utilização do código limpo é de extrema importância a fim de evitá-los, assim como facilita em aumentar o número de pessoas que possam trabalhar no projeto, pelo fato de o código ser compreensível não apenas ao seu autor original. (LATTE et al. 2019, p. 97).

A dificuldade do desenvolvedor em ficar horas analisando o código para compreender sua funcionalidade pode ser associado como um ambiente perigoso e desconhecido com armadilhas ocultas. Códigos produzidos em curto prazo acabam sendo entregues da forma citada anteriormente e sempre com promessas de refatoração, que raramente acontecem. (MARTIN, 2009).

Um código desorganizado e confuso gera custos adicionais na empresa, pois as alterações realizadas durante a vida do sistema se tornam cada vez mais instáveis afetando a produtividade da equipe e o custo. (MARTIN, 2009).

Segue gráfico sobre a produtividade com o passar do tempo:

Figura 7 – Gráfico de Produtividade x Tempo



Fonte: MARTIN, 2009, p.4

Em suma, o programador que escreve um código limpo é um artista que pode transformar um sistema confuso em um sistema elegantemente codificado, altamente escalável e fácil de manter e atualizar.

2.5.3 *Design Pattern* (padrão de projeto)

Design pattern pode ser descrito como a abstração de uma solução que resolve um problema, ou seja, uma composição de outros padrões atômicos ou compostos, seguindo diretrizes para alcançar uma sinergia e organização. (RIEHLE, 1997, p. 219).

O padrão atômico é um padrão isolado, logo não pode ser descrito como a composição de outros padrões. Por sua vez, o padrão composto é a soma de alguns padrões, podendo ser uma solução para problemas de design específicos, mas que não é tratado como um padrão próprio. (RIEHLE, 1997, p. 219).

Padrões de projeto independe da habilidade de programar, tendo como uma necessidade o uso de linguagens onde a orientação a objetos seja viável e o uso do *design pattern* demanda mais esforço, mas o esforço extra gera maior flexibilidade e reutilização de código. (GAMMA et al, 1995).

A razão pela qual apenas a orientação a objetos se torna viável é pelo simples fato das classes, sendo elas representações de objetos. Tendo suas modelagens em forma de diagramas para alcançar uma solução para o problema que o sistema irá solucionar. (RIEHLE, 1997, p. 220).

Entretanto, os diagramas de classes podem ser considerados uma solução do problema de distribuição de responsabilidades entre os objetos de forma eficiente e fácil de implementar. (RIEHLE, 1997, p. 220).

Os diagramas de função são considerados melhores para padrões baseados em colaboração de objetos do que os diagramas de classe, pois concentram-se melhor na solução real do problema como um conjunto de objetos colaborativos. (RIEHLE, 1997, p. 220).

Os padrões facilitam a reutilização de projetos e arquiteturas. Expressar técnicas comprovadas como padrões de design tornam mais acessíveis aos desenvolvedores de novos sistemas. Os padrões de design ajudam a escolher alternativas de design que tornam um sistema reutilizável e evitam alternativas que comprometem a reutilização. Os padrões de design podem até melhorar a documentação e a manutenção dos sistemas existentes, fornecendo uma especificação explícita das interações de classe e objeto e

sua intenção subjacente. Simplificando, os padrões de design ajudam um designer a obter um design "correto" mais rapidamente. (GAMMA et al, p. 2, 1995).

Os padrões também tornam um aplicativo mais sustentável e aprimoram a escalabilidade, mostrando como estender hierarquias de classes e como explorar a composição dos objetos. O acoplamento reduzido também aprimora a escalabilidade. Estender uma classe isoladamente é mais fácil se a classe não depender de muitas outras classes. (GAMMA et al, 1995).

2.5.4 Plataforma RESTful

No ano de 2000, após a crise de escalabilidade da web ter sido evitada, Fielding nomeou e descreveu o estilo arquitetônico da web como "*Representational State Transfer*" (REST) composto pelas restrições da *web*. (MASSÉ, 2012, p. 5).

Webservices são propositalmente construídos para suportar a necessidade de *sites* e aplicações, utilizando aplicações de interface programada (APIs – *Application Programming Interfaces*) que têm como objetivo a exposição de funções e dados através de interfaces *webservice* facilitando a troca de dados entre aplicações computacionais e as interfaces. A união da arquitetura REST com API faz com que o *webservice* se torne "RESTful". (MASSÉ, 2012, p. 5-6).

Desenvolvimento em REST é mais rápido e fácil, sendo a interface de API mais utilizada no para criação de *webservices* desenvolvidos através do uso de padrões populares e bem estabelecidos. (RICHARDSON; RUBY, 2007).

Portanto, possuindo um modelo simplificado da padronização de como a internet é utilizada, sendo estas características que fazem um *site* acessível podem ser as características que tornam a *webservice* fácil de ser utilizada. (RICHARDSON; RUBY, 2007).

Neste caso, a linguagem de programação é extremamente importante, unindo a facilidade de realizar o *design pattern* com a facilidade de integração *web*.

2.6 INTERFACE

Com os avanços da forma de se trabalhar com o *frontend*, layout e a interface dos dias atuais, a linguagem de programação utilizada é o Vue.js, por sua

popularidade entre os desenvolvedores, como aponta a pesquisa do GeekHunter. (FRIAS, 2020).

2.6.1 Usabilidade

Atualmente, com o advento da internet e páginas *web*, há um processo de avaliação de interfaces que seja acessível a todos os tipos de pessoas e de fácil entendimento e utilização.

Este é o campo científico chamado de Interface Humano-Computador (IHC).

2.6.2 Interface Humano-Computador (IHC)

Interface Humano-Computador é uma grande área interdisciplinar que engloba ciência da computação, psicologia, sociologia, antropologia e desenho industrial. (ACM SIGCHI, 1992 apud MATSUKUMA, 2012). Algumas preocupações são:

- Desempenho entre humano e computador;
- Comunicação entre humano e computador;
- Capacidade humana de utilizar o computador;
- Programação da interface;
- Problemas de engenharia; e,
- Especificação, projeção e implementação da interface.

Portanto, é impossível pensar em interface sem pensar nas pessoas que a estão utilizando.

2.7 CODIFICAÇÃO

Primeiramente, antes do desenvolvimento propriamente dito, é necessário organizar o ambiente para o desenvolvimento, como configurar as ferramentas, as tecnologias e bibliotecas que serão utilizadas no sistema, para assim começar a codificar. (REZENDE, 2005).

Seguindo a modelagem sugerida nos capítulos anteriores serão escritas inúmeras linhas de código que aos poucos ganhará forma para atender as funcionalidades necessárias para o funcionamento correto do sistema. (REZENDE, 2005).

Aliado ao desenvolvimento serão feitos testes junto ao código do sistema com finalidade de manter o sistema com falhas mínimas. Finalizando algumas atividades ou alguma funcionalidade será gerado versões e enviadas ao servidor com uma ferramenta de versionamento. (BEZERRA, 2007).

2.8 TESTES

As atividades de teste e análises ocorrem ao longo do desenvolvimento e da evolução do software. A qualidade do sistema depende de cada parte do processo, desde os requisitos até a parte pós lançamento. Assim, o processo de testes e análises garantem e geram um produto de alta qualidade. (PEZZÈ e YOUNG, 2008).

Um software com testes bem montados e gerenciados tem garantia de que o produto possui qualidade e apresenta menos erros durante a execução.

2.9 CASOS DE USOS

Os casos de usos é a principal forma de representar os requisitos em *Unified Modeling Language* (UML) e é a principal escolha para a especificação e análise de requisitos. (LEFFINGWELL, 2011)

Casos de uso são fundamentais para os entendimentos das interações que o sistema fará com os usuários e suas respectivas funcionalidades. (FOWLER, 2005)

Não existe padrão para descrever um caso de uso. Porém, há algumas estruturas nos casos de uso que são mandatórias:

- Nome: descreve o objetivo do Caso de Uso, de forma objetiva;
- Descrição: propósito do caso de uso, de forma breve;
- Ator(es): lista do(s) autor(es);
- Fluxo de eventos: interações entre os sistemas e os autores. Pode ser dividido em duas:

- Básico: caminho principal do caso de uso.
- Alternativo: eventos que ocorrem em uma circunstância alternativa.

E outros elementos opcionais, mas deve-se verificar a necessidade e acordo com o desenvolvimento e produto produzidos.

3 MÉTODO

Método é o estudo sistemático, a organização, a pesquisa e investigação. Pode-se afirmar que metodologia é o estudo das maneiras e ferramentas necessárias para elaborar uma pesquisa científica. (FONSECA, 2002).

Sendo uma pesquisa científica, é importante que se realize as devidas apresentações dos métodos que são utilizados na realização deste trabalho.

3.1 TIPO DE PESQUISA

Conceituando o significado de pesquisa científica, pode-se afirmar que se trata de uma pesquisa aplicada que levanta questionamentos para extrair dados de fim de alcançar uma solução da problemática.

A pesquisa aqui descrita propôs o desenvolvimento de um sistema para atender a problemática, conforme informado acerca do objetivo do presente trabalho.

Atividade básica das ciências na sua indagação e descoberta da realidade. É uma atitude e uma prática teórica de constante busca que define um processo intrinsecamente inacabado e permanente. É uma atividade de aproximação sucessiva da realidade que nunca se esgota, fazendo uma combinação particular entre teoria e dados. (Minayo, 1993, p.23).

Para processos técnicos descritos, a pesquisa mais compatível é a qualitativa, esta que busca analisar e estudar livros e bibliográficas com a finalidade de solucionar a problemática.

É considerada que há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números. A interpretação dos fenômenos e a atribuição de significados são básicas no processo de pesquisa qualitativa. Não requer o uso de métodos e técnicas estatísticas. O ambiente natural é a fonte direta para coleta de dados e o pesquisador é o instrumento-chave. (Silva e Menezes, 2005, p. 20).

3.2 ETAPAS METODOLÓGICAS

Este trabalho consiste na seguinte estrutura:

- Fundamentação teórica;
- Análise do sistema;

- Desenvolvimento;
- Testes; e,
- Documentação.

Em respeito às fases, a fundamentação teórica tem o foco em buscar todo um embasamento teórico-científico e referência bibliográfica, melhorando o desenvolvimento do sistema.

Na análise são desenhados todos os diagramas necessários para não ocorrerem erros nem problemas inesperados e para documentação do desenvolvimento do sistema.

Após uma análise criteriosa para minimizar, ou até mesmo zerar, acontecimentos imprevisíveis, gerando uma base estrutural reforçada para a codificação, o desenvolvimento do sistema é iniciado.

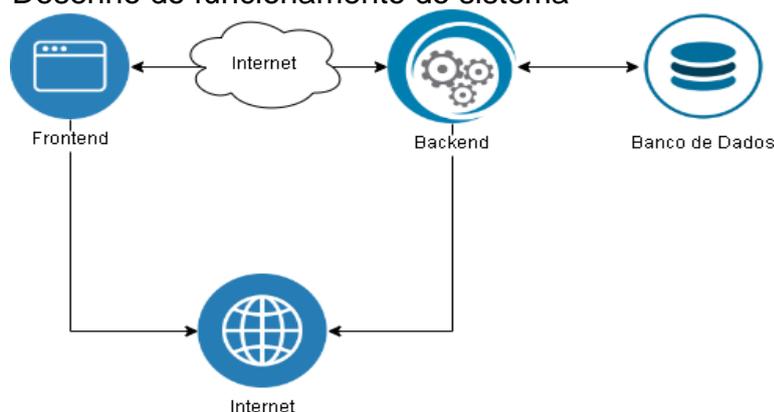
A todo tempo que demandas são geradas e desenvolvidas, até mesmo no fim do projeto, testes unitários e testes em geral são configurados e executados a para manter o sistema funcionando sem problemas não previsíveis.

Após o desenvolvimento e os testes finalizados, uma documentação sobre o funcionamento básico do sistema será elaborado. Testes e documentação se encaixam no processo do desenvolvimento do sistema para facilitar sua manutenção, custo operacional e o uso.

3.3 SOLUÇÃO PROPOSTA

Para o desenvolvimento do sistema, o desenho básico do sistema é o representado na Figura 8:

Figura 8 – Desenho do funcionamento do sistema



Fonte: Elaborado pelos autores, 2020.

O sistema terá 2 tipos de usuários, podendo um usuário ser de um tipo ou ambos ao mesmo tempo:

- Cliente: Pessoas ou empresas que são cadastradas no sistema apenas para busca de serviços e produtos.
- Fornecedor: Pessoas ou empresas cadastradas que oferecem serviços e produtos no sistema.

O banco de dados é responsável por salvar informações de usuários, produtos e serviços.

A aplicação será disponibilizada apenas em *WEB*, porém a engenharia utilizada facilita a integração com sistemas *mobile* como Android e iOS. O sistema disponibilizará produtos e serviços onde os usuários podem comprar ou contratar, podendo optar pela forma de contrato semanal, mensal ou apenas uma vez.

3.4 DELIMITAÇÕES

O sistema tem as seguintes delimitações:

- A aplicação será apenas disponibilizada em modelo *WEB*;
- Será necessário possuir cadastro e logado para acessar as funcionalidades do sistema;

- Por ser um sistema *WEB*, será apenas possível acessar o sistema online.

4 MODELAGEM DO SISTEMA

O sistema segue o formato de *webservice* para ter facilidade de expansão e elasticidade do sistema quanto a plataformas e sistemas diversos. Moldado com os melhores métodos de desenvolvimento e escalabilidade, facilitando seu suporte e melhorando sua performance.

O sistema tem o diferencial de você conseguir contratar serviços mensais, semanais, anuais e trimestrais.

Neste capítulo é abordado a modelagem do sistema que é produzida em UML. O sistema é representado através de modelagens do banco de dados, fluxo das funções do sistema, entre outros.

4.1 ANÁLISE DO ESTADO DA ARTE

Com sistemas parecidos apresentados, pode-se criar um quadro comparativo com a ideia proposta neste trabalho acadêmico.

No próximo quadro foram avaliados e comparados os sistemas semelhantes, a partir de suas ofertas de produtos e/ou serviços com a proposta de solução deste trabalho. Nessa comparação destaca-se que o diferencial é oferecer serviços recorrentes com pagamentos recorrentes.

O sistema aqui desenvolvido tem como diferencial trazer pessoas com conhecimentos e que possam oferecer em forma de serviço recorrente, assim quem contrata tem um serviço recorrente e quem oferece tem uma renda recorrente.

Quadro II – Comparativo completo entre sistemas

Produto	Oferta produtos	Oferta serviços	Serviços em “n” períodos	Classificação Produto/serviço
MercadoLivre ⁶	X	X		X
Enjoei ⁷	X			X
OLX ⁸	X			X
GetNinjas ⁹		X		X

⁶ Pode ser acessado via: <https://www.mercadolivre.com.br/>

⁷ Pode ser acessado via: <https://www.enjoei.com.br/>

⁸ Pode ser acessado via: <https://www.olx.com.br/>

⁹ Pode ser acessado via: <https://www.getninjas.com.br/>

Sellvice ¹⁰		X	X	
------------------------	--	---	---	--

Fonte: Elaborado pelos autores, 2020.

O serviço em “n” períodos é uma relação entre cliente e provedor, onde o provedor concede um serviço recorrente em troca de um pagamento recorrente.

4.2 ATORES

Os atores são os usuários do sistema. No sistema proposto, o ator-usuário representa as pessoas cadastradas que utilizam o sistema.

Figura 9 – Ator do sistema



Fonte: Elaborado pelos autores, 2020.

As definições do usuário estão detalhadas no seguinte quadro:

Quadro III – Definições do ator usuário

Ator	Usuário
Definição	Usuário com interesse em ofertar serviços ou consumir serviços de terceiros.
Frequência de uso	Diário.
Conhecimento em informática	Intermediário. Deve possuir qualquer dispositivo conectado à internet e um navegador.
Grau de escolaridade	Alfabetizado.
Permissão de acesso	Acesso comum.

Fonte: Elaborado pelos autores, 2020.

Com a definição do autor, pode-se começar a verificação dos requisitos necessários ao desenvolvimento do sistema.

¹⁰ Plataforma desenvolvida pela presente monografia.

4.3 REQUISITOS

Os requisitos são essenciais à análise de software, podendo assim documentar de forma estruturada todas as necessidades mapeadas em fase de desenvolvimento do sistema, analisando o desenvolvimento para permitir que ocorra de forma correta e mensurar a complexidade das necessidades do projeto.

Entretanto, assume-se que hoje em dia os requisitos são mais que funções do sistema, podem também ser propriedades, objetivos, restrições e padrões, especificamente alinhadas junto com o usuário final, assim, para atingir sua satisfação. (MEDEIROS, 2013).

4.3.1 Requisitos funcionais

É o que permite que o cliente, ou usuário, veja a funcionalidade em questão, garantindo que o desenvolvedor o entregue.

Requisitos funcionais são descritos como uma interação entre o ambiente e o sistema, a maneira de como o sistema deve se comportar dependendo da ação exercida pelo usuário ou pelo próprio sistema. (PFLEEGER, 2004).

Pode-se definir também como requisitos aqueles que definem o comportamento do sistema qual são abordados nos casos de uso. (CORDEIRO, 2010).

Quadro IV – Quadro de requisitos funcionais

REQUISITOS FUNCIONAIS		
Identificação	Nome	Descrição
RF01	Mostrar serviços	Mostrar serviços existentes e cadastrados no sistema
RF02	Cadastrar usuário	Deve permitir o cadastro do usuário
RF03	Cadastrar serviço	Deve permitir o usuário cadastrado como “lojista” cadastrar serviços
RF04	Cadastrar “loja”	Deve permitir um usuário cadastrado virar um “lojista” para cadastrar serviços

RF05	Conversar por chat	Deve permitir que usuário cliente do serviço possa conversar com o prestador de serviço via <i>whatsapp</i>
RF06	Login	Tendo um cadastro, o processo de login deve permitir o usuário a verificar, buscar e contratar serviços e se tornar um “lojista”

Fonte: Elaborado pelos autores, 2020.

4.3.2 Requisitos não funcionais

Requisitos não funcionais podem ser definidos como acessibilidade, performance e meios de exibição, isto é, pontos importantes para um sistema. (FAGUNDES, 2011).

No quadro V apresenta-se os requisitos não funcionais do aplicativo a ser desenvolvido:

Quadro V – Requisitos não funcionais

REQUISITOS NÃO FUNCIONAIS	
Identificação	Descrição
RNF01	Ter uma aplicação <i>web</i> responsiva.

Fonte: Elaborado pelos autores, 2020.

Na Quadro VI é apresentado os requisitos não funcionais referentes ao desempenho:

Quadro VI – Requisitos não funcionais (desempenho)

REQUISITOS NÃO FUNCIONAIS	
Identificação	Descrição
RNF02	Todas as requisições devem trazer os resultados em até 10 segundos.

Fonte: Elaborado pelos autores, 2020.

No quadro VII apresenta-se os requisitos não funcionais referentes ao desenvolvimento do sistema:

Quadro VII – Requisitos não funcionais (desenvolvimento)

REQUISITOS NÃO FUNCIONAIS	
Identificação	Descrição
RNF03	Aplicação desenvolvida com <i>Firebase Realtime Database</i> .
RNF04	Utilização do <i>framework</i> Vue.JS para desenvolvimento do <i>front-end</i> .

Fonte: Elaborado pelos autores, 2020.

No quadro VIII são apresentados os requisitos não funcionais referentes aos sistemas operacionais que devem suportar o aplicativo:

Quadro VIII – Requisitos não funcionais (sistemas operacionais)

REQUISITOS NÃO FUNCIONAIS	
Identificação	Descrição
RNF05	A plataforma deve suportar todos os sistemas operacionais através de navegadores de internet.
RNF06	Funcionar apenas em dispositivos conectados à internet.

Fonte: Elaborado pelos autores, 2020.

Quanto aos requisitos mostrados anteriormente, são mostrados para que no desenvolvimento do sistema exista um padrão entre os desenvolvedores.

No quadro IX são apresentados os requisitos não funcionais referentes a operabilidade do sistema:

Quadro IX - Requisitos não funcionais (operabilidade)

REQUISITOS NÃO FUNCIONAIS	
Identificação	Descrição
RNF07	O sistema deve possuir interfaces simplificadas, com claras aplicações de usabilidade.
RNF08	Informações no sistema devem ser de fácil acesso a quem tem permissão.
RNF09	Cadastros no sistema deve ser simples, fácil e rápido, em menos de 5 minutos.

RNF10	O sistema deve ter uma operação de <i>timeout</i> habilitado e ativo a todo tempo que o usuário estiver utilizando.
-------	---

Fonte: Elaborado pelos autores, 2020.

No quadro X é apresentado os requisitos não funcionais referentes a confiabilidade:

Quadro X – Requisitos não funcionais (confiabilidade)

REQUISITOS NÃO FUNCIONAIS	
Identificação	Descrição
RNF11	O sistema deve funcionar 24/7, em caso de alguma falha, o serviço deve retornar de maneira mais rápida.
RNF12	O sistema deve garantir que somente usuários com permissões tenham acesso.
RNF13	O sistema deve manter a integridade dos dados e informações gravadas em banco de dados.

Fonte: Elaborado pelos autores, 2020.

4.4 REGRAS DE NEGÓCIO

Regras de negócio são o que realmente determinam as principais funcionalidades de um sistema, mas descritas de forma atômicas para que fiquem legíveis e, assim, diminuindo a complexidade do entendimento. (FAGUNDES, 2011).

Também define como o negócio deve ser conduzido e suas restrições, pois o uso do sistema deve incorporar devidas regras e políticas. (SOMMERVILLE, 2007).

Estas são definidas no quadro a seguir:

Quadro XI – Regras de negócio.

REGRAS DE NEGÓCIO		
Identificação	Nome	Descrição

RN01	Informações sobre o serviço	Para cada serviço oferecido no sistema, a aplicação deve mostrar: <ul style="list-style-type: none"> • Nome do serviço; • Prestador do serviço; • Valor do serviço; • <i>Rating</i> do serviço;
RN02	Chat	O chat (<i>whatsapp</i>) para o usuário conversar com o prestador do serviço.
RN03	Campos obrigatórios	Para cadastrar no sistema, o usuário deve possuir um e-mail e um CPF/CNPJ. Cadastros únicos por CPF/CNPJ.

Fonte: Elaborado pelos autores, 2020.

Com o quadro montado, leva-se a conclusão das condições e restrições impostas para o desenvolvimento do sistema, quais devem ser respeitadas em todo o processo de desenvolvimento.

4.5 CASOS DE USO

Para iniciar o projeto, baseado no requisito funcional (RF03), é documentado abaixo o processo de cadastro de usuário, conforme o quadro XII.

Quadro XII – Caso de Uso (cadastro)

Caso de Uso (cadastro)	
Identificação	C1
Título	Realizar Cadastro
Descrição do caso de uso	Procedimento para o cadastro do usuário na plataforma. É necessário para que o usuário tenha acesso as ações do sistema, quanto oferecer ou solicitar serviço.
Atores	Usuário
Requisitos correlacionados	

Fluxo básico	
Pré-condições	1) Não possuir cadastro em mesmo CPF/CNPJ; 2) Possuir acesso à internet.
Pós-condições	1) Tela de busca e apresentação dos serviços disponíveis será apresentada.
Fluxo principal	
1) Usuário preenche campos obrigatórios para realizar cadastro; 2) Usuário clica em “Cadastrar” para finalizar cadastro; 3) Sistema persistirá os dados em banco retornando aviso de cadastro bem-sucedido.	
Tratamentos de exceção	
1) Dados obrigatórios não preenchidos; <ul style="list-style-type: none"> a. O sistema alerta falha e mostra campos com problema; b. Retorna ao fluxo principal 1. 2) CPF/CNPJ ou e-mail já cadastrado. <ul style="list-style-type: none"> a. O sistema alerta falha e mostra campos com problema; b. Retorna ao fluxo principal 1. 	

Fonte: Elaborado pelos autores, 2020.

O caso do quadro XIII demonstra o processo de autenticação, tendo registrado todas suas possibilidades e fluxos:

Quadro XIII – Caso de Uso (autenticação)

Caso de Uso (autenticação)	
Identificação	C2
Título	Autenticação
Descrição do caso de uso	Após cadastro, o usuário poderá realizar a autenticação em sua conta, liberando as ações de oferecer e contratar serviços.
Atores	Usuário
Requisitos correlacionados	

Fluxo básico	
Pré-condições	1) Não estar autenticado no sistema; 2) Possuir acesso à internet.
Pós-condições	1) Tela de busca e apresentação dos serviços disponíveis será apresentada.
Fluxo principal	
4) Usuário preenche campos obrigatórios para realizar a autenticação; 5) Usuário clica em “Logar” para autenticar-se no sistema; 6) Sistema valida os dados, retornando sucesso da autenticação.	
Tratamentos de exceção	
1) Dados obrigatórios não preenchidos; a. O sistema alerta falha e mostra campos com problema; b. Retorna ao fluxo principal 1.	

Fonte: Elaborado pelos autores, 2020.

No quadro XIV, o caso de uso apresentado é o de visualização de serviços disponíveis, sendo essa a tela principal do sistema:

Quadro XIV – Caso de Uso (visualizar serviços)

Caso de Uso (visualizar serviços)	
Identificação	C3
Título	Visualizar serviços disponíveis.
Descrição do caso de uso	Visualizar serviços disponíveis feita através da interface principal, onde apresentará uma lista de serviços disponíveis, ordenado por classificação.
Atores	Usuário
Requisitos correlacionados	
Fluxo básico	
Pré-condições	1) Estar autenticado no sistema; 2) Possuir acesso à internet.
Pós-condições	
Fluxo principal	

- 1) Usuário recebe a lista de serviços disponíveis mais bem ranqueados;
- 2) Usuário informa filtros para os serviços;
- 3) Usuário pressiona “Buscar”;
- 4) Sistema filtra dados em banco, retornando os serviços baseado nos filtros usados pelo usuário;
- 5) Sistema atualiza a lista com os serviços retornados.

Tratamentos de exceção

- 1) Dado de filtro inválido;
 - a. O sistema alerta falha e mostra campos com problema;
 - b. Retorna ao fluxo principal 1.

Fonte: Elaborado pelos autores, 2020.

O quadro a seguir demonstra o caso de uso dos detalhes dos serviços disponíveis, caso o usuário selecione um serviço da lista:

Quadro XV – Caso de Uso (detalhes do serviço)

Caso de Uso (detalhes do serviço)	
Identificação	C4
Título	Visualizar detalhes do serviço
Descrição do caso de uso	Ao clicar em um serviço na lista de apresentação de serviços disponíveis, exibir detalhes em outra tela.
Atores	Usuário
Requisitos correlacionados	C3
Fluxo básico	
Pré-condições	<ol style="list-style-type: none"> 1) Estar autenticado no sistema; 2) Possuir acesso à internet.
Pós-condições	<ol style="list-style-type: none"> 1) Contatar provedor do serviço.
Fluxo principal	
<ol style="list-style-type: none"> 1) Usuário seleciona o serviço para visualizar detalhes; 2) Sistema retorna dados do serviço selecionado para exibição em tela. 	

Fonte: Elaborado pelos autores, 2020.

O quadro XVI demonstra o processo de como contatar o provedor do serviço através da tela de detalhes:

Quadro XVI – Caso de Uso (contatar o provedor do serviço)

Caso de Uso (contatar o provedor do serviço)	
Identificação	C5
Título	Contatar o provedor do serviço.
Descrição do caso de uso	O usuário pode entrar em contato com o provedor de serviço na tela de detalhes do serviço.
Atores	Usuário
Requisitos correlacionados	C2, C4
Fluxo básico	
Pré-condições	1) Estar autenticado no sistema; 2) Possuir acesso à internet.
Pós-condições	1) Tela atualizará, encaminhando o usuário para o <i>whatsapp</i> para conversar com o provedor do serviço.
Fluxo principal	
1) Usuário pressiona o botão de chat com o provedor do serviço; 2) Sistema encaminha o usuário para o <i>whatsapp</i> para conversar com o provedor do serviço; 3) Usuário digita o texto e pressiona em “Enviar”; 4) Sistema salva o texto em banco e notifica o provedor do serviço.	

Fonte: Elaborado pelos autores, 2020.

O caso do quadro XVI é a contratação de serviço:

Quadro XVI – Caso de Uso (contratar o serviço)

Caso de Uso (contratar o serviço)	
Identificação	C6
Título	Contratar o serviço.

Descrição do caso de uso	O usuário pode solicitar o serviço, onde o provedor será notificado e poderá cancelar ou aceitar.
Atores	Usuário
Requisitos correlacionados	C2, C4
Fluxo básico	
Pré-condições	1) Estar autenticado no sistema; 2) Possuir acesso à internet; 3) Ter selecionado um serviço.
Pós-condições	1) Será adicionado ao histórico do usuário a solicitação do serviço.
Fluxo principal	
1) Usuário seleciona o serviço, para ver mais detalhes; 2) Usuário pressiona o botão de solicitar serviço; 3) Sistema apresenta um <i>modal</i> ; 4) Usuário coloca uma descrição do que necessita, caso necessário; 5) Usuário clica em “Enviar Proposta”; 6) Sistema processa o pedido em banco e notifica o provedor do serviço; 7) Provedor de acesso pode aceitar ou recusar um serviço.	
Tratamentos de exceção	
1) Solicitação em aberto. <ul style="list-style-type: none"> a. O sistema alerta que já existe uma solicitação desse serviço em andamento; b. Retorna ao fluxo 1. 	

Fonte: Elaborado pelos autores, 2020.

No caso do quadro XVIII é apresentada a visualização da tela de histórico de pedidos:

Quadro XVIII – Caso de Uso (visualizar histórico de pedidos)

Caso de Uso (visualizar histórico de pedidos)	
Identificação	C8
Título	Visualizar o histórico de pedidos.

Descrição do caso de uso	O usuário poderá ver seu histórico de pedidos de solicitações de serviço.
Atores	Usuário
Requisitos correlacionados	C2
Fluxo básico	
Pré-condições	<ol style="list-style-type: none"> 1) Estar autenticado no sistema; 2) Possuir acesso à internet. 3) Possuir ao menos uma solicitação de serviço.
Pós-condições	
Fluxo principal	
<ol style="list-style-type: none"> 1) Usuário acessa página de solicitações através de qualquer página, pelo menu e seleciona o menu “Solicitações”; 2) O usuário seleciona a aba “Realizado por mim”; 3) O sistema abrirá todos os pedidos do usuário autenticado; 4) Usuário informa filtros para os pedidos; 5) Usuário pressiona “Buscar”; 6) Sistema filtra dados em banco, retornando os pedidos baseado nos filtros usados pelo usuário; 7) Sistema atualiza a lista com os pedidos retornados. 	
Tratamentos de exceção	
<ol style="list-style-type: none"> 1) Dado de filtro inválido; <ol style="list-style-type: none"> a. O sistema alertará falha e mostrará campos com problema; b. Retorna ao fluxo principal 1. 	

Fonte: Elaborado pelos autores, 2020.

O quadro a seguir apresenta o caso de uso sobre a visualização dos pedidos em abertos para o usuário aceitar e prover o serviço a outro usuário:

Quadro XIX – Caso de Uso (visualizar pedidos em abertos)

Caso de Uso (visualizar pedidos em abertos)	
Identificação	C9

Título	Visualizar os pedidos em aberto.
Descrição do caso de uso	O usuário poderá ver seus pedidos em abertos para serviços que ele presta.
Atores	Usuário
Requisitos correlacionados	C2
Fluxo básico	
Pré-condições	<ol style="list-style-type: none"> 1) Estar autenticado no sistema; 2) Possuir acesso à internet; 3) Possuir um serviço disponível; 4) Possuir ao menos uma solicitação de serviço em aberto.
Pós-condições	<ol style="list-style-type: none"> 1) Sistema atualizará a lista de pedidos em aberto.
Fluxo principal	
<ol style="list-style-type: none"> 1) Usuário acessa página de solicitações através de qualquer página, pelo menu e seleciona o menu “Solicitações”; 2) Usuário seleciona a aba “Pedidos a mim” 3) O sistema abrirá todos os pedidos do usuário autenticado; 4) Usuário informa filtros para os pedidos; 5) Usuário pressiona “Buscar”; 6) Sistema filtra dados em banco, retornando os pedidos baseado nos filtros usados pelo usuário; 7) Sistema atualiza a lista com os pedidos retornados. 	
Tratamentos de exceção	
<ol style="list-style-type: none"> 2) Dado de filtro inválido; <ol style="list-style-type: none"> a. O sistema alerta falha e mostra campos com problema; b. Retorna ao fluxo principal 1. 	

Fonte: Elaborado pelos autores, 2020.

No próximo quadro é apresentado o caso de responder uma solicitação de serviço:

Quadro XX – Caso de Uso (responder solicitação do serviço)

Caso de Uso (responder solicitação do serviço)	
Identificação	C10
Título	Responder solicitação do serviço.
Descrição do caso de uso	O usuário pode aceitar ou recusar a solicitação de prestação de serviço.
Atores	Usuário
Requisitos correlacionados	C2, C6, C9
Fluxo básico	
Pré-condições	<ol style="list-style-type: none"> 1) Estar autenticado no sistema; 2) Possuir acesso à internet. 3) Possuir ao menos uma solicitação de serviço em aberto.
Pós-condições	1) Tela atualizará, mostrando as solicitações atualizadas.
Fluxo principal	
<ol style="list-style-type: none"> 1) Usuário acessa página de “Pedidos a mim” e seleciona a solicitação; 2) O sistema abre o perfil do solicitante; 3) Usuário pode aceitar ou recusar a solicitação e enviar um orçamento ao solicitante. 4) Uma notificação é enviada ao usuário solicitante. 	

Fonte: Elaborado pelos autores, 2020.

Para então finalizar, o quadro abaixo apresenta o caso de aprovação mútua:

Quadro XXI – Caso de Uso (aprovação mútua)

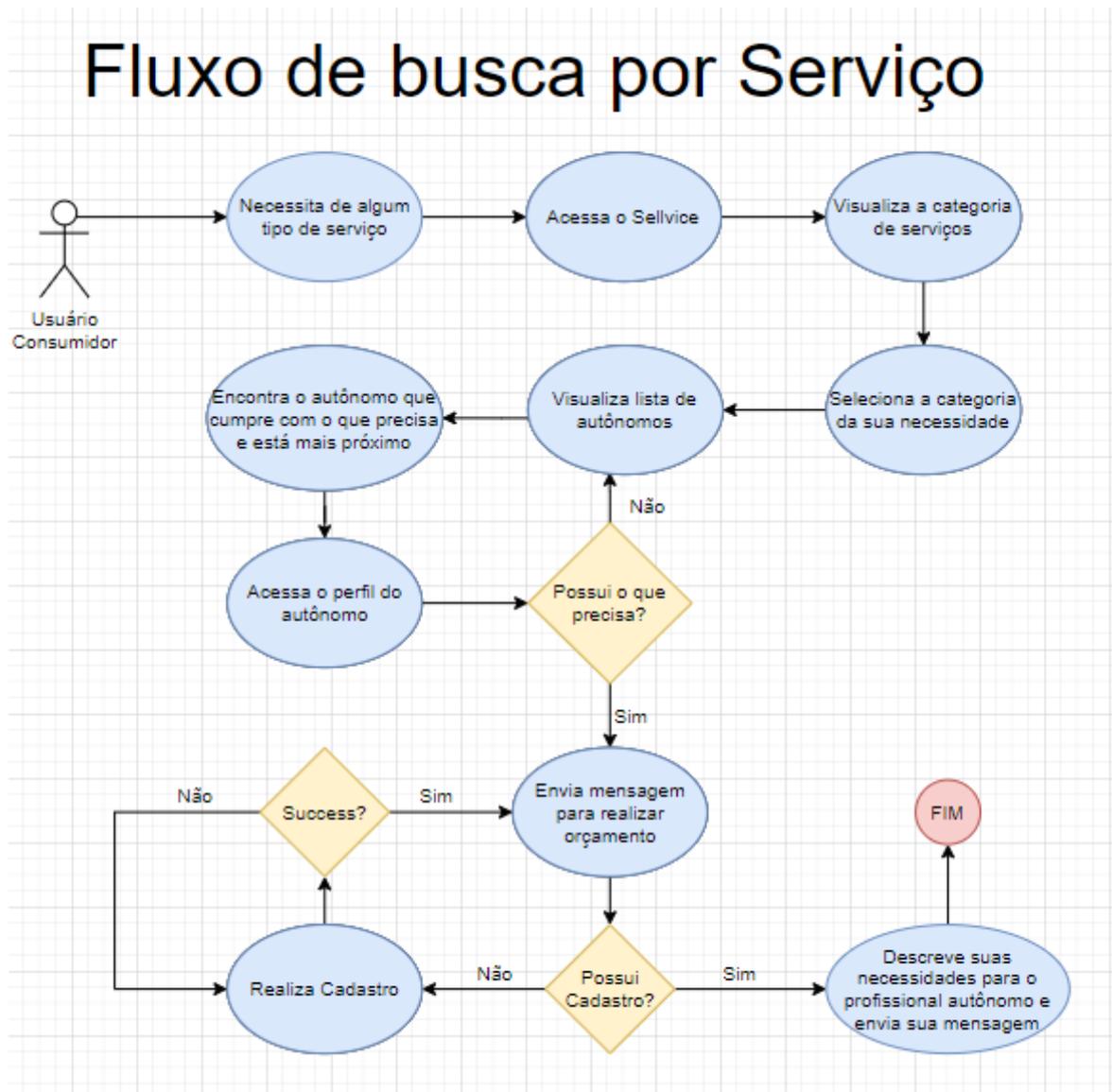
Caso de Uso (aprovação mútua)	
Identificação	C11
Título	Usuário solicitante recebe a proposta do provedor.

Descrição do caso de uso	O usuário solicitante recebe a proposta do provedor com um orçamento para você aprovar ou recusar.
Atores	Usuário
Requisitos correlacionados	C2, C6, C8, C10
Fluxo básico	
Pré-condições	<ol style="list-style-type: none"> 1) Estar autenticado no sistema; 2) Possuir acesso à internet. 3) Possuir ao menos uma solicitação de serviço em aberto.
Pós-condições	1) Tela atualiza mostrando as solicitações atualizadas.
Fluxo principal	
<ol style="list-style-type: none"> 1) Usuário acessa página de solicitações através de qualquer página, pelo menu e seleciona o menu “Solicitações”; 2) O usuário seleciona a aba “Realizado por mim”; 3) Usuário seleciona o pedido; 4) O sistema abre o perfil do solicitante; 5) Usuário pode aceitar ou recusar a solicitação atualizada com o orçamento do provedor. 	

Fonte: Elaborado pelos autores, 2020.

Nas figuras 9, 10 e 11 são evidenciados os diagramas de caso de uso do projeto, expondo todos os casos de uso documentados de maneira assente, tipificando e demarcando todas as ações realizadas pelos atores envolvidos no projeto, tais quais os usuários.

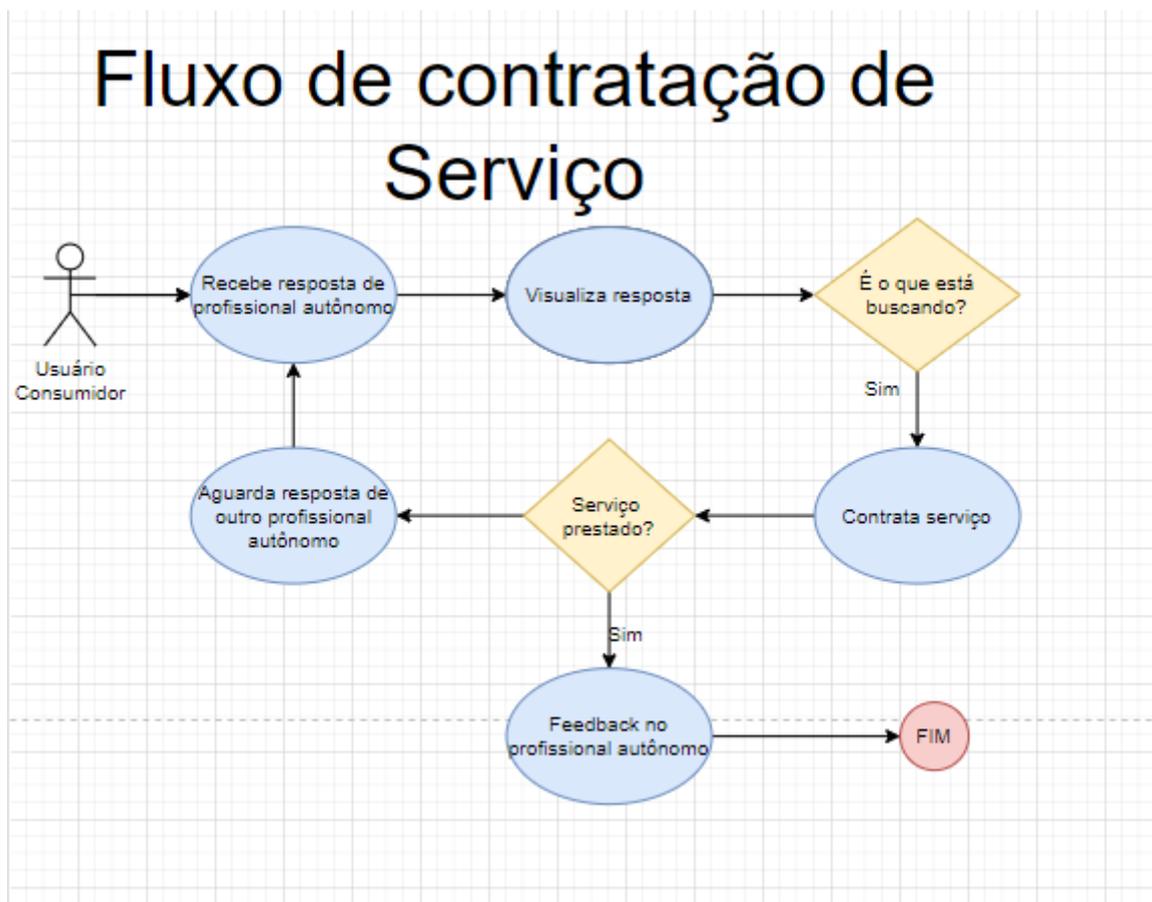
Figura 10 – Diagrama de caso de uso (fluxo de busca de serviço)



Fonte: Elaborado pelos autores, 2020.

A figura acima demonstra o fluxo de busca por um serviço utilizando o sistema desenvolvido neste trabalho. O usuário consumidor em busca de um serviço no sistema proposto visualiza, filtra e busca o serviço de sua necessidade, ao achá-lo e com a intenção de contratar, o sistema verifica o cadastro. Caso não seja cadastrado, solicita um cadastro e, então, envia uma mensagem ao profissional prestador do serviço para que ele responda com um orçamento.

Figura 11 – Diagrama de Caso de uso (fluxo de contratação de serviço)

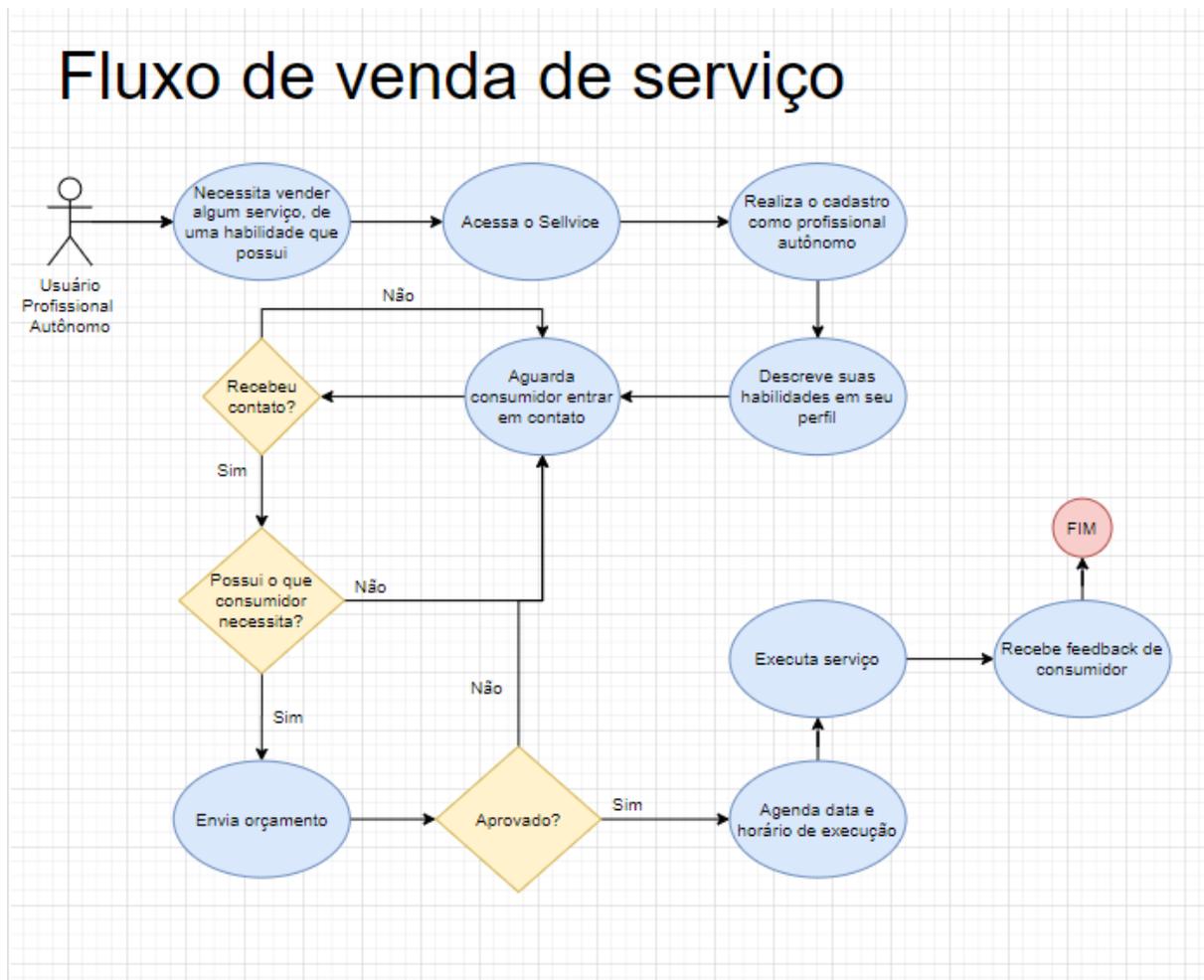


Fonte: Elaborado pelos autores, 2020.

Na figura acima, é demonstrado como o usuário realiza uma contratação de serviço dentro do sistema desenvolvido neste trabalho.

Assim que um usuário necessita de algum serviço, ele acessa o sistema e o solicita, o prestador de serviço responderá o consumidor com o orçamento, se for de seu interesse, o usuário contrata o serviço.

Figura 12 – Diagrama de Caso de Uso (fluxo de venda de serviço)



Fonte: Elaborado pelos autores, 2020.

A figura acima mostra o fluxo para um usuário cadastrar um serviço no sistema desenvolvido neste trabalho. Assim que um usuário normal tiver a necessidade de ofertar um serviço na plataforma, ele realiza um cadastro para ser reconhecido como um profissional autônomo e preenche seu perfil com suas habilidades. Quando alguém entrar em contato quanto ao serviço proposto, ele envia um orçamento ao colaborador e, assim que aprovado por ambas as partes, é agendado um dia e uma hora entre eles para a execução do serviço.

4.6 PROTOTIPAÇÃO DE TELAS

Prototipação é um processo que tem como função avaliar as ideias geradas e validar, ou não, todos os requisitos estabelecidos. (OBJECTIVE, 2019).

Porém, este processo também auxilia no entendimento do propósito do *software* desenvolvido, a regra de negócio para o cliente e propor melhorias, minimizando os riscos e maximizando os lucros. (DEXTRA, 2013).

Considera-se “a prototipação um processo importante, pois é um meio rápido e bastante eficiente para validar as ideias, uma nova abordagem ou novo conceito”. (DEXTRA, 2013).

São expostos a seguir todos os protótipos produzidos das telas do sistema, e a navegação entre as telas:

4.6.1 Protótipo da tela de Autenticação

A figura exposta a seguir apresenta a interação entre o usuário e o sistema na tela de autenticação:

Figura 13 – Protótipo da tela de Autenticação



Fonte: Elaborada pelos autores, 2020.

Caso o usuário tenha esquecido a senha pode clicar no botão “Esqueceu sua senha?” para o sistema resetar a senha e encaminhar o procedimento para o e-mail cadastrado no sistema.

Caso seja um novo usuário, deve que pressionar o botão “Não tem conta? Crie já”, o usuário é redirecionado para a tela de cadastro.

4.6.2 Protótipo da tela de Cadastro

Nesta tela o usuário pode cadastrar seus dados em nosso sistema, informando usuário, e-mail e senha:

Figura 14 – Protótipo da tela de Cadastro



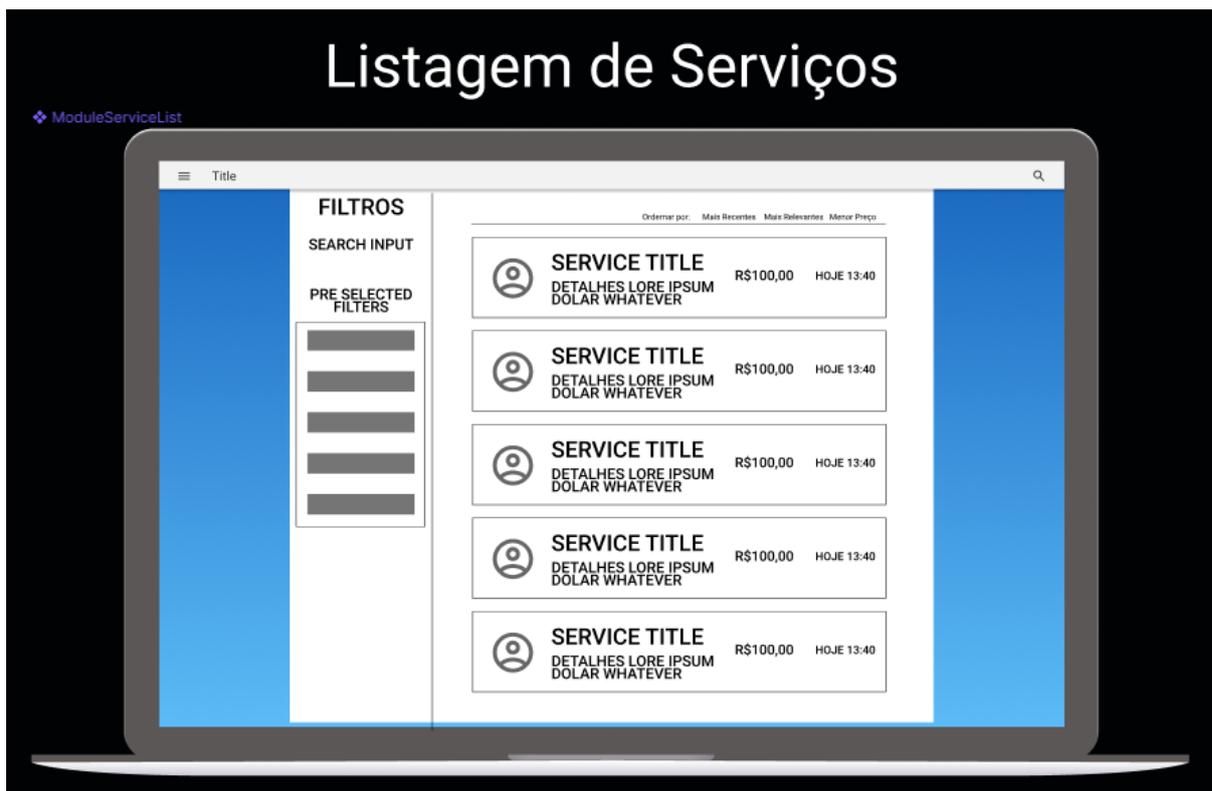
Fonte: Elaborado pelos autores, 2020.

Após finalizar o cadastro o usuário é redirecionado para a tela principal do sistema, que é a lista de serviços.

4.6.3 Protótipo da tela de listagem de serviços

A tela de listagem de serviços é a tela principal do sistema. O funcionamento ocorre de forma com que a listagem em banco se altera e a alteração é mostrada na tela.

Figura 15 – Protótipo da tela de listagem de serviços



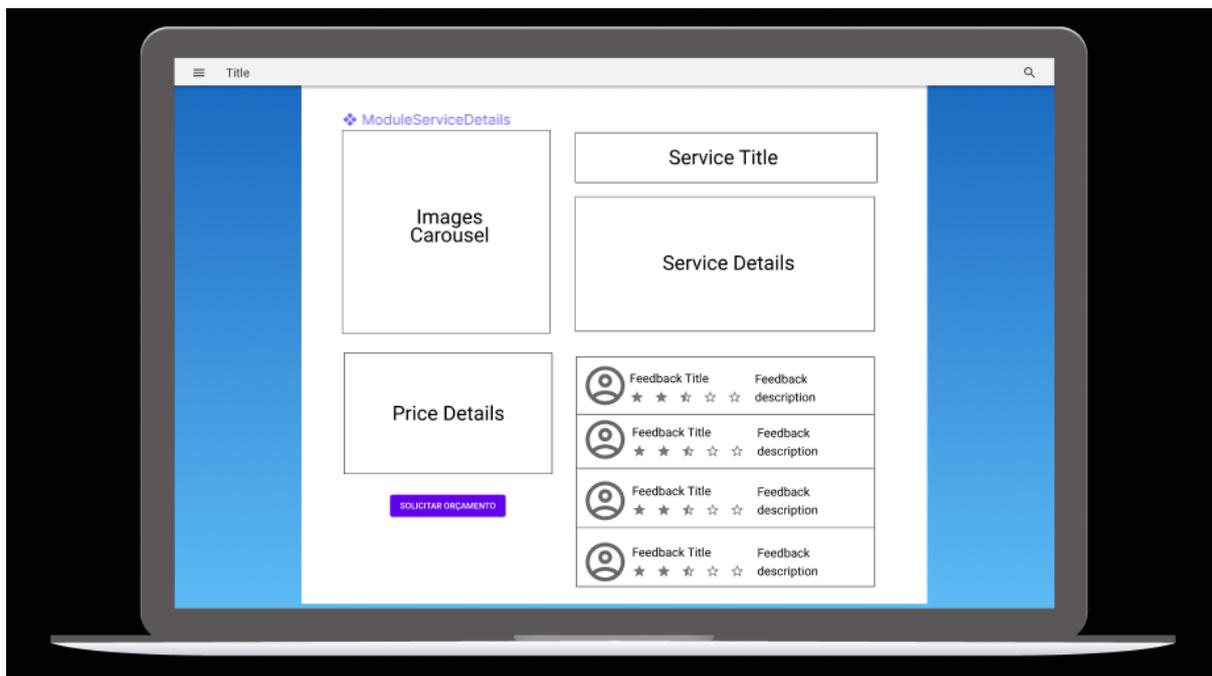
Fonte: Elaborado pelos autores, 2020.

O usuário é redirecionado para a tela de detalhes do serviço quando selecionar o serviço.

4.6.4 Protótipo da tela de detalhes de serviço

É nesta tela que o usuário pode visualizar os detalhes dos serviços, assim que selecionar um serviço na tela de listagem:

Figura 16 – Protótipo da tela de detalhes de serviço



Fonte: Elaborado pelos autores, 2020.

Ao clicar em “Solicitar orçamento”, o usuário envia uma solicitação ao provedor do serviço.

4.6.5 Protótipo da tela de criação e edição de um serviço

Na tela de edição e de criação de um serviço é onde o dono do serviço pode alterar qualquer informação necessária:

Figura 17 – Protótipo da tela de criação

Criação/Edição de Serviço

Titulo

Descrição

Preço/hora

Área de atuação

Santa Catarina Palhoça São José

Upload Images

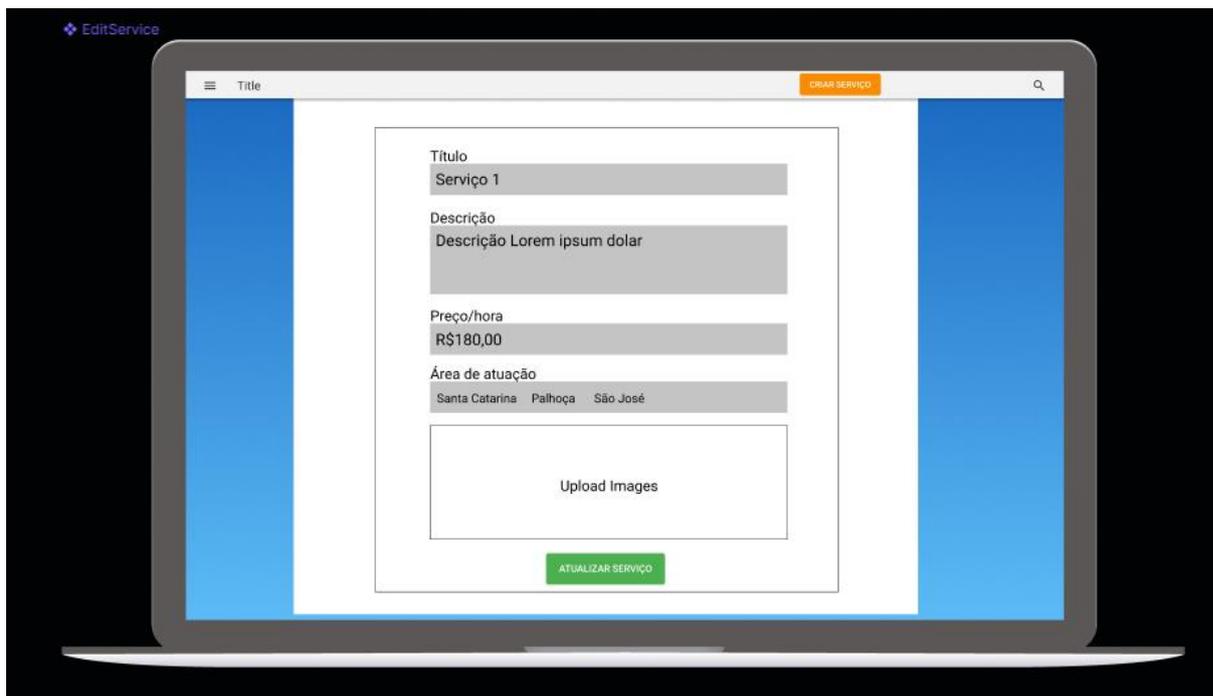
CRIAR SERVIÇO

Fonte: Elaborado pelos autores, 2020.

Como pode ser percebido na tela anterior, os dados essenciais para o cadastro de um serviço são: título, descrição, preço por hora, área de atuação e fotos relacionadas. Após clicar no botão “Criar serviço”, o sistema cria o serviço, liberando o mesmo em todas as buscas, ficando disponível para edição a qualquer momento pelo proprietário.

Segue imagem a seguir da tela de edição.

Figura 18 – Protótipo da tela de edição de serviço



Fonte: Elaborado pelos autores, 2020.

Ao atualizar clicando no botão “atualizar serviço”, o serviço é atualizado em banco e a partir daquele momento o serviço é mostrado com os dados atualizados para todos os usuários do sistema.

5 DESENVOLVIMENTO

Neste capítulo são demonstradas as ferramentas e a sua utilização, apresentadas dentro do escopo de desenvolvimento do sistema, assim como métodos e padrões aplicados.

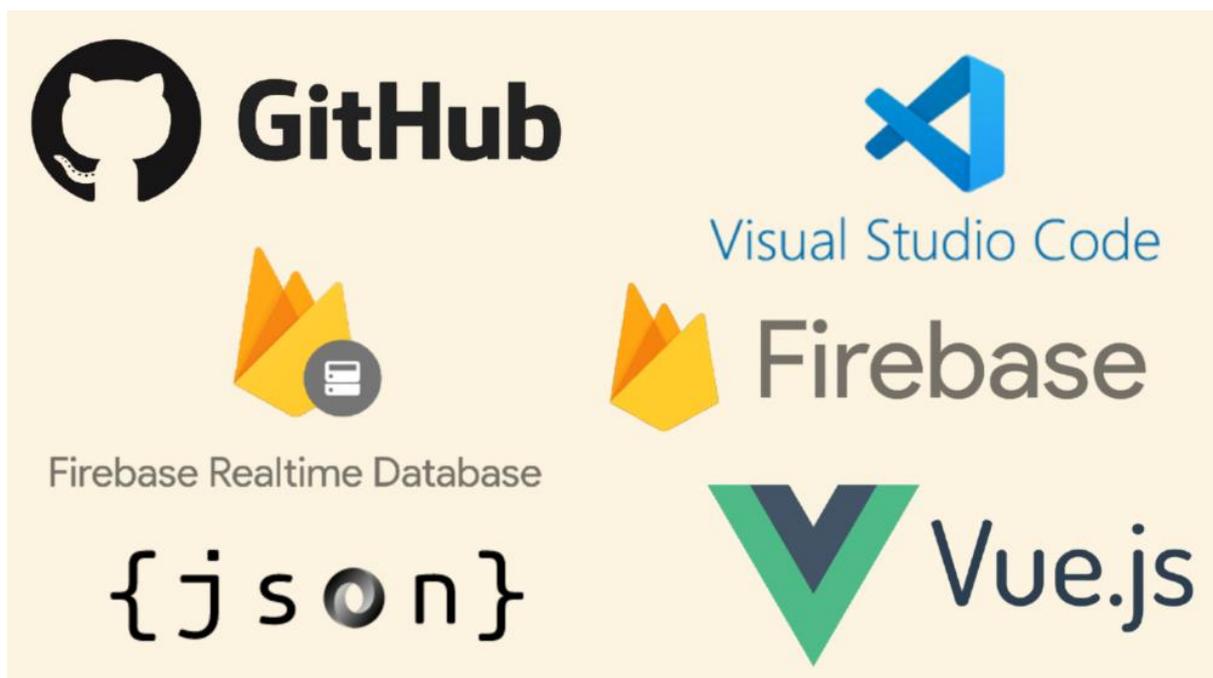
Para o desenvolvimento do sistema é utilizado o *framework* Vue.JS, que faz uso da linguagem Java Script, CSS e HTML. O Vue.JS foi escolhido pelos seus benefícios em agilidade no desenvolvimento e performance, visto que os autores possuem experiência neste framework.

5.1 TECNOLOGIAS E FERRAMENTAS

Com a modelagem do sistema definida, junto com seus requisitos funcionais e não funcionais, casos de uso e regra de negócio, escolhe-se as ferramentas para desenvolver o sistema.

Entre diversas ferramentas pesquisadas foram escolhidas as que mais atendem as necessidades para desenvolver o sistema proposto neste trabalho. As tecnologias escolhidas estão na figura a seguir.

Figura 19 – Tecnologias utilizadas



Fonte: Elaborado pelos autores

Para conhecimento das tecnologias, a seguir é elucidado os motivos de escolha para cada uma das tecnologias.

5.1.1 Firebase

Serviços em nuvem para desenvolvedores. É como se já existisse um *backend* completo para as aplicações, com visual limpo e de fácil uso. O Firebase é uma plataforma dedicada para construção de aplicativos. (BATTISTELLI, 2018)

5.1.1.1 Firebase Realtime Database

Esta tecnologia é escolhida por trazer facilidade em desenvolver o sistema, isto é, agilidade no desenvolvimento.

O ambiente do *Firebase* proporciona interação dinâmica dos usuários, cria análises de resultados, notifica em várias plataformas, possui suporte múltiplas plataformas, direito de uso gratuito e pago, armazenamento em nuvem, servidor para testar os aplicativos antes do lançamento, análise de erros em tempo real, autenticação, etc.

Com tantas funcionalidades, pode-se afirmar que pode otimizar os aplicativos como quiser. Por ser uma ferramenta digital com tantas funcionalidades, é possível um desenvolvimento mais eficaz em diferentes dispositivos e para diferentes plataformas.

O *Firebase Realtime Database* é um serviço do *Firebase* que possui sincronização de dados em tempo real em múltiplas plataformas, desenvolvendo sem necessidade de um *backend* propriamente dito, pois o *Firebase Realtime Database* propõe-se em realizar este trabalho, otimizando *caches* nos dispositivos para facilitar o uso *offline*. Assim que o usuário voltar a ficar *online*, os dados são sincronizados novamente com a segurança de acesso e de dados da própria *Google*.

Além da possibilidade de trabalhar com dados em tempo real e em múltiplas plataformas, o sistema monta o banco e com base na modelagem das tabelas e atributos é gerado, em formato de *API*, as requisições com o banco, tornando desnecessário o desenvolvimento de um *backend* bem estruturado. Sendo assim, proporciona agilidade no desenvolvimento de um *MVP* (produto viável mínimo).

Esta ferramenta é necessária para armazenar, editar, registrar e consultar os dados que são armazenados no próprio banco de dados.

5.1.2 JSON

JSON é um formato de representação de dados, objetos e atributos baseado em Java Script, por isso JSON significa *Java Script Object Notation* – Notação de Objetos em Java Script. Hoje em dia, a maior parte das linguagens, e alguns bancos de dados, possuem suporte para o JSON. (ALVES, 2018).

Existem banco de dados NoSQL que armazenam os dados em formato JSON, como por exemplo, CouchDB, MongoDB e o próprio *Firebase*.

O uso se faz necessário por conta da notação que o banco de dados do *Firebase* utiliza, o JSON, e ao *framework* de desenvolvimento das interfaces que é em Java Script.

5.1.3 GitHub

GitHub é uma plataforma *web* de serviço *git*, gerenciamento de projetos, arquivos e versionamento de códigos oferecendo funcionalidades extras. É uma plataforma gratuita que permite hospedagem de projetos pessoais. (SCHMITZ, 2018)

É necessário o uso do GitHub para manter o sistema organizado e o código com versões atualizadas, caso ocorra algum problema, para ter um sistema para voltar o código e resolver o problema.

5.1.4 Visual Studio Code

Microsoft lançou em 2015 esta aplicação gratuita com o intuito de ajudar e facilitar o desenvolvimento *web*. Possuindo suporte para mais de 30 linguagens de programação e disponibilidade nos sistemas operacionais Windows, MacOS e Linux. (MACORATTI, 2016).

Essa ferramenta é leve e opera em diversas linguagens, sendo assim um ambiente perfeito para trabalhar com Vue, pois ele utiliza de HTML, CSS e Java Script, como é descrito no capítulo a seguir, por isso a necessidade de usá-la.

5.1.5 Vue.JS

Um *framework* montado em Java Script de código aberto, lançado em fevereiro de 2014. Muito utilizado para criar aplicações de página única e interfaces que possuem necessidades de maior interação e experiência para o usuário.

Vue.JS é um dos frameworks mais populares do mundo e a aprovação é entorno de 90%. Possui características como versatilidade, modularização e a facilidade de aprender. (PICOLLO, 2020).

O *framework* utiliza arquivos '.vue', que são constituídos de HTML, CSS e Java Script, facilitando o isolamento dos componentes e a manutenção das funcionalidades. Cada componente é declarado por um escopo isolado tanto de lógica quanto de estilo. A renderização da página é feita em um documento virtual, que é atualizado sempre que um componente sofre alteração. (PICOLLO, 2020).

A escolha deste *framework* foi feita por conta da facilidade, do desempenho e das experiências já existentes entre os desenvolvedores e autores deste trabalho.

5.2 HISTÓRICO DO DESENVOLVIMENTO

O projeto surgiu em uma conversa entre os autores e desenvolvedores deste trabalho no que diz respeito à dificuldade que pessoas autônomas tem em garantir clientes e oferecer serviços, principalmente para pessoas que não possuem CNPJ. Após discussão, foi realizada validação entre amigos e familiares.

Ao transcorrer do projeto foram levantados os requisitos com foco total em usabilidade, acessibilidade, utilização e venda da ideia de que todos possuem *hobbies* ou algo que saibam fazer e possam fazer deles algo rentável, a fim de atingir o maior número de pessoas possíveis e contribuir com pessoas que realmente precisam de uma renda ou até mesmo um complemento.

Contudo, a utilização do *framework* Vue.JS é exclusivamente por conta da experiência dos autores com o *framework*. Com relação ao *backend*, várias ideias vieram à tona, entre elas: AWS, Node.JS, *Microservice multi-language*, Go, entre outras. No entanto, a escolha foi baseada em facilidade de aprendizado, desenvolvimento ágil, entregas de valores rápidas e contínuas, sempre buscando utilizar tecnologias modernas no ambiente de desenvolvimento.

Analisando e estudando garantir o sistema o mais fácil possível, com usabilidade e boas experiências dos usuários, para o cadastro de um usuário comum, é exigido o mínimo de informações possíveis para garantir a maior segurança possível dos usuários, dessa forma ele passará por um segundo cadastro para validação.

5.3 APLICATIVO DESENVOLVIDO

No presente capítulo é apresentado algumas telas já desenvolvidas, conforme os requisitos e protótipos levantados no capítulo 4. No decorrer da seção de modelagem foram apresentadas as telas em forma de protótipos, bem como suas devidas funcionalidades e, nesta seção, as telas desenvolvidas com o *framework* Vue.JS.

O *framework* Vue.JS, entre tantos outros de boa qualidade, vem em alto crescimento e pode-se dizer que é o que mais possui aceitação entre os desenvolvedores. (PICOLLO, 2020)

5.3.1 Tela de Login

Figura 20 – Tela de Login

A imagem mostra a tela de login do aplicativo Sellvice. O lado esquerdo tem um fundo azul com o texto 'Sellvice' e uma mensagem de boas-vindas. O lado direito tem um fundo branco com o título 'Bem vindo de volta!' e um formulário de login com campos para 'Email' e 'Senha', um botão 'Entrar' e links para 'Esqueci a senha' e 'Criar uma conta'.

Sellvice

Bem vindo ao Sellvice, uma aplicação desenvolvida especialmente para você, que precisa encontrar um especialista para executar um serviço.

Bem vindo de volta!

Email
Digite aqui seu email...

Senha

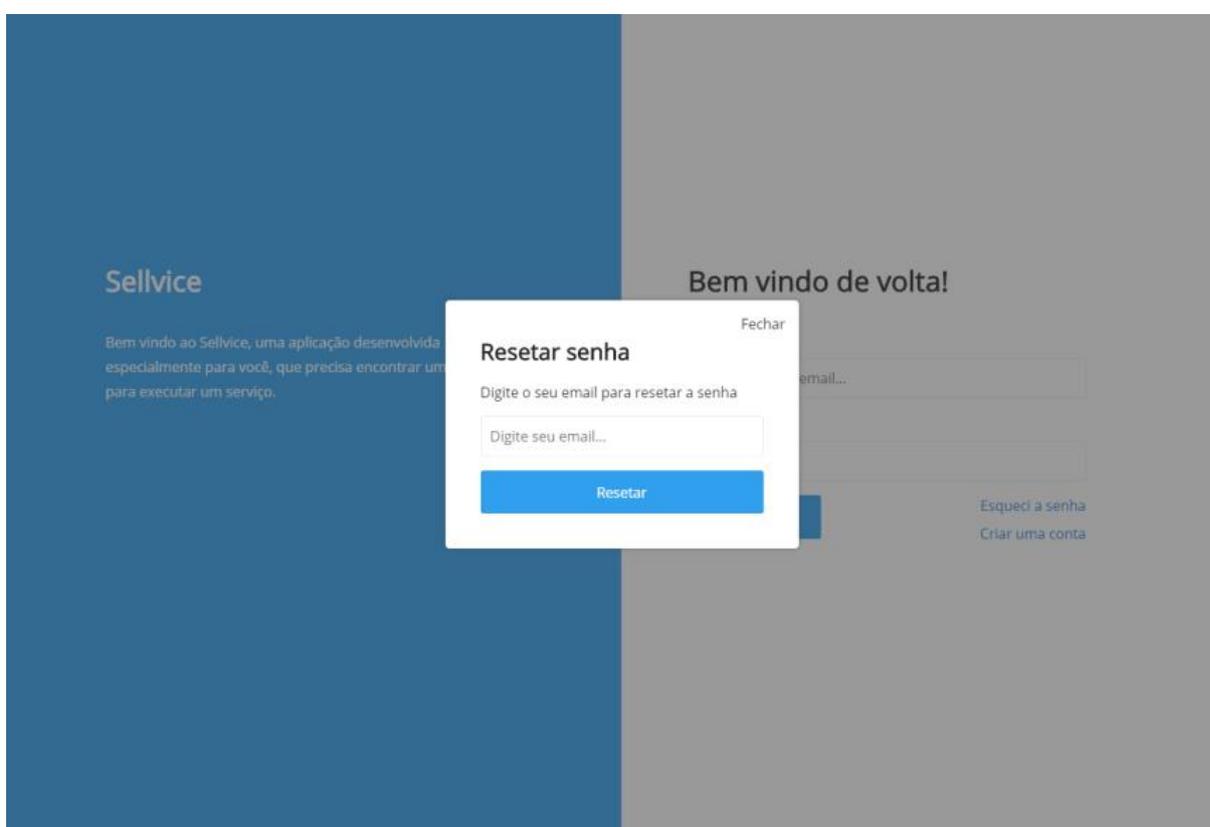
[Entrar](#) [Esqueci a senha](#) [Criar uma conta](#)

Fonte: elaborado pelos autores, 2020.

Nesta tela o usuário tem a possibilidade de criar uma conta ou caso tenha esquecido a senha, possui a possibilidade de recuperá-la.

Logo, caso o usuário tenha perdido a senha, pode clicar no botão “Esqueci a senha” para o sistema redefini-la e encaminhar o procedimento para o e-mail cadastrado no sistema. Informando o e-mail, a senha é restaurada e enviada ao e-mail.

Figura 21 – Tela de “Esqueci a senha”



Fonte: elaborado pelos autores, 2020.

Caso seja um novo usuário, pressionando o botão “Criar uma conta”, o usuário é redirecionado à tela de cadastro.

5.3.2 Tela de Cadastro

Nesta tela o usuário pode cadastrar seus dados em nosso sistema, informando primeiro nome, sobrenome, e-mail e senha:

Figura 22 – Tela de Cadastro

Sellvice

Bem vindo ao Sellvice, uma aplicação desenvolvida especialmente para você, que precisa encontrar um especialista para executar um serviço.

Cadastro

Primeiro Nome
Digite aqui seu primeiro nome...

Sobrenome
Digite aqui seu sobrenome...

Email
Digite aqui seu email...

Senha
Mínimo 6 caracteres

Cadastrar [Voltar para Login](#)

Fonte: elaborado pelos autores, 2020.

Após finalizar o cadastro o usuário é redirecionado para a tela principal do sistema, que é a lista de serviços.

5.3.3 Tela de listagem de serviços

A tela de listagem de serviços é a tela principal do sistema.

Figura 23 – Tela de listagem de serviços

The screenshot shows the Selvice interface. At the top, there are buttons for 'ANUNCIAR' (orange) and 'JEFF' (blue). Below the header, there is a 'Filtros' (Filters) section on the left with a search input, dropdowns for 'Estados' (States) and 'Cidades' (Cities), and price range fields for 'Preço min.' and 'Preço max.' both set to 'R\$ 0,00'. A green 'BUSCAR' button is at the bottom of the filters. The main content area displays a list of services:

	Posso ser seu motoboy Faço o que for necessário! Pegar algo, levar algo, entregar algo, uber! É MOTO! Sempre use seu capacete!	Preço/Hora R\$10,00 Criado em: 17/10/2020 19:59:49
	Lavo sua calçada e área Eu lavo sua calçada e área. Levo meus produtos e minha máquina!	Preço/Hora R\$25,00 Criado em: 17/10/2020 19:42:10
	Formatação de computadores Formatação de computadores em toda a região da grande Florianópolis, instalação de todos os drivers e última versão do Windows atualizada.	Preço/Hora R\$30,00 Criado em: 17/10/2020 19:39:01
	Carona para o centro de Florianópolis Trabalho todos os dias, saio do Aririú, Palhoça e vou até Florianópolis! EU TE PEGO ONDE MARCAR COMIGO! Saída as 05:30, chegada em média às 06:30 até umas 07:00	Preço/Hora R\$10,00 Criado em: 17/10/2020 19:49:23

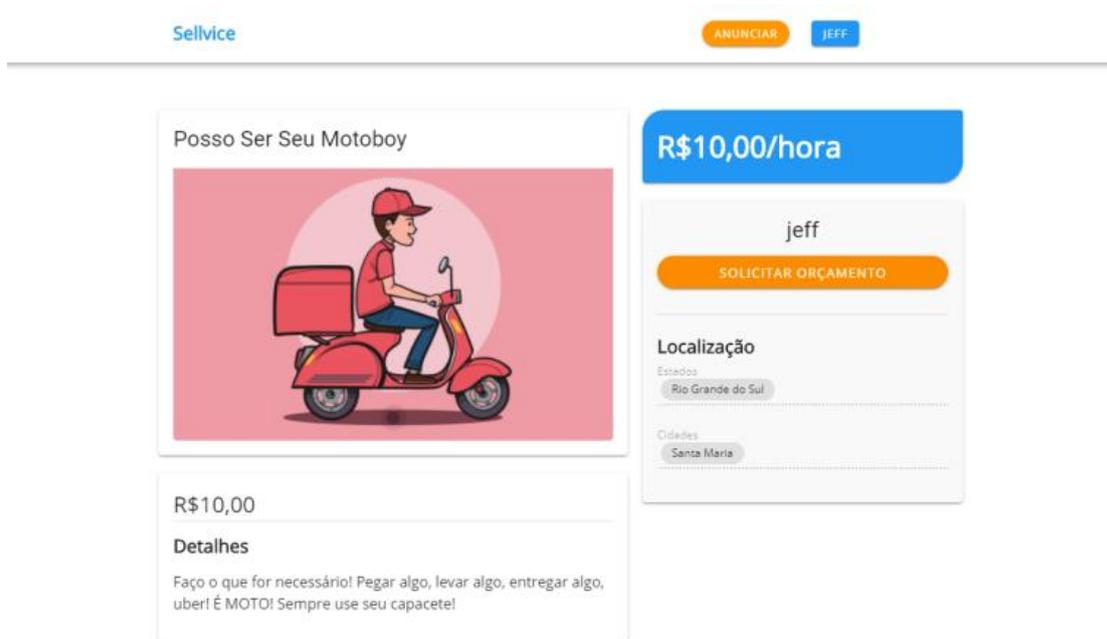
Fonte: elaborado pelos autores, 2020.

Na parte da esquerda da página tem os filtros para os serviços, sendo estes o nome do serviço, estados, cidades, preços mínimos e preços máximos.

5.3.4 Tela de detalhes do serviço

Na tela de detalhes é onde o usuário visualiza os detalhes dos serviços, assim que selecioná-lo na tela de listagem.

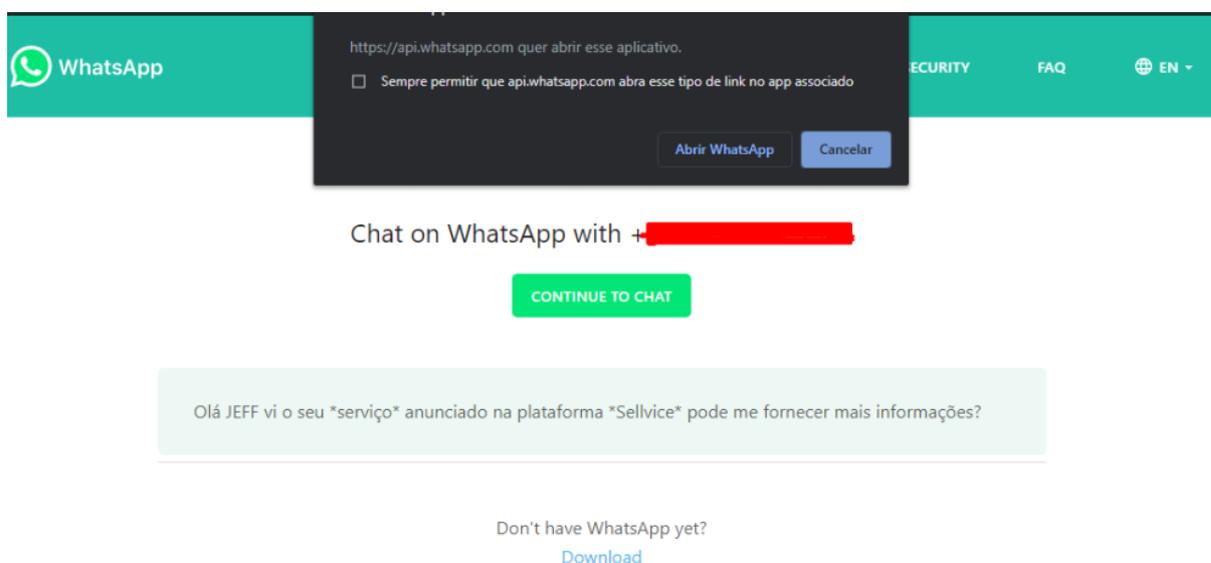
Figura 24 – Tela de detalhes do serviço



Fonte: elaborado pelos autores, 2020.

Ao clicar em “solicitar orçamento”, o usuário entra em contato direto com o provedor do serviço e é redirecionado ao *whatsapp* para conversar com o provedor do serviço.

Figura 25 – Tela redirecionada para o *whatsapp*



Fonte: elaborado pelos autores, 2020.

5.3.5 Tela de Criação e Edição de serviço

A tela de edição e criação de serviços do sistema são praticamente as mesmas, diferenciando-se apenas na possibilidade de edição dos dados, visto que estes já veem preenchidos.

Figura 26 – Tela de detalhes do serviço

Sellvix ANUNCIAR JEFFERSON

Criar um Serviço

Preencha as informações do serviço:

Título
Formatação de computadores

Descrição
Formatação de computadores em toda a região da grande Florianópolis, instalação de todos os drivers e última versão do Windows atualizada.

Preço por hora de serviço:
R\$ 30,00

Estados
Santa Catarina

Cidades
Florianópolis Palhoça São José

Imagens
640x361...cf2.jpg (39.3 kB)

FORMATAÇÃO

CRIAR SERVIÇO

Fonte: Elaborado pelos autores, 2020.

Aqui há possibilidades de colocar dados como título, descrição, preço por hora, estados e cidades atendidas e imagens.

5.4 AVALIAÇÃO DO APLICATIVO

Nesta seção a avaliação realizada sobre o sistema é apresentada. Foi feito e aplicado um questionário e com ele foram obtidas 23 respostas. Assim, é verificado o posicionamento de possíveis usuários do sistema proposto.

5.4.1 Cenário do questionário

O questionário foi realizado para pessoas próximas dos autores, como amigos, colegas de trabalho e familiares, todos com idade entre 20 e 42 anos, sendo a maior parte do sexo masculino.

5.4.2 Elaboração do questionário

A elaboração da avaliação foi feita com total embasamento na satisfação dos usuários em 5 questões, conforme quadro XXII adiante.

As perguntas realizadas foram baseadas na usabilidade, facilidade de uso do sistema, quanto ao objetivo do sistema, se ele foi atingido ou não, bem como possíveis sugestões.

As respostas para quatro delas foram elaboradas na escala Likert¹¹, sendo:

- 1) Discordo totalmente;
- 2) Discordo parcialmente;
- 3) Indiferente;
- 4) Concordo parcialmente;
- 5) Concordo totalmente.

A quinta pergunta é aberta para qualquer tipo de comentários, críticas, sugestões e alguma sugestões para o sistema.

Para a avaliação acontecer, foi utilizada a plataforma da Google, Forms¹², por ser uma plataforma gratuita e de fácil uso. O questionário foi disponibilizado por uma semana (O questionário está disponível e pode ser visualizado através do anexo A).

A seguir, são apresentadas as perguntas que foram elaboradas para o questionário:

Quadro XXII – Quadro das perguntas do questionário

Número	Perguntas
1	O sistema é um protótipo/MVP. Você acha que a interface deve permanecer neste padrão? Ela é amigável e adequada para o seu uso?
2	Você acredita que a proposta deste aplicativo pode auxiliar pessoas a garantirem rendas?
3	Pode-se dizer que a interface de listagem de serviços agiliza a pesquisa?

¹¹ Escala Likert - Uma escala tipo Likert é constituída por questões que o respondente além de concordar ou não, apresenta o grau de intensidade das respostas (CUNHA, 2007; ALEXANDRE et al., 2003).

¹² Google Forms - <https://www.google.com/forms>

4	Se o aplicativo fosse um sistema mais completo e finalizado, você usaria?
5	Comente a seguir: sugestões, críticas, eventualmente propostas de trabalhos futuros. Principalmente para respostas qual foi marcada "discordo".

Fonte: elaborado pelos autores, 2020.

5.4.3 Aplicação do questionário

O questionário foi aplicado com objetivo de verificar se o aplicativo proposto atende às expectativas dos usuários e, caso haja, identificar possíveis melhorias.

Para a aplicação do questionário, foram elaboradas etapas anteriores a sua aplicação:

- Apresentação inicial das funcionalidades propostas;
- Disponibilizar o sistema para que os usuários possam conhecer o sistema;
- E então, aplicação do questionário.

5.4.4 Análise dos resultados

Nesta seção são apresentados os resultados do questionário para verificar se a proposta é válida e aceita por possíveis usuários e, assim, realizar conclusões para obter conhecimento de que o sistema é útil para seu uso no dia a dia.

De acordo com os dados obtidos, que se encontra no quadro XXIII, no anexo B, parte-se à análise dos resultados com finalidade de obter uma conclusão.

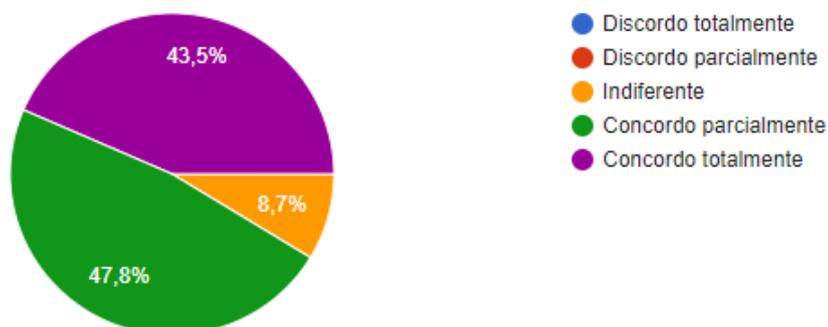
5.4.5 Considerações da avaliação

Com o questionário finalizado, pode-se verificar os resultados obtidos através da plataforma e gráficos apresentados a seguir:

Figura 27 – Questão número um

O sistema é um protótipo/MVP. Você acha que a interface deve permanecer neste padrão? Ela é amigável e adequada para o seu uso?

23 respostas



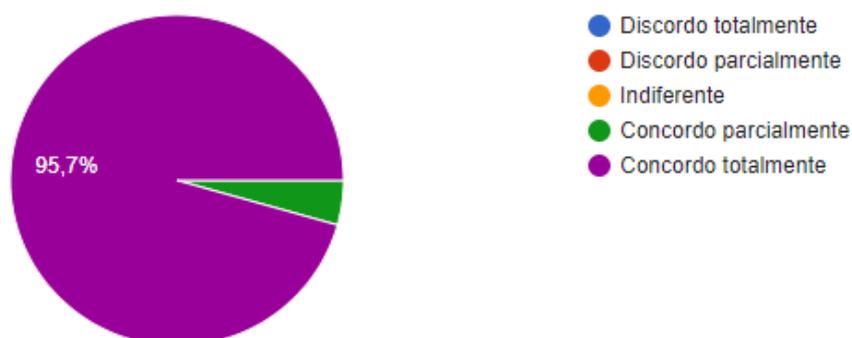
Fonte: elaborado pelos autores, 2020.

Nenhum dos entrevistados discordou, nem mesmo parcialmente, com um aproveitamento de 91,3%, isto demonstra uma grande aceitação do layout do sistema.

Figura 28 – Questão número dois

Você acredita que a proposta deste aplicativo pode auxiliar pessoas a garantirem rendas?

23 respostas



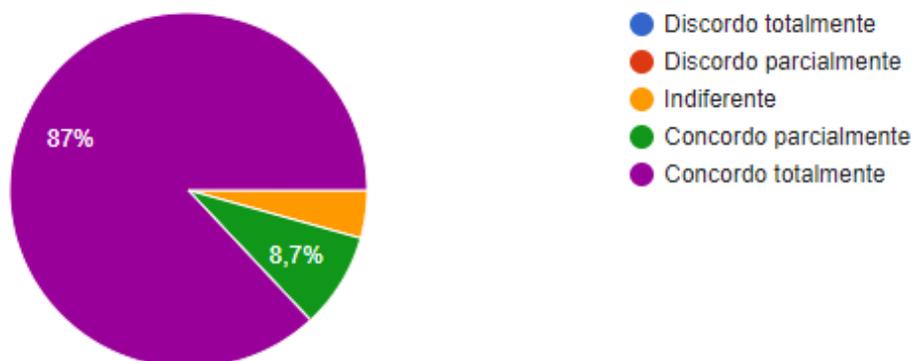
Fonte: elaborado pelos autores, 2020.

Nesta questão, houve um aproveitamento de 100%, sendo apenas de 4,3% de “concordo parcialmente”, quanto a proposta do aplicativo realmente atender a sua expectativa.

Figura 29 – Questão número três

Pode-se dizer que a interface de listagem de serviços agiliza a pesquisa?

23 respostas



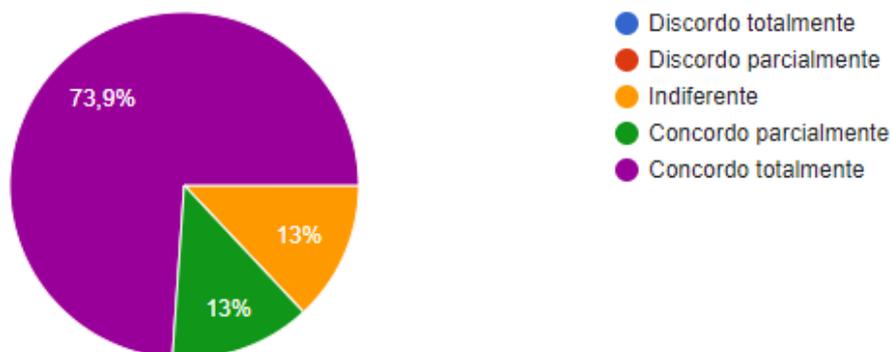
Fonte: elaborado pelos autores, 2020.

Na terceira questão, houve uma aprovação de 95,7% e apenas 4,3% de indiferença quanto à agilidade do motor de busca da listagem de serviços.

Figura 30 – Questão número quatro

Se o aplicativo fosse um sistema mais completo e finalizado, você usaria?

23 respostas



Fonte: elaborado pelos autores, 2020.

Com 87% de concordância, significa que esperam um sistema mais completo para o uso, e que 13% usariam independente do sistema estar completo ou não.

Para a última questão, por ser aberta, foram apresentadas algumas sugestões, declaradas no quadro a seguir:

Quadro XXIII – Quadro com melhores sugestões

Entrevistado #	Sugestão
#7	Separação por categorias de serviços acho que ficaria melhor para encontrar o que precisa
#8	Ideia muito legal, forma de avaliação de um determinado prestador de serviço seria interessante.
#10	Achei a proposta do aplicativo bem útil, porém não me agradou o layout dele... Acho que há grandes oportunidades de melhoria
#12	Fazer um ranking entre os melhores prestadores de serviços e os oferecer prêmios para chamar pessoas a plataforma
#13	Melhoria a parte busca, deixaria mais completo com filtro de localização por mapa.
#16	estrelas para qualidade de serviço
#17	Filtro por categorias
#23	A proposta é bem boa, a interface ficou simples e fácil de utilizar, de ponta a ponta é muito fácil entrar em contato com uma pessoa que está oferecendo um determinado serviço.

Fonte: elaborado pelos autores, 2020.

Com toda a análise, verifica-se a aprovação do sistema para todos os entrevistados. 91,3% dos entrevistados concordam com a interface do jeito que ela foi projetada. Quanto a utilidade e funcionalidade do sistema, a aprovação é de 100%, já quanto ao design e agilidade da tela de design, a aprovação foi de 95,7%.

Porém algumas sugestões para esta tela foram recomendadas pelos entrevistados e 87% das pessoas ainda esperam um sistema mais finalizado e completo, porém com a questão aberta pode-se dizer que há a necessidade de ajustes para um sistema mais completo.

6 CONCLUSÕES E TRABALHOS FUTUROS

Esta monografia é desenvolvida através de uma problemática e um mercado em expansão, juntamente com conceitos, plataformas e ferramentas sólidas e bem vistas pelo mercado de desenvolvimento atual. Contudo, a possibilidade da criação da solução se tornou viável com sua função de auxiliar a sociedade na busca de um complemento ou geração de renda, o que hoje em dia é um fator comum na sociedade.

Durante a análise do projeto, compreende-se que os diagramas, os requisitos e regras de negócios são ótimas ferramentas para mapear o sistema como todo, juntamente com suas funcionalidades e fluxos de serviços. Portanto, possibilitando previsão de problemas e erros que seriam fatores de atraso, reduzindo tempo e problemas com modelagem de banco e com a codificação, juntamente com a execução de testes unitários de *frontend* com os quais evitam se erros indesejados.

Com o desenvolvimento do sistema e da monografia, os autores adquiriram e trocaram experiências muito relevantes, agregando o desenvolvimento profissional.

O sistema pronto demonstrou que a ferramenta é intuitiva, fácil de usar e que ainda tem espaço para agregar mais valor, além do fato de auxiliar pessoas em busca de serviços, como também de pessoas que buscam complementar ou gerar renda.

A ideia do sistema surgiu em uma necessidade real da sociedade, observada pelos autores, levantada e discutida, demonstrando-se extremamente útil, principalmente no atual momento de pandemia. Com o sistema pronto, há grande interesse dos autores, mesmo após a conclusão da presente monografia, em continuar o atual trabalho no sistema e abrir possibilidades de ganhos monetários.

6.1 TRABALHOS FUTUROS

O projeto aqui discorrido permite trabalhos futuros, visando aprimoramento do sistema, como:

- Desenvolver integração com propagandas no sistema, visando o ganho monetário;

- Desenvolvimento de métodos de pagamentos internos com parcerias de sistemas de pagamento, tais como: PicPay ou MercadoPago;
- Desenvolvimento de uma aplicação mobile para uso de aplicativos em *smartphones*;
- Desenvolvimento do *backend* com testes unitários e centralizado em forma de API e/ou microsserviços;
- Realizar campanhas de marketing para divulgação do sistema, para assim conseguir maior volume de usuários e serviços;
- Desenvolver avaliação do serviço e prestador de serviço;
- Desenvolver categorias para os serviços e filtro para eles;
- Desenvolver sistema inteligente de verificação de cadastro de serviços
- Desenvolver sistema inteligente de verificação de cadastro de serviços;
- Desenvolver sistema de avaliação de serviço e prestador de serviço.

REFERÊNCIAS

ALVES, Gustavo Furtado de Oliveira. **O mínimo que você precisa saber sobre JSON para ser um bom programador!** Disponível em: <<https://dicasdeprogramacao.com.br/o-que-e-json/>>, 5 set. 2018. Acesso em: 25 set. 2020.

BATTISTELLI, Juliana. **Google Firebase for dummies: o que é e como funciona a plataforma.** Disponível em: <<https://blog.mastertech.com.br/tecnologia/google-firebase-for-dummies-o-que-e-e-como-funciona-plataforma/>>, 6 abri. 2018. Acesso em: 25 set. 2020.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML.** São Paulo: Brooklin, 2007.

CORDEIRO, J. C. M. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML.** 5ª Edição, Rio de Janeiro: Brasport, 2010.

DEXTRA. **Prototipação e sua importância no desenvolvimento de software.** Disponível em: <<https://dextra.com.br/blog/prototipacao-e-sua-importancia-no-desenvolvimento-de-software>>, 05 nov. 2013. Acesso em 17 set. 2020.

FONSECA, J. J. S. **Metodologia da pesquisa científica.** Fortaleza: UEC, 2002. Apostila.

FOWLER, Martin. **UML Essencial: Um Breve Guia para Linguagem Padrão.** 2005.
FRIAS, Thiago. **React vs Vue vs Angular: qual escolher?** Disponível em: <<https://blog.geekhunter.com.br/react-vs-vue-vs-angular-qual-escolher/>>. Acesso em: 4 jun. 2020.

GAMMA, Erich, et al. **Design Patterns: Elements of Reusable Object-Oriented software.** 37. ed. Addison-Wesley: Massachusetts, 1995.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Número de pessoas que trabalham em veículos cresce 29,2%, maior alta da série.** Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/26424>>. Acesso em: 10 abr. 2020.

LATTE, Björn; HENNING, Sören; WOJCIESZAK, Maik. **Clean Code: On the Use of Practices and Tools to Produce Maintainable Code for Long-Living Software.** In: EMLS 2019: Collaborative Workshop on Evolution and Maintenance of Long-Living Systems, 6., 2019, Stuttgart, Paper. Stuttgart: Kiel University, 2019. p. 96-99.

LEFFINGWELL, D. **Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise,** 1st edition. Boston: Addison-Wesley Professional, 2011.

LIMA, Edilaneide Justiano de. **Comércio Informal: um estudo sobre possíveis contribuições na Economia e na Renda Familiar.** 2017. Trabalho de Conclusão de

Curso (Bacharelado em Administração) – Universidade Federal da Paraíba, João Pessoa, 2017.

MACORATTI, José Carlos. **Visual Studio Code – Apresentando o editor multiplataforma da Microsoft.** Disponível em: <<https://imasters.com.br/desenvolvimento/visual-studio-codeapresentando-o-editor-multiplataforma-da-microsoft>>, 22 jul. 2016. Acesso em: 26 set. 2020.

MACRAE, Callum. **Vue.js Up & Running: Building accessible and performant web apps.** 3. ed. Sebastopol: O'Reilly, 2018.

MASSÉ, Mark. **REST API: Design Rulebook.** 1. ed. Sebastopol: O'Reilly, 2012.

MARTIN, Robert C. **Clean Code: A Handbook of Agile Software Craftsmanship.** Prentice Hall: Nova Jersey, 2009.

MATSUKUMA, Marcos. **Produção e avaliação de interfaces, segundo princípios da interação humano-computador.** 2012. Dissertação (Mestrado em Ensino de Ciência) – Universidade de São Paulo: São Paulo, 2012. Disponível em: <https://teses.usp.br/teses/disponiveis/81/81131/tde-23102012-134939/publico/Marcos_Matsukuma.pdf>. Acesso em: 12 maio 2020.

MEDEIROS, Higor. **Introdução aos requisitos de software.** 2013. Disponível em: <<https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>>. Acesso em: 28 jun. 2020.

MINAYO, Maria Cecília de Souza. **O desafio do conhecimento.** São Paulo: Hucitec, 1993.

NITAHARA, Akemi. **Informalidade no mercado de trabalho é recorde, aponta IBGE:** Empregos informais chegam a 41,4% da força de trabalho ocupada no país. Agência Brasil, Brasília, 31 de out. 2019. Disponível em: <<https://agenciabrasil.ebc.com.br/economia/noticia/2019-10/informalidade-no-mercado-de-trabalho-e-recorde-aponta-ibge>>. Acesso em: 13 abr. 2020.

OBJECTIVE. **A Prototipação no desenvolvimento de software.** Disponível em: <<https://objective.com.br/a-prototipacao-no-desenvolvimento-de-software>>. Acesso em 17 set. 2020.

PEZZÈ, Mauro; YOUNG, Michal. **Teste e Análise de Software:** processos, princípios e técnicas. São Paulo: Bookman, 2008.

PICOLLO, Lucas. **Vue JS:** o que é, como funciona e vantagens. Disponível em: <<https://blog.geekhunter.com.br/vue-js-so-vejo-vantagens-e-voce/>>. Acesso em: 01 out. 2020.

PFLEEGER, S. L. **Engenharia de Software:** teoria e prática, 2. ed, São Paulo: Prentice Hall, 2004.

PRESSMAN, Roger S. **Engenharia de software**: uma abordagem profissional, Porto Alegre: AMGH, 2011.

PULUCENO, Thiago Vieira. **Estudo de caso sobre uma API REST utilizando a abordagem de programação orientada e eventos a plataforma Node.js**. 2012. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Universidade Federal de Santa Catarina, 2012.

REZENDE, Denis Alcides. **Engenharia de Softwares e Sistemas de Informação**. Rio de Janeiro: Sergio Marins, 2005.

RICHARDSON, L.; RUBY, S. **Restful Web Services**. Beijing: O'Reilly Media, 2007. ISBN 0596529260

RIEHLE, Dirk. **Composite design patterns**. In: OOPSLA97: Conference on Object Oriented Programming Systems Languages and Applications, 12., 1997, Atlanta, Paper. New York: Association for Computing Machinery, 1997. p. 218-228.

SCHMITZ, Daniel. **Tudo que você queria saber sobre Git e GitHub, mas tinha vergonha de perguntar**. Disponível em: < <https://tableless.com.br/tudo-que-voce-queria-saber-sobregit-e-github-mas-tinha-vergonha-de-perguntar/> >, 7 out. 2015. Acesso em: 26 set. 2020.

SILVA, E.L da; MENEZES, E.M. **Metodologia da Pesquisa e elaboração de Dissertação**. 4a Ed. UFSC, Florianópolis, 2005. Disponível em:< https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dis>. Acesso em: 27 jun. 2020.

SIQUEIRA NETO, José Francisco. **Flexibilização**, desregulamentação e direito do trabalho no Brasil. In: OLIVEIRA, Carlos Alonso B. (Org). Crise e trabalho no Brasil: modernidade ou volta ao passado?. 2. ed. São Paulo: Scritta, 1996.

SIQUEIRA, Giselly; ACCIOLY, Neri. **Flexibilização**. Revista Anamatra, v. 1, n. 53, p. 14, 2007. Disponível em: <<https://www.anamatra.org.br/attachments/article/24324/00000455.pdf>>. Acesso em: 22 abr. 2020.

SOMMERVILLE, Ian. **Engenharia de software**, São Paulo: Pearson Education do Brasil, 2011.

WIRTH, Niklaus. **A Brief History of Software Engineering**. Disponível em: < <https://people.inf.ethz.ch/wirth/Miscellaneous/IEEE-Annals.pdf> >. Acesso em: 16 maio 2020.

ANEXOS

ANEXO A – QUESTIONÁRIO

Figura 30 – Questionário do aplicativo

Sellvice - Avaliação da aplicação

Questionário para avaliar o sistema desenvolvido

***Obrigatório**

O sistema é um protótipo/MVP. Você acha que a interface deve permanecer neste padrão? Ela é amigável e adequada para o seu uso? *

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

Você acredita que a proposta deste aplicativo pode auxiliar pessoas a garantirem rendas? *

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

Pode-se dizer que a interface de listagem de serviços agiliza a pesquisa? *

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

Se o aplicativo fosse um sistema mais completo e finalizado, você usaria? *

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

Comente a seguir: sugestões, críticas, eventualmente propostas de trabalhos futuros. Principalmente para respostas qual foi marcada "discordo". *

Sua resposta

Enviar

ANEXO B – RESULTADO DA PESQUISA

Quadro XXIX – Análise dos resultados

Pessoa	Pergunta 1	Pergunta 2	Pergunta 3	Pergunta 4
#01	Concordo parcialmente	Concordo totalmente	Indiferente	Concordo totalmente
#02	Concordo totalmente	Concordo totalmente	Concordo totalmente	Indiferente
#03	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#04	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo parcialmente
#05	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#06	Concordo parcialmente	Concordo totalmente	Concordo parcialmente	Indiferente
#07	Concordo parcialmente	Concordo parcialmente	Concordo totalmente	Concordo totalmente
#08	Concordo parcialmente	Concordo totalmente	Concordo parcialmente	Indiferente
#09	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#10	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#11	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo parcialmente
#12	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo parcialmente
#13	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#14	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#15	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#16	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#17	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#18	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#19	Indiferente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#20	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#21	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#22	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
#23	Indiferente	Concordo totalmente	Concordo totalmente	Concordo totalmente

Fonte: elaborado pelos autores, 2020.