

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
FACULDADE DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Ewertom Mozart Moraes de Miranda

**STARTUP DE APLICAÇÃO MÓVEL ISERVICE –  
UMA PLATAFORMA DE DIVULGAÇÃO E PRESTAÇÃO DE  
SERVIÇOS MULTIGÊNERO**

Belém  
2017

Ewertom Mozart Moraes de Miranda

**STARTUP DE APLICAÇÃO MÓVEL ISERVICE –  
UMA PLATAFORMA DE DIVULGAÇÃO E PRESTAÇÃO DE  
SERVIÇOS MULTIGÊNERO**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Prof. Dr. Alfredo Braga Furtado.

Belém  
2017

Ewertom Mozart Moraes de Miranda

**STARTUP DE APLICAÇÃO MÓVEL ISERVICE –  
UMA PLATAFORMA DE DIVULGAÇÃO E PRESTAÇÃO DE  
SERVIÇOS MULTIGÊNERO**

Belém, 28 de setembro de 2017

Conceito:

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Prof. Dr. Alfredo Braga Furtado.

Banca examinadora

---

Prof. Dr. Alfredo Braga Furtado  
Professor Orientador

---

Prof. Dr. Raimundo Viégas Júnior  
Membro

---

Prof. M. Sc. José Maria Nascimento Bitar  
Membro

## **AGRADECIMENTOS**

À Universidade Federal do Pará

Ao Prof. Alfredo Braga Furtado, pela orientação atenciosa e precisa, e paciência pelos momentos difíceis durante a construção do trabalho.

À minha companheira de vida, que esteve ao meu lado sempre.

Aos meus pais por me educarem e fazerem, de mim, o que sou, sempre tentando fazer o meu melhor e com dedicação.

Aos meus amigos de curso que me proporcionaram ensinamentos pessoais e bons momentos para seguir na jornada com a cabeça erguida.

A Deus, pois sem fé eu não poderia chegar até aqui!

“A vida é boa quando você está feliz; mas a vida é muito melhor quando os outros estão felizes por sua causa”

Papa Francisco

## RESUMO

O objetivo deste trabalho é abordar a construção de um aplicativo móvel para divulgação e contratação de serviços, com a pretensão de resolver problemas e propor melhorias de interação entre prestadores e consumidores – problemas os quais, sob o olhar do autor, ainda não foram resolvidos com as ferramentas atualmente disponíveis para o ramo. Para tanto, procurou-se amadurecer a ideia com opinião de terceiros por meio de uma pesquisa de campo. E, para garantir a construção do projeto de software com qualidade, o desenvolvimento da ferramenta partiu desde a especificação de requisitos, progredindo para análise e projeto de sistemas, até chegar à implementação e testes da ferramenta.

**Palavras-chave:** prestadores de serviços, aplicativos móveis, consumo de serviços.

## **ABSTRACT**

The objective of this work is to discuss the construction of a mobile application for the dissemination and contracting of services, with the aim of solving problems and proposing improvements in the interaction between providers and consumers - problems that, under the author's eye, have not yet been solved with the tools currently available for the branch. In order to do so, we tried to mature the idea with the opinion of third parties through a field research. And to ensure the construction of the software project with quality, the development of the tool started from the specification of requirements, progressing to analysis and design of systems, until the implementation and testing of the ferment.

**Keywords:** service providers, mobile applications, service consumption

## LISTA DE FIGURAS

Figura 1 – Gráfico de participação percentual do setor terciário no valor adicionado, entre 2003 e 2016 (IBGE, 2016) .....	25
Figura 2 – Taxas de Desocupação no Brasil (IBGE, 2016) .....	28
Figura 3 – Questionário de Pesquisa – Pergunta com opção descritiva .....	47
Figura 4 – resultados para uma pergunta pertinente da pesquisa de campo sobre a existência do problema que a startup iservice pretende resolver .....	48
Figura 5 – Pergunta se a pessoa concorda em receber e-mail futuro para contato .....	49
Figura 6 – Diagrama de subsistemas .....	61
Figura 7 – Diagrama de casos de uso do subsistema Busca de Serviços .....	62
Figura 8 – Diagrama de casos de uso dos subsistemas prestação de serviços .....	63
Figura 9 – Fragmento do diagrama de classes, demonstrando o primeiro passo da especificação de entidades .....	66
Figura 10 – Construção parcial do diagrama de classes contemplando o cadastro de pessoa e divulgação básica de serviços .....	68
Figura 11 – Diagrama de classes de entidades completo .....	71
Figura 12 – Diagrama de atividade de atendimento .....	73
Figura 13 – Diagrama de sequência de pesquisa de serviços .....	75
Figura 14 – Diagrama de estados da classe unidade .....	76
Figura 15 – Modelo de arquitetura em camadas (PRESSMAN e MAXIM, 2016) .....	79
Figura 16 – Diagrama de implantação .....	81
Figura 17 – Diagrama de componentes .....	83
Figura 18 – Diagrama de classes de projeto do subsistema API .....	86
Figura 19 – Diagrama de classes de projeto do subsistema App .....	87
Figura 20 – Interface web do componente crud de pessoas .....	88
Figura 21 – Quadro de especificação de pseudocódigo de cada método da classe AutenticadorConcreto .....	89
Figura 22 – Quadro de especificação da interface Perfil de pessoa .....	91
Figura 23 – Protótipo da interface Perfil de usuário, demonstrando menu de acesso rápido .....	92

Figura 24 – Diagrama de comunicação da navegabilidade entre interfaces do usuário. ....	93
Figura 25 - Consulta de registro do banco de dados.....	102
Figura 26 – como uma aplicação usa JWT para verificar a autenticidade de um usuário. ....	105
Figura 27 – Formulário de cadastro de pessoa com regras de validação aplicada aos campos.....	107
Figura 28 – Resposta da API quando autenticação de usuário é bem sucedida .....	108
Figura 29 – Primeira etapa do cadastro de serviço.....	109
Figura 30 – Interface Definir unidade .....	110
Figura 31 – cadastro de horário de atendimento, no momento da seleção dos dias da semana.....	111
Figura 32 – Terceiro passo do cadastro de serviço.....	112
Figura 33 – Diálogo para agendar atendimento.....	114
Figura 34 – Andamento do atendimento. Consumidor aguardando confirmação de encontro pelo prestador.....	115
Figura 35 – Interface Conversa .....	116
Figura 36 – exemplo de teste unitário automatizado .....	118
Figura 37 – Caso de teste automatizado para consultar serviços de uma pessoa. ....	119
Figura 38 – resultados da pergunta sobre a rapidez de aprendizado .....	126
Figura 39 – resultados da pergunta sobre facilidade de uso após aprendizado .....	127
Figura 40 – resultados da pergunta sobre nota para capacidade de memorização do uso .....	128
Figura 41 – resultados da pergunta sobre nota do quanto o aplicativo é agradável .....	128
Figura 42 – resultados da pergunta sobre nota quanto a rapidez de executar as tarefas .....	129
Figura 43 – resultados da pergunta sobre não encontrar alguma opção esperada .....	130
Figura 44 – resultados da pergunta sobre não saber o que fazer em algum momento da interação.....	130

Figura 45 – resultados da pergunta sobre sentimento de ter ficado sem resposta do sistema ..... 131

## LISTA DE QUADROS

Quadro 1 – mecanismos de busca por aplicações móveis de prestação de serviços. ....	32
Quadro 2 – Definições de campos especiais de formulário pós-atendimento. .	54
Quadro 3 – Dados de cada campo de formulários de cadastro e atendimento.	57
Quadro 4 – Exemplos de confecção de formulário de cadastro. ....	57
Quadro 5 – Exemplos de confecção de formulário de atendimento. ....	58
Quadro 6 – Esquema de testes de software projetados.....	95
Quadro 7 – Problemas identificados nos registros em vídeo dos testes de usabilidade. ....	123

## LISTA DE ABREVIATURAS

MVP	Minimum Viable Product
REST	Representational State Transfer
PO	Persistent Object
API	Application Programming Interface
UML	Unified Modeling Language
MVC	Model View Controller
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
GPS	Global Positioning Systems
APP	Application
PDF	Portable Document Format
HTTP	Hypertext Transfer Protocol

## SUMÁRIO

<b>1</b>	<b>INTRUDUÇÃO .....</b>	<b>16</b>
1.1	MOTIVAÇÃO .....	16
1.2	JUSTIFICATIVA .....	17
1.3	OBJETIVO GERAL .....	20
1.4	OBJETIVOS ESPECÍFICOS .....	20
1.5	METODOLOGIA .....	21
1.6	CONTRIBUIÇÕES DO TRABALHO .....	22
1.7	ORGANIZAÇÃO DO TRABALBHO .....	23
<b>2</b>	<b>PRESTAÇÃO DE SERVIÇOS E O TERCEIRO SETOR DA ECONOMIA .....</b>	<b>25</b>
2.1	SETOR TERCIÁRIO .....	25
2.2	CRISE ECONÔMICA .....	26
2.3	DESOCUPAÇÃO DO BRASILEIRO .....	27
2.4	RELAÇÃO ENTRE PRESTADORES E INTERESSADOS .....	29
<b>3</b>	<b>O ESTADO-DA-ARTE DE SISTEMAS DE PRESTAÇÃO DE SERVIÇOS .....</b>	<b>31</b>
3.1	MISSION 7.....	32
3.2	GOOGLE MEU NEGÓCIO .....	33
3.3	GETNINJAS.....	35
3.4	HI JOB .....	37
3.5	CONSIDERAÇÕES FINAIS SOBRE AS FERRAMENTAS DESCRITAS .....	38
<b>4</b>	<b>PROPOSTA DA STARTUP ISERVICE .....</b>	<b>40</b>
4.1	CONCEPÇÃO .....	40
4.2	PLANEJAMENTO .....	41
4.3	PÚBLICO-ALVO .....	41
4.4	MODELO DE NEGÓCIO.....	42
4.5	CARACTERÍSTICAS .....	42
4.5.1	<i>Convergência de Necessidades .....</i>	<i>42</i>
4.5.2	<i>Sistema dinâmico.....</i>	<i>43</i>
4.6	DESAFIOS LEVANTADOS .....	43
4.6.1	<i>Como coletar Requisitos em Projetos Empreendedores – Startup.....</i>	<i>43</i>
4.6.2	<i>Como firmar Sistemas Multilaterais .....</i>	<i>44</i>
4.6.3	<i>Como evitar a fuga de transações de pagamento de serviço.....</i>	<i>45</i>
4.7	PESQUISA DE CAMPO .....	46
4.7.1	<i>Definição de Questionário .....</i>	<i>46</i>

4.7.2	<i>Resultados Obtidos</i> .....	47
4.7.3	<i>Recrutamento Obtido</i> .....	48
4.8	CONFIRMAÇÃO DO PROBLEMA PARA EXISTIR A STARTUP ISERVICE .....	49
<b>5</b>	<b>ESPECIFICAÇÃO DE REQUISITOS</b> .....	<b>50</b>
5.1	TÉCNICAS DE ESPECIFICAÇÃO EMPREGADAS .....	50
5.2	ESCOPO DO PROJETO .....	51
5.3	FUNCIONALIDADES DO SISTEMA.....	51
5.4	CASOS DE USO .....	60
5.5	REQUISITOS NÃO FUNCIONAIS.....	63
<b>6</b>	<b>MODELO DE ANÁLISE</b> .....	<b>65</b>
6.1	LEVANTAMENTO DE ENTIDADES E SEUS RELACIONAMENTOS.....	65
6.2	COMPORTAMENTO DO SISTEMA .....	72
<b>7</b>	<b>MODELO DE PROJETO</b> .....	<b>78</b>
7.1	PROJETO DE ARQUITETURA.....	79
7.2	PROJETO DE COMPONENTES .....	83
7.3	PROJETO DE CLASSES.....	85
7.4	PROJETO DE INTERFACES .....	89
7.5	PROJETO DE TESTES.....	94
<b>8</b>	<b>DESENVOLVIMENTO</b> .....	<b>97</b>
8.1	FERRAMENTAS E LINGAGUENS .....	97
8.1.1	<i>Javascript</i> .....	98
8.1.2	<i>Máquina Virtual NodeJS</i> .....	99
8.1.3	<i>Aplicativos móveis nativos e híbridos</i> .....	99
8.1.4	<i>Ionic Framework</i> .....	100
8.1.5	<i>Banco de Dados MongoDB</i> .....	101
8.2	IMPLEMENTAÇÃO DA API.....	101
8.2.1	<i>Modelos de dados</i> .....	102
8.2.2	<i>Controladores</i> .....	103
8.2.3	<i>Conexão WebSocket para respostas em Tempo Real</i> .....	105
8.3	IMPLEMENTAÇÃO DO APLICATIVO MÓVEL.....	106
8.4	TESTES DE SOFTWARE .....	116
8.4.1	<i>Testes Automatizados</i> .....	117
8.4.2	<i>Testando chamadas à API</i> .....	118
8.4.3	<i>Testando a Interface de Usuário</i> .....	119
8.5	CONSIDERAÇÕES SOBRE O DESENVOLVIMENTO DE SOFTWARE REALIZADO.....	119

<b>9</b>	<b>AVALIAÇÃO DE USABILIDADE .....</b>	<b>120</b>
9.1	OJETIVO .....	120
9.2	ESTRATÉGIA.....	121
9.3	RECRUTAMENTO DE USUÁRIOS .....	121
9.4	APLICAÇÃO DA AVALIAÇÃO .....	122
9.5	COLETA DE RESULTADOS DAS GRAVAÇÕES DE TELA .....	123
9.6	COLETA DE RESULTADOS DO QUESTIONÁRIO .....	126
9.7	CONCLUSÕES DA AVALIAÇÃO.....	131
<b>10</b>	<b>CONCLUSÕES .....</b>	<b>132</b>
	<b>REFERÊNCIAS .....</b>	<b>134</b>
	<b>APÊNDICE A – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS.....</b>	<b>137</b>
	<b>APÊNDICE B – DOCUMENTO DE ESPECIFICAÇÃO DE ANÁLISE.....</b>	<b>152</b>
	<b>APÊNDICE C – DOCUMENTO DE MODELO DE PROJETO .....</b>	<b>181</b>
	<b>APÊNDICE D – PESQUISA DE CAMPO PARA APERFEIÇOAMENTO DE REQUISITOS COM RESULTADOS</b>	<b>217</b>
	<b>APÊNDICE E – QUESTIONÁRIO PARA AVALIAÇÃO DE USABILIDADE .....</b>	<b>237</b>

## 1 INTRODUÇÃO

O trabalho apresentado trata da concepção, especificação e construção de uma solução em tecnologia para atender a um problema constatado no ramo de prestação de serviços.

O *software IService* – como se chama a solução – é uma aplicação móvel que possui a proposta de englobar a comunicação entre prestadores de serviço e clientes para contato, divulgação, contratação de quaisquer tipos de serviço. Pessoas físicas e jurídicas podem construir e divulgar seu perfil, informando quais serviços disponibiliza. Os interessados podem procurar quem os possa atender por palavras, proximidade por localização geográfica, disponibilidade de horário, reputação, etc. Quando um cliente adquirir o serviço, poderá dar uma nota para o mesmo, semelhante como é em sites de lojas virtuais, com produtos.

### 1.1 MOTIVAÇÃO

Quando precisamos que alguém ou uma empresa nos ajude com determinado tipo de tarefa, primeiramente temos que saber quem pode nos ajudar. Até poucos anos, os meios de busca eram principalmente perguntando às pessoas próximas, ou então procurando em uma lista telefônica. Atualmente, as formas de encontrar quem possa nos ajudar ficaram mais ágeis e práticas. Podemos, simplesmente, pesquisar na *web* quem realiza aquela atividade e, então, anotamos o contato.

Entretanto a praticidade para por aí. Se quisermos, por exemplo, saber quem conserta geladeira, perto de nossa casa, e que tenha orçamento grátis em domicílio, com a pesquisa genérica na *web* provavelmente teríamos um certo trabalho ou então nem conseguiríamos extrair esta informação em *sites* de busca. Talvez, precisaríamos recorrer aos métodos tradicionais - perguntando ou saindo para procurar pelas proximidades. Além disso, outros tipos de serviço, hoje, nem sequer possuem resultados na *web*, além de outros que, de igual forma, ainda possuem um diferencial de serem ambulantes - como mototaxista, ônibus, dentre outros.

Para completar, mesmo que com uma busca consigamos encontrar um prestador de serviço, que seja próximo de nós, ainda não poderíamos saber, por exemplo, se esse serviço é de qualidade - pelo menos não diretamente.

Logo, se existisse uma ferramenta que pudesse relacionar diretamente prestadores de serviço e pessoas interessadas, dando a possibilidade de buscar serviços por nome, categoria, reputação, localização, aproximação, preço, dentre outras características, facilitaria a vida tanto de pessoas que buscam serviços, tanto de prestadores de serviço que, aproveitando os recursos de maneira inteligente, poderiam aumentar suas vendas, dando mais visibilidade ao seu negócio.

Para finalizar, a aplicação poderia estar presente, até mesmo, em serviços públicos, como transporte público, o qual os empresários poderiam aderir à plataforma para melhorar o uso dos coletivos pelos usuários, e também para ter o controle da frota em tempo real, ou ainda taxistas que, mesmo havendo ferramentas próprias para sua modalidade de serviço, poderiam usar a plataforma *IService*, já que esta seria uma ferramenta genérica e globalizada para qualquer serviço.

## 1.2 JUSTIFICATIVA

Estamos em um momento de grandes transformações na maneira com que resolvemos problemas, buscamos informações e interagimos com o resto do mundo. Com as tecnologias atuais já conseguimos desempenhar atividades com mais rapidez e eficácia que não imaginaríamos, há alguns anos. E com este pensamento é que devemos sempre questionar e enxergar o que ainda podemos melhorar, na maneira de interagir e resolver problemas.

No mercado estão disponíveis algumas ferramentas para busca e contratação de serviços. Algumas destas são para serviços específicos, como carona privada, contratação de babás, e até mesmo de muitos tipos de serviço em uma só ferramenta. Entretanto, quase todas ferramentas não possuem recursos para contratação do serviço direto pelo aplicativo, outras são para serviços específicos, e mais as que são restritas à um local, uma região, ou essas características são combinadas em uma mesma ferramenta.

Desta forma, é importante analisar o que poderia ser entendido como uma limitação destas ferramentas. Pensou-se em estudar casos de insatisfação destas aplicações, percepções pessoais sobre como tudo em relação à prestação de serviços poderia funcionar.

Alguns questionamentos, anseios e perspectivas, foram levantados como justificativas para propor algo novo para o ramo de *e-business*. São eles:

- Por que não se pode saber o horário de atendimento em tempo real de prestadores de serviços e estabelecimentos?
- Com tantos aplicativos de redes sociais, em que é possível, trocar mensagens de texto em tempo real com uma pessoa, por que não podemos ter um canal direto de contato com um prestador de serviço? Assim como se tivéssemos nos deslocado até o local de atendimento, seria bom podermos nos comunicar com um atendente, o dono do serviço, de forma direta e fácil.
- Muitas pessoas sabem fazer alguma atividade econômica, mas ainda não divulgam. Ou, se divulgam, não possuem uma ferramenta que as possibilite atender diretamente pelo aplicativo – existem para poucos ramos de serviço, como o transporte. Um aplicativo que possibilitasse realizar divulgação e, também, contratação de praticamente qualquer tipo de serviço, poderia incentivar muitos setores econômicos para o mundo de negócios virtuais.
- As redes sociais são uma oportunidade para divulgar serviços, mas não são especializadas em serviços e contratação. Existem muitas condições e possibilidades sobre serviços que as redes sociais não estão aptas para atender, para resolver.
- Por não possibilitar recursos como atendimento virtual, muitos prestadores de serviço parecem não se interessar por ferramentas digitais para usarem como ferramenta de divulgação. Basta comparar uma rua, em um bairro de periferia, com o que há divulgado nos aplicativos que possibilitem divulgação.

- A localização geográfica poderia ser mais abordada nas ferramentas atuais. Por exemplo, sobre pessoas que trabalham de forma ambulante, se locomovendo durante o dia. Como identificar a localização destas pessoas? Como pode ser descoberto se, em um dado local, existem prestadores de serviço de um determinado tipo, para rápida contratação?

Ainda, além destes questionamentos apresentados, deve ser considerada uma importante questão: existem incontáveis pessoas que sabem realizar alguma atividade econômica, mas não divulgam. Se existem tantas pessoas com capacidade para exercer uma nova atividade, seja extra ou para lhe tirar de uma condição de desemprego, por que não se vê toda essa população em massa disponível nos aplicativos de serviço? Será que existe uma barreira entre essas pessoas e a tecnologia? Em tempos de crise – em que o país se encontra atualmente – pessoas que não estão empregadas poderiam prestar algum tipo de serviço por conta própria. Neste cenário, uma ferramenta que possibilitassem divulgar, contratar qualquer tipo de serviço, dando mais liberdade de expressão e propaganda, poderia ajudar bastante novos negócios. Poderia servir de incentivo para estas pessoas aproveitarem seus recursos e competências disponíveis.

A seguir, algumas situações exemplo de usos da aplicação. Dentre elas:

- Um usuário precisa de uma assistência técnica de Refrigerador, perto de sua residência. O usuário acessa o aplicativo e informa, no campo de busca por palavras, o termo “refrigerador”. A Aplicação realiza a busca e exhibe, em um mapa, os resultados de prestadores de serviço que satisfazem os critérios de pesquisa a partir da localização do usuário.
- O usuário precisa pegar um mototaxista assim que descer no ponto de ônibus. Ele abre o aplicativo e busca pela categoria mototaxista. O usuário visualiza em um mapa, assim como em aplicativos de táxi habituais, os mototaxistas cadastrados e sua localização em tempo real (os pontos que os identificam se

movem no mapa, conforme o movimento destes). O usuário solicita um mototaxista específico ou apenas solicita que um deles o encontre em um determinado local. O usuário também poderá informar o horário aproximado da corrida.

- Antes de sair do trabalho, o usuário abre o aplicativo e verifica onde estão os ônibus que ele pretende pegar. O usuário encontra-se na Avenida Nazaré e pelo mapa, percebe que o coletivo de interesse mais próximo está, ainda, no Entroncamento. Então o usuário aguarda, até que o coletivo esteja mais próximo, para poder se locomover ao ponto de ônibus.
- Um usuário que é técnico em refrigeração faz serviços em domicílio. Este possui um carro para se locomover até seus clientes. Sempre ele ativa sua localização em tempo real no aplicativo, para que quando estiver em determinada localidade atendendo um cliente, outras pessoas o vejam pela plataforma e se interessem pela praticidade de ter, naquele instante, um técnico em refrigeração por perto.
- Um entregador de gás ambulante usa o aplicativo com sua localização em tempo real. Assim, quem realmente precisa de gás poderá acionar uma entrega.
- Uma pizzaria usa o aplicativo para prover seus serviços de entrega em domicílio. O cliente solicita o pedido, informa o que deseja, realiza o pagamento pela plataforma assim que a pizza chegar à residência.

### 1.3 OBJETIVO GERAL

Projetar e implementar um sistema de informação para divulgação e contratação de serviços de diversos ramos em caráter experimental e de incerteza – uma *startup*.

### 1.4 OBJETIVOS ESPECÍFICOS

Os objetivos específicos, necessários para se chegar ao objetivo geral, são:

- Analisar o mercado de trabalho e econômico brasileiro para encontrar fundamentos para constatar a existência de oportunidades expiratórias de anseios de pessoas sobre prestação de serviços.
- Realizar uma pesquisa de campo, em forma de questionário, para ouvir o que as pessoas acham de uma ideia como a imaginada, a fim de validar o pensamento coletivo sobre a existência do problema levantado.
- Levantar funcionalidades essenciais para a solução dar certo.
- Especificar detalhadamente, aplicando conceitos de engenharia de software, como os requisitos podem ser enxergados no contexto de sistemas de informação computacionais.
- Desenvolver a ferramenta com um incremento oficial de implementação, a fim de atingir um mínimo produto viável (MVP) para ter, rápido e economicamente, um produto que posse ser palpável, testável e que sirva de demonstração para interessados em investir e fazer parcerias.
- Realizar um teste de usabilidade com usuários representantes do público alvo para a solução, a fim de validar questionas não funcionais, como a facilidade de uso da ferramenta desenvolvida.

## 1.5 METODOLOGIA

Durante a etapa de elaboração, uma pesquisa de campo será aplicada constatar a existência do problema, para atestar o compartilhamento da mesma visão do autor do projeto, e para auxiliar na coleta e qualidade dos requisitos.

Para a engenharia de software do projeto serão usadas normas e técnicas tradicionais da engenharia de software, com a construção formal de documentos como o de especificação de requisitos, de análise de sistemas e de modelo de projeto, utilizando metodologias como o padrão de diagramas UML

Para o desenvolvimento – implementação – serão usadas linguagens como *Javascript*, *Html*, *CSS*. Para persistência de Informações, um banco de dados orientado a documentos será utilizado. Além disto, ferramentas de desenvolvimento como *sublime text* e *Ionic* serão adotadas para auxiliar nesta etapa.

Por fim, para uma avaliação de usabilidade, são empregadas técnicas e conceitos da área de interação humano-computador(IHC), para uma efetiva avaliação de usabilidade do sistema desenvolvido. Esta avaliação é feita com o objetivo de aperfeiçoar a ferramenta, na medida em que o que se mostrar insatisfatório será ajustado depois da avaliação.

## 1.6 CONTRIBUIÇÕES DO TRABALHO

Com este trabalho, principalmente, será possível compreender um desenvolvimento de *software* para dispositivos móveis utilizando tecnologias e técnicas tradicionais, mas também incomuns e inovadoras.

Considerando a concepção do projeto, a ideia da ferramenta, e o problema que ela propõe resolver, espera-se contribuir para melhoria de comunicação entre as pessoas que querem prestar e consumir serviços de qualquer gênero, gerando praticidade à vida das pessoas, principalmente. Esta é a principal contribuição da ferramenta, quando, e caso, esta seja firmada no mercado de aplicativos móveis.

Na implementação do software, o uso de banco de dados não relacional, assim como o uso do protocolo de rede *websocket* são destaques para a concepção de novas vertentes sobre desenvolvedores, pois são ferramentas ainda pouco empregadas, quando comparadas com as tradicionais formas de atender às necessidades de persistência de dados e troca de mensagens entre os componentes de uma aplicação disponível na *web*.

É importante dizer que o trabalho apresentado não possui a premissa de criar, formular, novas teorias ou técnicas de desenvolvimento de software, ou qualquer que seja o aspecto na área de ciência da computação e teoria de sistemas. Entretanto, deve ser levado em consideração que toda atividade em uma área de conhecimento, por mais que não exponha uma teoria, hipótese,

pode contribuir expondo a prática de todo o processo adotado, com as técnicas utilizadas. Todo projeto de software, assim como este, possui diferenças e sutilizas. Estas que podem despertar nos leitores, que são da comunidade científica da área, novos aprendizados, argumentações, indagações, estas, por fim, eventualmente gerando novos pensamentos e considerações sobre como elaborar e desenvolver *software*.

## 1.7 ORGANIZAÇÃO DO TRABALHO

O trabalho é dividido em dez capítulos, contando com este, além de cinco apêndices – documentos de especificação de requisitos, análise de sistema, modelo de projeto de sistemas, além dos resultados da pesquisa de campo e da avaliação de usabilidade realizadas.

No capítulo seguinte (dois) algumas considerações sobre o assunto de prestação de serviços são dispostas. Questões como crise econômica e índices da desocupação do brasileiro são explicadas, além de um ponto de vista de como funciona a relação entre prestadores de serviço e consumidores.

No capítulo três são expostos alguns trabalhos, produtos, que se assemelham com o trabalho a ser apresentado, documentando o uso destas ferramentas e, no fim, realizada uma reflexão sobre os resultados e feita uma comparação com a proposta deste trabalho.

No capítulo quatro é dado o início à documentação, exposição, sobre a ideia deste trabalho, colocando a mesma sob o conceito de *startup*. O Público-alvo é definido e documentando que, para este, foi disponibilizado um questionário, objetivando equiparar o pensamento do idealizador do projeto com as expectativas destas pessoas.

O capítulo cinco é o inicial quanto ao processo de desenvolvimento do software, no qual são expostos os casos de uso, requisitos levantados para a aplicação ser desenvolvida. Já no capítulo seis, estes requisitos são aperfeiçoados com a modelagem de sistema, usando diagramas de classe, de sequência, dentre outros. Por seguinte, no capítulo sete o modelo de projeto do sistema é documentado, com o estilo de arquitetura escolhido, os componentes de aplicação levantados, as interfaces de usuário, os testes projetados.

Dado o trabalho de engenharia de software especificados nos capítulos anteriores, o processo de implementação é, então, trabalhado no capítulo oito. Este desenvolvimento gerou uma versão – incremento – da ferramenta suficiente para ser submetido a uma avaliação de usabilidade, a qual é relatada no capítulo nove.

Por fim, no capítulo dez, as considerações gerais sobre o trabalho apresentado são argumentadas.

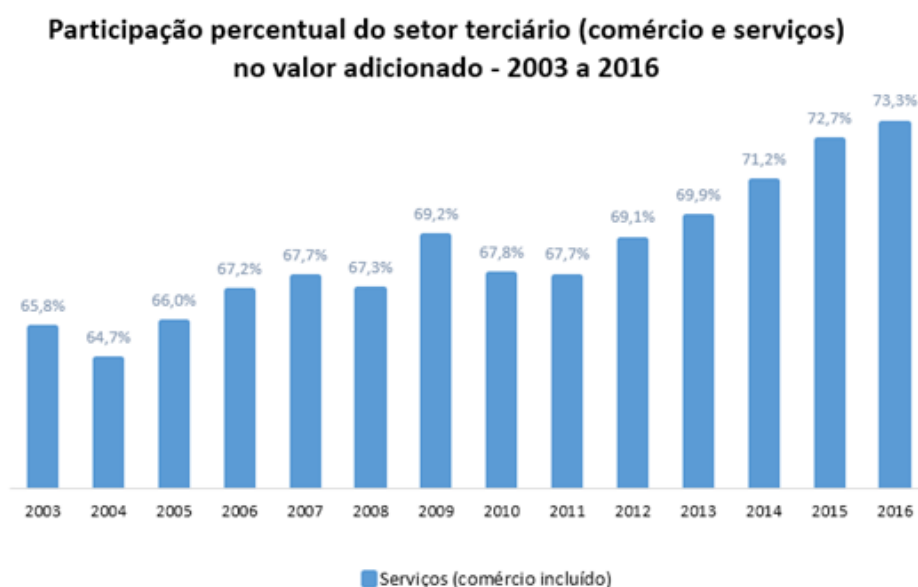
## 2 PRESTAÇÃO DE SERVIÇOS E O TERCEIRO SETOR DA ECONOMIA

Neste capítulo a prestação de serviços será discutida. Abordando o conceito, características, e pontos chave deste setor econômico, pretende-se expor o quão ainda ele pode ser explorado, principalmente no que se diz respeito ao mercado digital.

### 2.1 SETOR TERCIÁRIO

O setor de serviços vem crescendo gradativamente nos últimos anos. Apesar de momentos de crise econômica, este crescimento, mesmo com quedas momentâneas, continua a evoluir.

Esta área tem ganhado cada vez mais importância na economia do Brasil. O Ministério da Indústria (2017) relata que o setor de serviços tem apresentado relevante influência na economia brasileira e que, nos últimos anos, o crescimento do PIB vem sendo influenciado por este setor.



**Figura 1 – Gráfico de participação percentual do setor terciário no valor adicionado, entre 2003 e 2016 (IBGE, 2016)**

Os serviços estão situados no 3º setor da economia, ou setor terciário. Este setor, o maior da economia, engloba quaisquer tipos de serviços, desde a administração pública, passando por serviços como os de professores e advogados, até serviços domésticos. Como mencionado no Simbracs, Simpósio

brasileiro de políticas públicas para comércio e turismo (2013), os serviços avançam na construção de um país mais inovador e competitivo.

Assim, o setor mostra-se cada vez mais importante para a economia. Setor este, com sua infinidade de possibilidades de surgimento de novos tipos de serviços, que pode crescer ainda mais no decorrer dos próximos anos. Todo este potencial deve ser observado, de perto, por investidores e economistas. Este é um mercado em plena ascensão.

## 2.2 CRISE ECONÔMICA

No Brasil, a partir do ano de 2015, uma crise econômica se instalou no país. Este momento difícil afetou muitos setores da economia, no qual o terciário não ficou de fora.

Antônio Oliveira Santos, presidente da Confederação Nacional de Comércio de Bens, Serviços e Turismo (2017), menciona:

Até o ano de 2014, as vendas do comércio varejista ainda expandiram 2,2%, mas a partir de 2015 as vendas caíram, chegando a menos 4,3%. Em 2016, a queda foi de 6,2%, quando foram fechadas 108,7 mil lojas no Brasil.

É interessante observar esta realidade levando em consideração o comportamento da população frente à crise. Houveram ocorrências de pessoas procurarem renda alternativa ou complementar, com o intuito de amenizar as dificuldades enfrentadas pela alta dos preços, aumento do desemprego. No Jornal nacional (2017), publicou-se que:

“A crise econômica dos últimos anos fez milhões de brasileiros perderem o emprego, ou a condição financeira de manter o negócio que estavam conduzindo. E uma parte deles acabou mudando de atividade para entrar num caminho profissional completamente novo”

E, por Santos (2017), é dito que: “(...) vivemos hoje uma verdadeira revolução tecnológica, que afeta particularmente o comércio e a prestação de

serviços, setores que se apresentam em permanente transformação para se ajustarem à realidade”.

A crise econômica pode, portanto, assumir um lado positivo, ocasionando o surgimento de novos prestadores de serviço, tanto na formalidade, quanto na informalidade. Pode incentivar pessoas a buscarem novas fontes de renda, novas formas de trabalho, para que consigam equilibrar complementar seus orçamentos.

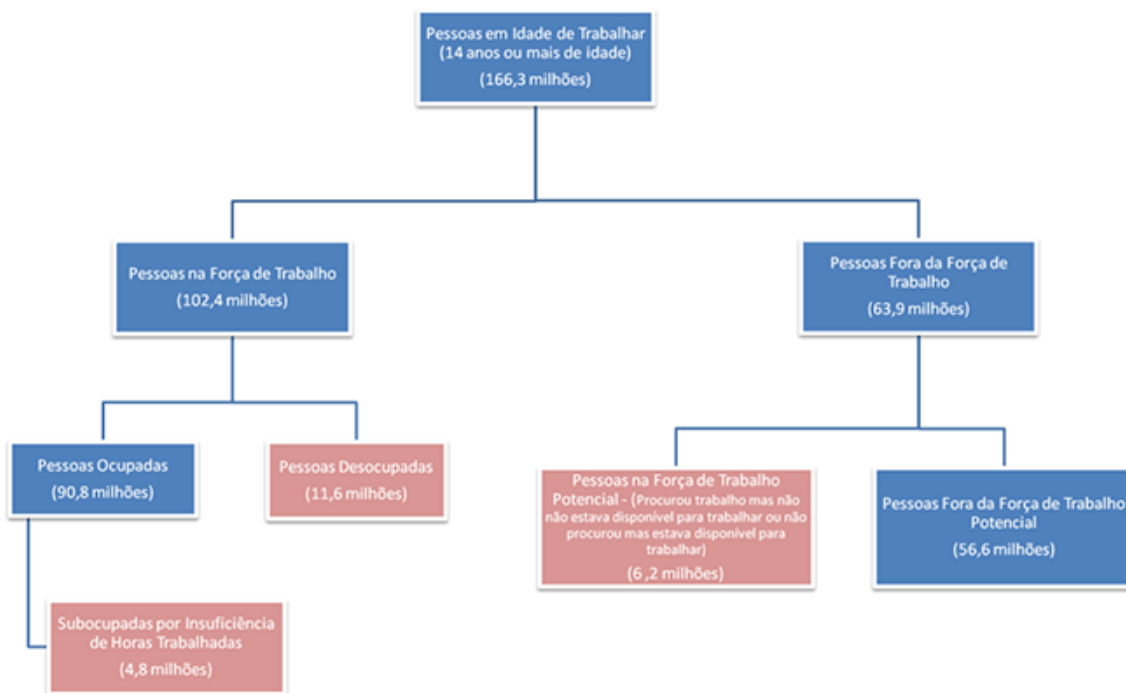
### 2.3 DESOCUPAÇÃO DO BRASILEIRO

Todo país possui seus índices de ocupação e desocupação da população. Trata-se de estudos que levam a resultados estatísticos os quais possam servir para analisar a situação da população quanto à ocupação de seu tempo, em atividades econômicas.

Segundo o IBGE (2016), os indicadores referentes a este assunto podem ser definidos como:

- A “taxa combinada da subocupação por insuficiência de horas e da desocupação” esta que representa ocorrências de pessoas que trabalham menos de 40 horas semanais, mas aceitariam trabalhar mais do que isso.
- A “taxa combinada da desocupação e da força de trabalho potencial”, a qual diz respeito às pessoas que gostariam de trabalhar, mas não procuram emprego
- E, por último, a “taxa composta da subutilização da força de trabalho” que sintetiza as duas anteriores.

A Figura 2 exhibe uma visão geral destes indicadores.



**Figura 2 – Taxas de Desocupação no Brasil (IBGE, 2016)**

Estes números mostram o que é notório no cotidiano. Sabe-se que muitas pessoas estão desempregadas ou não possuem aproveitamento de todo horário disponível para trabalhar. Por exemplo, algumas pessoas possuem emprego fixo de meio período e poderiam utilizar mais horas do dia com outra atividade de trabalho.

Associa-se a estes dados a ideia de que pessoas que necessitam de renda extra poderiam estar se ocupando com atividades remuneradas nas suas horas vagas. Esta ocupação poderia ser por meio de um novo emprego fixo adicional, ou com um negócio autônomo por conta própria, formal ou não.

Isso mostra que o brasileiro, tanto aquele atualmente empregado, como o que não possui emprego ou atividade remunerada, dispõe de tempo para realizar atividades as quais o ajudariam a ganhar renda, seja complementar ou principal. Muitas pessoas exercem uma profissão, mas acham que conseguem exercer outras atividades de trabalho, seja por escolha ou necessidade.

Pode ser fundamental a tecnologia ser aliada neste cenário, atuando como facilitadora, melhorando a visibilidade de negócios de diferentes segmentos, e incentivando indivíduos a ingressarem em novas áreas ou

expandirem seus negócios, com uso de ferramentas digitais (redes sociais, lojas virtuais, aplicativos).

## 2.4 RELAÇÃO ENTRE PRESTADORES E INTERESSADOS

A busca por serviços mudou com o passar do tempo. Antes da era digital, até mesmo do telefone, o meio de buscar serviços era bastante manual e rudimentar. Para que as pessoas conseguissem encontrar um serviço, precisavam se locomover, pedir informação para outras pessoas, ou recorrer a jornais publicitários.

O cenário mudou quando os meios de comunicação evoluíram no decorrer do século XX. Com a chegada do rádio, tv e telefone, a evolução da propaganda deu um salto significativo. Era possível, a partir dali, divulgar, para uma grande massa de público, serviços e produtos.

Atualmente tem-se a Internet como um dos principais meios de comunicação, com potencial enorme e que revolucionou a maneira de divulgar serviços. O marketing digital trouxe um mar de possibilidades para prestadores de serviços e empresas divulgarem suas marcas, produtos.

Entretanto, o que vemos, mesmo com tanta tecnologia disponível para buscar serviços, é, ainda, a dependência de meios convencionais para ter a informação que necessitamos sobre determinado estabelecimento ou pessoa. Alguns serviços são difíceis de serem encontrados na internet, redes sociais e outros meios de comunicação, outros sequer encontramos. Alguns tipos de serviço, dado sua natureza ou simplicidade, são mais comuns de ainda precisarmos busca-los com meios tradicionais. A exemplo:

- Mecânico de carros.
- Fruteiras, Tabernas, Mercarias de pequeno porte.
- Vendedores de bombons e doces (ou simplesmente “bombonzeiros”)
- Barracas de lanche
- Sapateiros
- Relojeiros
- Chaveiros

Existem muitos outros tipos de serviços que poderiam ser citados como exemplos neste cenário. Talvez a questão esteja relacionada com a simplicidade, a natureza do serviço, ou por sua distância com a informatização.

Além disso, existe o fator da mobilidade. Para serviços com natureza móvel, por mais que esses comerciantes e prestadores de serviço passassem a divulgar em redes sociais e sites de busca, não teria muito sentido pois, nestes serviços da *web*, o que se tem é a divulgação de um endereço fixo, geralmente.

No capítulo seguinte é abordado com mais detalhes o funcionamento das principais ferramentas de busca de serviço existentes, explanando as principais funcionalidades e restrições de cada uma, objetivando contextualizar o que há disponível para realizar atividades de prestação de serviço em aplicativos móveis.

### 3 O ESTADO-DA-ARTE DE SISTEMAS DE PRESTAÇÃO DE SERVIÇOS

Atualmente as opções de *software* para diversos tipos de dispositivos são as mais variadas. Quando um usuário precisa de um aplicativo que tenha uma função desejada, seja por lazer, para facilitar seu cotidiano, automatizar alguma tarefa, quase sempre acaba encontrando um ou vários aplicativos que satisfaçam suas necessidades. Lógico, existem exceções, mas esta busca quase sempre é bem-sucedida – encontram-se resultados quantitativos.

Entretanto, nem sempre o usuário consegue encontrar exatamente as funções que deseja. O que se vê, para muitos tipos de aplicações, é quantidade, mas nem sempre qualidade. Todavia, sempre há qualidade em alguns dos aplicativos de cada ramo, gênero. Geralmente os que possuem mais qualidade são os mais requisitados por usuários, com maior número de usos e, conseqüentemente, maior visibilidade.

Por Pressman e Maxim (2016) é afirmado que um produto útil certamente satisfaz às exigências explícitas dos envolvidos, assim como requisitos implícitos como usabilidade e segurança. Aplicativos, portanto, devem satisfazer às necessidades as quais os usuários definiram, resolver os problemas que eles tenham e, se possível, também tornar o uso da aplicação agradável.

Para o ramo de serviços também existem diversas opções disponíveis aos usuários. A disponibilidade está tanto em aplicações *web* como em móveis. Neste capítulo serão apresentados alguns trabalhos correlatados com este, principais aplicativos que estão disponíveis no mercado. O objetivo é expor os pontos positivos e negativos de cada um, baseando-se em comentários de usuários destes aplicativos na internet, e opinião do autor deste trabalho.

Para encontrar os trabalhos que serão exibidos aqui, foram utilizados métodos de busca na internet, com determinados termos, em determinados *sites*. O Quadro 1 exhibe as formas e canais de busca empregados.

Quadro 1 – mecanismos de busca por aplicações móveis de prestação de serviços.

<a href="https://www.google.com.br">https://www.google.com.br</a>	“tcc aplicativo serviço”, “trabalhos acadêmicos aplicativo serviço”, “aplicativos de serviço”, “projeto aplicativo serviço”, “trabalho acadêmico aplicativo prestadores de serviço”
<a href="http://g1.globo.com">http://g1.globo.com</a>	“estudante aplicativo serviço”, “aplicativo serviço”, “aplicativo reparo”, “aplicativo encanador”, “aplicativo pedreiro”.
<a href="http://www.tcc.sc.usp.br">http://www.tcc.sc.usp.br</a>	“Aplicativo serviço”, “aplicativo prestadores”, “prestadores de serviço”
<a href="http://www.periodicos.capes.gov.br">http://www.periodicos.capes.gov.br</a>	“aplicativo serviço”, “prestar serviço”

### 3.1 MISSION 7

Trata-se de um trabalho originado por jovens de Manaus, do BEPiD Fucapi (*Brazilian Education Program for IOS Development*). Os jovens propuseram um aplicativo móvel para unir prestadores de serviço de áreas específicas e pessoas que precisem destes serviços.

Segundo Xavier (2017), o aplicativo possui três categorias de serviços a serem divulgados:

- *Fixer* (voltada para limpeza e serviços domésticos)
- *Support* (direcionado para manutenção e conserto)
- *Runner* (com serviços de entrega de encomendas)

Ainda segundo Xavier (2017), outra funcionalidade do projeto é a de permitir a conversa entre prestador e consumidor, para conversarem e acordarem um valor pelo serviço. Apesar da negociação por conversa, os preços começaram com um valor mínimo, determinado para cada tipo de serviço.

Um aspecto interessante é que a pessoa que irá prestar o serviço é titulada como “herói” dentro da ferramenta. Pessoas (heróis) que tenham tempo livre, e saibam realizar uma atividade coberta pelo aplicativo, podem se disponibilizar para atender pessoas que necessitam daquele serviço. Este atendimento, apesar do nome dado aos prestadores, é remunerado.

Outra característica é a que os prestadores de serviço terão de se cadastrar junto aos responsáveis pela ferramenta, fornecendo documentos de identificação (RG, CPF) e outros como comprovante de residência e ficha de antecedentes criminais. Isto é bom por um lado, pois gera mais segurança aos futuros clientes de serviços, mas dificulta o interesse pelo aplicativo, desde o início, por parte dos prestadores.

O projeto encontra-se em fase de recrutamento de usuários que irão prestar serviços. Haverá a segunda etapa (versão) do aplicativo, a qual se destinará aos consumidores finais dos serviços que serão ofertados.

Um ponto questionável sobre este projeto é o fato de ser divulgado apenas a sua existência na plataforma *iOS*. O Público-alvo reduz drasticamente, uma vez que a maioria dos smartphones funcionam com o sistema Android. Além disso, pelo texto apresentado por Xavier (2017), o projeto aparenta ser inicialmente planejado para atuar no Estado do Amazonas, especificamente.

### 3.2 GOOGLE MEU NEGÓCIO

A Google tem como um de seus principais produtos a plataforma *maps*, a qual possui o mapeamento do mundo todo, exibindo, com detalhes, as ruas, formas, locais. É por meio desta plataforma que o Google Meu Negócio funciona.

A ferramenta de negócios da Google permite que pessoas físicas e jurídicas realizem um cadastro baseado em um endereço, como prestadoras de algum serviço ou proprietárias de um estabelecimento. O usuário informa, por meio da ferramenta *maps*, qual o endereço daquele negócio, além de outras informações. As principais funcionalidades da ferramenta, segundo o *site* oficial da ferramenta (2017), são:

- Uso gratuito para prestadores de serviço e consumidores.
- Permite personalizar a página de apresentação do negócio cadastrado com textos, imagens de fundo.
- Outras pessoas podem sugerir edições na página do serviço divulgado, que passará a valer mediante aprovação do responsável.
- Permite cadastrar um endereço como um estabelecimento comercial ou de prestação de serviço.
- Permite adicionar fotos para compor a apresentação do serviço.

Os consumidores podem utilizar as ferramentas de pesquisa da Google, bem como o serviço *maps* para buscar estabelecimentos, o qual sempre disponibiliza o mapa com os resultados mais próximos da localização do usuário. Com os resultados dispostos, o consumidor pode acionar um destes para que seja exibida a página de apresentação do serviço. Analisando a interface do aplicativo, foi observado que a página de um serviço exibe as seguintes informações:

- Nome.
- Distância entre o usuário e o local.
- Opção “Descreva este lugar”, onde qualquer pessoa pode informar dados categorizados sobre o local.
- Endereço.
- Horário de atendimento.
- Telefone.
- Website.
- Médias de Pontuação dada pelo público que visita a página.
- Resultados relacionados da web – onde ficam resultados de pesquisa do site *google* que se relacionam com a categoria do estabelecimento.

Este produto da Google é interessante, pois funciona de forma integrada entre diversos serviços da empresa, na *web*. A união entre o *site* e aplicativo de pesquisa Google, a plataforma *maps* e o aplicativo Meu Negócio

transforma a busca por serviços prática, divulgando informações de estabelecimentos comerciais e prestadores de serviço com base em endereços físicos. O contato entre o estabelecimento – ou prestador – e o consumidor é por meio de canais externos, como telefone e e-mail, ou pela locomoção do usuário até o local divulgado, cabendo à plataforma, somente cumprir o papel de informar a consumidores sobre estes serviços e estabelecimento.

### 3.3 GETNINJAS

*GetNinjas* é uma aplicação com o propósito de divulgar serviços que pessoas físicas e jurídicas possam realizar. Consumidores divulgam que precisam de um serviço, e prestadores aptos são notificados da ocorrência. A plataforma principal é móvel, mas também possui portal web para busca de serviços.

Ao abrir o aplicativo, é perguntado se o usuário é um prestador de serviço ou consumidor, o que reflete em uma mudança de contexto da interface e interação com usuário. O contexto de consumidor exhibe opções de categorias as quais o usuário pode selecionar para buscar serviços. São elas: Assistência técnica, Reformas, Eventos, Serviços Domésticos, Aulas, Moda e Beleza, Design e Tecnologia, Consultoria, Saúde, Autos.

Após a seleção de uma categoria, o usuário é perguntado sobre um aspecto relacionado àquela categoria, com o objetivo de direcioná-lo a uma subcategoria da anteriormente selecionada. Por seguinte outras perguntas são feitas ao usuário, como a forma que o serviço deve ser feito, a data prevista para realização, a frequência da necessidade, dentre outras. A seguir, o resultado do uso do aplicativo para buscar um serviço, com a conversa entre o usuário e o sistema ao selecionar a categoria “Serviços Domésticos”:

1. O usuário seleciona a categoria “Serviços Domésticos”.
2. O sistema exhibe uma lista de subcategorias.
3. O usuário seleciona a opção “Limpeza de Piscina”.
4. O sistema pergunta qual o tipo de serviço que o usuário necessita.
5. O usuário seleciona a opção “Limpeza de Filtro”.

6. O sistema pergunta qual o tipo de material da piscina.
7. O usuário seleciona a opção “Concreto”.
8. O sistema pergunta qual a quantidade de piscinas.
9. O usuário seleciona “2 a 3 piscinas”.
10. O sistema pergunta qual a frequência do serviço.
11. O usuário seleciona a opção “Uma vez por mês”.
12. O sistema pergunta qual o local do serviço.
13. O usuário seleciona a opção “Casa”.
14. O sistema pergunta para quando o usuário precisa do serviço.
15. O usuário seleciona a opção “Nos próximos 30 dias”.
16. O sistema pede para o usuário descrever informações adicionais por meio de um campo de texto livre.
17. O usuário opta por escrever ou não informações adicionais e prossegue.
18. O sistema exibe um mapa e pergunta ao usuário em qual bairro será o serviço.
19. O usuário seleciona o ponto no mapa e confirma.
20. O sistema pergunta o telefone do usuário.
21. O usuário informa o telefone e confirma.
22. O sistema exibe sucesso na busca pelo serviço e relata que prestadores serão alertados sobre a necessidade dele e que entrarão em contato.

Ao responder, na tela de entrada, que é prestador de serviço, o aplicativo muda o contexto e pergunta qual das áreas disponíveis o usuário trabalha. Em seguida são realizadas, também, algumas perguntas para saber mais sobre o serviço do usuário. A seguir, o registro desta interação:

1. O usuário seleciona a categoria “Aulas”.
2. O sistema pergunta quais serviços de aulas o usuário realiza.
3. O usuário seleciona as opções “Informática” e “Música”
4. O sistema pergunta quais dos serviços o usuário realiza como professor de música.

5. O usuário seleciona a opção “Bateria”.
6. O sistema pergunta onde o usuário realiza o serviço (presencial ou remoto)
7. O usuário seleciona a opção “Presencial”.
8. O sistema pergunta quais serviços o usuário realiza como Professor de Informática.
9. O usuário seleciona a opção “Programação”.
10. O sistema pergunta onde o usuário realiza o serviço.
11. O usuário marca ambas as opções “Presencial” e “Remoto”.
12. O sistema pede para o usuário dar um título e descrição textual para o serviço a ser publicado.
13. O sistema pede também dados de endereço, como CEP, logradouro, bairro, cidade

O e-mail recebido após cadastrar-se como prestador de um serviço na plataforma menciona: “O objetivo do GetNinjas é fazer com que você não precise se preocupar em correr atrás de clientes e possa se focar no que você faz melhor: o seu serviço! ”. Ainda, no e-mail, o fluxo de negociação entre prestadores e clientes é relatado como a seguir:

- O cliente procura profissionais no *GetNinjas*
- São enviados o contato e as informações do cliente para o usuário prestador e mais 4 outros prestadores do serviço requerido pelo cliente.
- O prestador deve entrar em contato com cliente para conseguir fechar o negócio.

### 3.4 HI JOB

*Hi Job* é um aplicativo móvel disponível para as plataformas *Android* e *iOS*. A ideia também se trata de conectar prestadores de serviço às pessoas que necessitam deles em qualquer momento.

Segundo o site do produto (Hi Job, 2017), a ferramenta é gratuita e está em funcionamento apenas na região metropolitana de Belo Horizonte, capital do estado de Minas Gerais. O fluxo da ferramenta, ainda no site do produto, é descrito da seguinte maneira:

1. A cliente posta a sua necessidade, indicando o tipo de serviço que ele precisa, o local do atendimento, a data e o horário sinaliza o quanto espera pagar por aquele serviço.
2. O prestador saberá de todas as demandas da sua categoria, dentro de um raio de distância que desejar, e poderá, caso tenha interesse, enviar uma proposta indicando sua disponibilidade.
3. O cliente analisa as propostas e fecha negócio.

### 3.5 CONSIDERAÇÕES FINAIS SOBRE AS FERRAMENTAS DESCRITAS

As ferramentas apresentadas foram encontradas por meio de pesquisas na internet, tanto em sites de busca como naqueles especializados em publicação de trabalhos acadêmicos. Houve dificuldade em encontrar trabalhos acadêmicos, com os termos empregados, que abordasse a proposta semelhante o suficiente para ser considerado um trabalho análogo ao proposto neste trabalho de conclusão de curso.

Por outro lado, a existência de ferramentas no mercado similares com a proposta deste trabalho foi constatada, mesmo nenhuma tendo exatamente as mesmas características, funcionalidades, fluxos e estratégias com o trabalho presente. Aspectos como os tipos de serviços disponíveis para publicação, interação entre prestador e consumidor e funcionalidades disponíveis são pontos que o autor deste trabalho identificou como os mais divergentes entre as ferramentas correlatas e a proposta a apresentar, neste artigo.

Acredita-se que alguns pontos poderiam ser trabalhados com outra visão de interação entre usuários de uma plataforma como esta, a exemplo da possibilidade de qualquer pessoa conversar com qualquer outra, dando mais liberdade à livre negociação, oportunidade de publicação de qualquer tipo de serviço, de qualquer área econômica (dentro da legalidade), maior importância e destaque à geolocalização e outras vertentes.

O que o autor, principalmente, almeja, em sua ideia para este ramo de aplicativos, uma maior interação entre o real e virtual para prestação de serviços, de tal forma que pudéssemos sentir o impacto disto em nosso

cotidiano. Que atitudes como saber se um estabelecimento encontra-se funcionando possa ser feito com apenas um chamado em um aplicativo, que seja possível ter maior visibilidade dos serviços ao redor, de todos os gêneros, e que a necessidade de contrariar, vender, comprar um produto ou serviço qualquer possa ser tão simples quanto ver as horas em um relógio, no pulso.

No próximo capítulo a proposta de solução em tecnologia para serviços defendida neste trabalho é explicada de forma mais abrangente, tratando a ideia como um startup de negócios digitais.

## 4 PROPOSTA DA STARTUP ISERVICE

Este capítulo trabalha a essência da ideia do projeto, explana o seu surgimento como uma simples ideia do autor e sua evolução com novas abordagens para um negócio digital potencial a ser explorado.

### 4.1 CONCEPÇÃO

*IService* nasceu a partir da seguinte indagação: “Por que não ter, em um ambiente único, informações sobre quaisquer gêneros de serviços, para tomar decisões sobre como e quando usufruir destes, onde quer que estejamos, na palma de nossas mãos? ”

Esse questionamento, por parte do autor, foi ainda mais profundo com o passar do tempo. Trouxe, consigo, pensamentos óbvios de como essa ideia poderia se expandir, como:

- Por que devemos ter um aplicativo para cada ramo de serviço existente? Temos aplicativos para transporte privado, outro para refeições, mais outro para conseguir um técnico. Quer dizer, de quantos aplicativos eu preciso para poder satisfazer todas as minhas necessidades sobre serviços?
- Temos aplicativos sobre serviços que só dão informações estáticas sobre estabelecimentos, como horário de atendimento. Não há ninguém, nesses aplicativos, que possa tirar uma dúvida sobre aquele lugar. É o que está ali e ponto.
- Por que não podemos, assim como em aplicativos de transporte privado, saber geograficamente o quão perto estamos de um prestador de serviço de qualquer tipo?
- Por que não podemos ter em um só lugar a possibilidade de encontrar serviços que precisamos, mas também até divulgar o que sabemos fazer, e que possa nos servir de renda extra?

Todas essas indagações, e mais algumas não citadas, levaram a ideia de criar uma aplicação a qual pudesse suprir necessidades quaisquer de prestação de serviços, não somente de quem já se titula como tal, mas para

quaisquer pessoas que tenham o interesse de divulgar um serviço, uma habilidade, técnica, para quem sabe, até, render um bom dinheiro.

## 4.2 PLANEJAMENTO

Com a ideia criada, procurou-se definir objetivos do projeto, em concordância com práticas de projetos de software, seguindo as etapas: de concepção, desenvolvimento, implementação, teste, implantação e avaliação de usabilidade. Assim, o que foi planejado foi o seguinte:

1. Realizar uma Pesquisa, em forma de questionário, para pessoas de diferentes perfis, para discutir e aprimorar a ideia.
2. Com base nas respostas da pesquisa, analisar pontos de melhoria no que se pretende desenvolver.
3. Desenvolver o Aplicativo por meio de técnicas de especificação e análise de requisitos, modelando e revisando, objetivando a fidelidade aos requisitos iniciais e, ao mesmo tempo, a melhoria destes.
4. Implementar o software (aplicativo), baseado em um modelo de projeto de software consistente, concretizando o que foi analisado e planejado, no produto final, em uma versão mínima (aplicando a ideia de mínimo produto viável).
5. Recrutar Usuários para participarem de uma Avaliação de Usabilidade do produto.

## 4.3 PÚBLICO-ALVO

Como a proposta ferramenta possui a premissa de unificar prestadores de serviços e clientes de todos os gêneros, o público-alvo acaba se tornando extenso e generalizado.

Este é um aspecto interessante, pois apresenta, consigo, um lado positivo e outro negativo. Positivo pois qualquer pessoa com um dispositivo móvel poderia ser atraída para utilizar a ferramenta. Negativo, pois não temos como focar em um tipo de usuário, o que transforma a etapa de coleta e análise de requisitos bastante desafiadora – temos de agradar diferentes perfis de

usuários, levando em consideração questões de usabilidade como facilidade de uso, acessibilidade, etc.

#### 4.4 MODELO DE NEGÓCIO

Primeiramente, a iniciativa apresentada pode ser considerada uma *startup*. “Uma startup é uma instituição humana projetada para criar novos produtos e serviços sob condições de extrema incerteza” (REIS, 2011, p. 26).

Assim como qualquer *startup*, um modelo de negócio deve ser definido, seguindo critérios, de acordo com o tipo de projeto e o que ele propõe.

Para esta ferramenta, devemos refletir que devem existir dois tipos de usuários: os que prestam serviço e os que contratam, consomem, serviços (O que também pode ter usuários que assume os dois papéis). Assim sendo, logo, conclui-se que para a ferramenta funcionar é necessária a existência dos dois tipos de usuários para, juntos, gerarem transações de atendimento.

Considerando este e outros aspectos, procurou-se identificar um modelo de negócio que se encaixasse no projeto. Um modelo de negócio, segundo Osterwalder (2010), é um esquema para a estratégia ser implementada por meio das estruturas organizacionais dos processos e sistemas. Este esquema pode ser naturalmente identificado devido a existência de características específicas do projeto, como a que foi relatada previamente.

Buscando informações sobre os diversos modelos de negócio existentes, o projeto pôde ser categorizado como um de Modelo de negócios Multilateral. Osterwalder (2010) explica que este modelo uni dois ou vários grupos interdependentes de consumidores, os quais são de valor para um dos grupos apenas se os outros grupos também estiverem presentes.

#### 4.5 CARACTERÍSTICAS

##### 4.5.1 Convergência de Necessidades

O aplicativo, como mencionado em sua concepção (item 5.1.1), reuniria em um só lugar, múltiplas necessidades das pessoas. Colocaria em uma interface única, por exemplo, a possibilidade de contratar um transporte privado - como carro, carona, taxis – para chegar a casa, e em seguida pedir uma

refeição de uma lanchonete por perto. Em outro momento, até, poderia divulgar sua disponibilidade de consertar computadores nos finais de semana, em sua residência.

Assim, temos em uma única ferramenta, as duas vias da prestação de serviços: quem presta serviço e quem adquire o serviço. Quem presta, também pode usufruir serviços, e vice-versa.

#### 4.5.2 Sistema dinâmico

A ideia é englobar várias áreas de serviços em um só local. Este aspecto traz consigo o desafio de atender, de igual forma, várias necessidades diferentes.

Para tanto, é necessário que trabalhem com abstrações e mecanismos capazes de contornar diferentes necessidades entre prestadores, necessidades específicas que possam existir em cada um.

Lógico, isto tudo só é possível pois temos, em um universo de centenas de tipos de serviços, várias necessidades comuns, as quais podem ser satisfeitas de igual forma a todos os prestadores e clientes.

Assim, precisa-se construir, e muito bem, aquelas propriedades que são comuns à grande maioria, fornecendo funcionalidades no software consistentes, potentes e, principalmente, analisar necessidades específicas de certos tipos de serviços, que, se desenvolvidas, podem atrair e atender uma determinada classe de prestadores de serviço. Estas funcionalidades dinâmicas serão melhor discutidas na seção 5.3 - Funcionalidades do Sistema.

## 4.6 DESAFIOS LEVANTADOS

A seguir, são discutidos os desafios identificados e percebidos durante a concepção e desenvolvimento do projeto.

### 4.6.1 Como coletar Requisitos em Projetos Empreendedores – *Startup*

Este questionamento diz respeito a como identificar requisitos consistentes com a realidade dos futuros clientes, sem que haja contato direto com eles.

Pelas técnicas de coletas de requisitos conhecidas em engenharia de software, foi adotado o questionário, por entender que neste tipo de projeto, no qual não há clientes e usuários específicos e identificados, o melhor é realizar uma série de perguntas, ordenadas e bem planejadas, para confrontar a ideia do projeto, o problema que se pretende resolver, com o pensamento de pessoas que poderão ser futuros usuários da aplicação – o público alvo.

Segundo Torres (2013), o mais importante, em um startup, é identificar se existe, de fato, o problema a qual a ideia se propõe a resolver. Neste sentido, o questionário vem para complementar a concepção da ideia, refinando tudo que ela pode abranger, e o que ela não deveria abranger, pontos fortes e fracos da solução que se propõe para o problema identificado. Na seção 4.7(Pesquisa de campos), a definição do questionário será melhor trabalhada.

#### 4.6.2 Como firmar Sistemas Multilaterais

Como afirmado na seção sobre modelo de negócios (4.4), esta é uma aplicação a qual segue a ideia do modelo de negócio Multilateral. Uma aplicação como esta precisa de um número considerável de usuários cadastrados. Isso porque o objetivo é realizar transações de atendimento (interação de prestação de serviço) entre pessoas.

Qual deverá ser a principal motivação para utilizar a ferramenta? Para consumidores buscar informações sobre prestadores de serviços e para prestadores aumentar sua visibilidade e lucratividade. E isto, obviamente, precisará destes prestadores de serviços cadastrados, assim como dos outros usuários com perfil de consumidores para usufruírem destes serviços.

Assim sendo, como começar uma aplicação a qual necessita de grande número de usuários e de perfis distintos?

Então, se começarmos a recrutar usuários de forma não balanceada entre os dois perfis – mais pessoas com perfil de prestar serviços do que pessoas com perfil de consumir serviços ou vice-versa, o insucesso pode ser decretado, afinal, como mencionado, sobre o modelo de negócios do projeto (seção 4.4), os grupos distintos de clientes (usuários) são interdependentes e um não gera valor

sem o outro. Portanto necessariamente devemos buscar um equilíbrio no recrutamento de usuários, conquistando ambos os perfis na mesma proporção.

Além dos dois perfis e seu equilíbrio necessário, outro aspecto é o de equilibrar pessoas com mesmo interesse. Isso quer dizer, de nada adianta termos a mesma proporção de prestadores e consumidores, se, por exemplo, temos, de início, vários prestadores de serviços domésticos e técnicos, mas o público de consumidores é jovem e procura locais de lazer, lanchonetes.

Esse é um aspecto desafiador de qualquer projeto que se enquadre neste modelo de negócio. Cabe aos responsáveis pelo desenvolvimento da ferramenta encontrar meios de recrutar usuários dos diferentes grupos – e que tenham usuários de mesmo interesse de todos os grupos – para que o negócio dê realmente certo.

#### 4.6.3 Como evitar a fuga de transações de pagamento de serviço

O faturamento da ferramenta será por meio de comissões sobre transações de pagamentos por serviços, bem como já foi dito. Entretanto, isso cria um problema. A ferramenta possibilitará que pessoas se comuniquem por mensagens. Se um prestador de serviço não quisesse pagar pela comissão, poderia não efetuar a transação de atendimento pela plataforma, e realiza-la “por fora”, usando o aplicativo somente para marcar o encontro com o cliente.

Esse é um problema importante a ser superado, pois sem um mecanismo para barrar isto, anularia o faturamento do projeto, o que colocaria em risco todo o esforço de desenvolver e investir na ferramenta.

Uma saída, então, pensada foi a de bonificar os prestadores que utilizam realizarem a transação pela plataforma. Algumas formas de bonificar poderiam ser:

- Quando o prestador alavancasse certo número de atendimentos realizados, ganharia uma divulgação gratuita de seu produto para uma determinada massa de interessados.

- *Ranking* dos 10 prestadores de serviços mais bem avaliados pelos clientes em atendimentos. Este *ranking* serviria para destacar os prestadores em buscas pela plataforma.
- Determinado número de atendimentos realizados, gerar um pequeno crédito para ganhar descontos em serviços de outras pessoas que o prestador, porventura, quisesse adquirir.

## 4.7 PESQUISA DE CAMPO

Para auxiliar no aprimoramento da ideia e o problema o qual a ferramenta se propõe a solucionar, uma pesquisa de campo foi realizada, resultados foram coletados e analisados para obtenção da constatação da existência de interesses de usuários e melhoria nos detalhes elaborados, de início, pelo autor.

### 4.7.1 Definição de Questionário

Um questionário foi elaborado, com perguntas simples, instigando pessoas a refletirem sobre algumas situações as quais a ferramenta propõe resolver.

As perguntas iniciais identificam a pessoa com seu nome, idade, e-mail e possíveis áreas de interesses sobre serviços. Estes dados podem servir no futuro para contato e convite para utilizar a plataforma.

Adiante, foram colocadas perguntas com valores pré-definidos para serem respondidos. A estratégia seguida, aqui, com perguntas de valores discretos, objetiva gerar, facilmente, dados estatísticos para a análise quantitativa dos resultados.

Entretanto, em meio as alternativas de cada questão, em algumas foi disponibilizado um item descritivo, rotulado como “Outros”, para que os entrevistados, se não encontrassem uma alternativa coerente com sua expectativa, pudessem informar uma resposta mais assertiva. Este questionário encontra-se integralmente no Apêndice D deste trabalho.

Se existisse um aplicativo para divulgação de serviços, seria interessante se...

- Pudéssemos saber o horário de atendimento de serviços e estabelecimentos
- Que houvesse sempre uma sinalização se o local ou a pessoa que presta serviço está realmente atendendo no momento que consultássemos
- Que fosse possível, pra alguns serviços, poder agendar um atendimento futuro
- Que pudéssemos realizar pagamento por um serviço diretamente no aplicativo
- Que fôssemos avisados de promoções sobre área de serviços ou estabelecimentos específicos , próximo de nós.
- Que fosse possível, eu, em meu nome, divulgar minhas habilidades e serviços que eu possa realizar, pra ter uma renda extra.
- Que fosse possível dar uma nota depois de ser atendido por uma pessoa ou estabelecimento
- Que fosse possível ver a localização móvel de prestadores e até pessoas comuns que costumam atender em Domício
- Outro: \_\_\_\_\_

**Figura 3 – Questionário de Pesquisa – Pergunta com opção descritiva**

#### 4.7.2 Resultados Obtidos

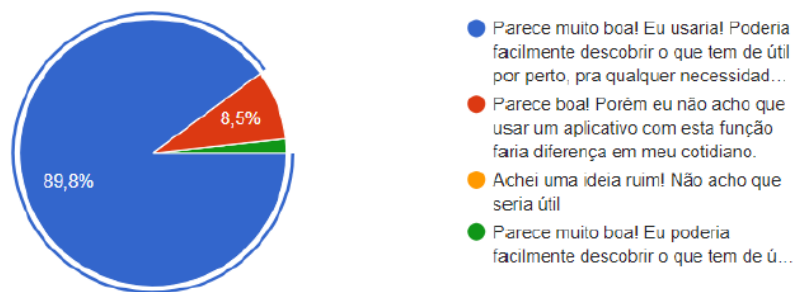
Com a aplicação do questionário, foi possível interpretar, dos resultados, a constatação da existência do problema o qual a proposta da startup *iservice* se propõe a resolver. Além disso, foi possível mensurar quais das seguintes ideias de funcionalidades do autor eram mais concordadas de serem úteis, também pelos participantes.

A Figura 4 exhibe o resultado para uma das perguntas que ajudava a entender se, de fato, o problema que originou a startup existia na visão dos entrevistados.

Imagine que existisse um aplicativo o qual você pudesse ter a visibilidade em um mapa de todos os tipos de serviço perto de você, de ambulantes, camelôs, barracas de lanche, pessoas comuns que saibam fazer algum serviço, até grandes estabelecimentos. Com um clique você poderia saber sobre as informações de qualquer um desses estabelecimentos ou pessoas, poderia se comunicar com elas por mensagens, contratar pelo aplicativo com cartão de crédito, avaliar os serviços. O que acha da experiência de um aplicativo com esta função na sua vida? \*

- Parece muito boa! Eu usaria! Poderia facilmente descobrir o que tem de útil por perto, pra qualquer necessidade que eu tenha de serviços
- Parece boa! Porém eu não acho que usar um aplicativo com esta função faria diferença em meu cotidiano.
- Achei uma ideia ruim! Não acho que seria útil

59 respostas



**Figura 4 – resultados para uma pergunta pertinente da pesquisa de campo sobre a existência do problema que a startup iservice pretende resolver**

#### 4.7.3 Recrutamento Obtido

Com o questionário aplicado, foi possível coletar dados de contato de todos os participantes. Os usuários que responderam positivamente a determinada pergunta, que pedia a confirmação do usuário se poderia ser convidado no futuro para se cadastrar no aplicativo, criaram a possibilidade de serem recrutados, em um momento futuro, após a aplicação do questionário. Este recrutamento não foi baseado só nesta resposta. A viabilidade de recrutar usuários está inserida nas perguntas do formulário. Uma vez que os participantes autorizem, poderá ser possível, com as respostas do questionário, realizar um pre-cadastro na plataforma, quando ela entrar no ar.

## Contato Futuro

Você aceitaria receber um e-mail, futuramente, quando este aplicativo for lançado, para se cadastrar e experimentar? \*

Sim

Não

Figura 5 – Pergunta se a pessoa concorda em receber e-mail futuro para contato

### 4.8 CONFIRMAÇÃO DO PROBLEMA PARA EXISTIR A STARTUP ISERVICE

Com a definição dada para a ideia da ferramenta *iservice*, o problema que esta estava proposita a resolver, e coletando os resultados da pesquisa de campo, foi possível constatar que o problema levantado realmente existe, sobre a visão da grande maioria dos participantes da pesquisa, estabelecendo, portanto, a viabilidade de existir a ideia como uma startup, propriamente dita.

Portanto, o início da construção da ferramenta, poderá, agora, fazer sentido. No capítulo seguinte, esta construção passará a ser abordada, iniciando-a com uma especificação de requisitos, que foram definidos partindo das ideias do autor da iniciativa em apoio com os resultados da pesquisa de campo realizada.

## 5 ESPECIFICAÇÃO DE REQUISITOS

Neste capítulo são apresentados o escopo do projeto e as funcionalidades esperadas, as técnicas empregadas para coletar, especificar e modelar requisito. São expostos fragmentos do documento de especificação de requisitos da aplicação e realizada uma autoanálise de cada requisito funcional e não funcional.

### 5.1 TÉCNICAS DE ESPECIFICAÇÃO EMPREGADAS

A construção da ferramenta apresentada neste trabalho, como já mencionado em capítulos anteriores, trata-se de uma ideia do autor, a qual surgiu com sua própria necessidade de buscar serviços e reflexão sobre o cenário. E por meio de questionário público, foi coletada a opinião de outras pessoas para encontrar concordância com seu pensamento da necessidade de existir uma ferramenta como esta.

Como todo projeto de software, é necessário realizar a coleta de requisitos das pessoas e partes interessadas, também chamados de *stakeholders*. Entretanto neste projeto o responsável por fazer o software e o autor da ideia são a mesma pessoa. Este aspecto é bastante comum em projetos de empreendedores da informática – ou *startups*. Dado uma situação como esta, é de imaginar que o responsável pela concepção e construção da ferramenta deva procurar um equilíbrio próprio entre suas opiniões e decisões sobre o projeto para definir requisitos e projetar o sistema.

Este papel duplo traz ao projeto uma dificuldade a mais para identificar pontos críticos de especificação, por um lado. Em que ponto termina o pensamento do autor como papel de *stakeholders*, e começa o de analista e desenvolvedor do software? Este é um desafio, pois foge dos padrões de coleta e análise de requisitos de projetos habituais, em que normalmente partes interessadas e analistas de sistemas são pessoas diferentes, de áreas e interesses distintos.

Entretanto, esse desafio foi amenizado quando se considerou aplicar um questionário de opinião pública sobre o atual cenário de prestação de

serviços por meio digital e a hipótese de existência de um aplicativo com as funcionalidades e objetivos que o autor pretendia desenvolver. Portanto, o questionário apresentado na seção 4.7 é produto da técnica de coleta de requisitos empregado para este projeto. Este questionário possibilitou afirmar ao autor quais requisitos previamente definidos teriam relevância a potenciais usuários futuros da aplicação.

É sabido que existem diversas técnicas de coleta de requisitos de *software*. “Especificação pode ser um documento por escrito, um conjunto de modelos gráficos, um modelo matemático formal, um conjunto de cenários de uso, um protótipo ou qualquer combinação dos fatores citados” (PRESSMAN e MAXIM, 2016, p. 135).

## 5.2 ESCOPO DO PROJETO

O escopo do projeto foi definido como uma aplicação móvel para divulgação e prestação de serviços de múltiplos gêneros, interligando em um só ambiente prestadores de serviço e consumidores de todas as áreas, tornando possível a comunicação entre eles. Deverá ser possível pessoas físicas e jurídicas se cadastrarem, buscarem informações sobre serviços de seus interesses, com informações de localização por um mapa com *gps*, preços e condições de cada serviço. Qualquer pessoa poderá assumir papel de consumidor e/ou prestador de serviços. Opcionalmente a contratação do serviço poderá ser diretamente pela plataforma, para o momento presente ou futuro agendado.

## 5.3 FUNCIONALIDADES DO SISTEMA

As funcionalidades levantadas para o projeto estão citadas a seguir, uma a uma, descrendo-as e realizando uma análise e relevância aos objetivos do software a ser desenvolvido.

### **Criação de Perfil de Usuário**

Como ponto de partida da aplicação, é esperado que pessoas físicas e jurídicas possam se cadastrar na ferramenta, informando seus dados pessoais, interesses por serviços dentre outros. A pessoa cadastrada possuirá sua página de perfil a ser visualizada por outras pessoas cadastradas.

O cadastro de usuário poderá ser realizado por meio de um formulário para ser preenchido com dados pessoais, ou por meio de vínculo à rede social *facebook*. Esta última forma visa facilitar o ingresso de usuários, com praticidade para a maioria do público alvo, os quais normalmente possuem conta na rede social.

Todos os usuários cadastrados terão sua página de perfil publicada. Esta interface normalmente será acessada por outros usuários quando o dono do perfil for responsável por um serviço ou unidade de atendimento de serviço, e também quando prestador do serviço quiser ver detalhes sobre seu cliente.

### **Divulgação de serviços do usuário**

Usuários poderão divulgar serviços que queiram prestar. O tipo de serviço a ser divulgado será bastante variado. No momento do cadastro, o usuário deverá informar se o serviço está habilitado para ser contratado pela plataforma, o que exige pagamento com cartão de crédito.

O registro de serviço deve possuir, ainda, uma descrição de texto livre, faixa de preço e parâmetros como o que indicará que, caso o serviço esteja habilitado para contratação na plataforma, exigirá ou não encontro com o cliente, por meio de um mapa com rastreamento de localização do prestador e consumidor.

O usuário poderá manter mais de um serviço publicado. Não há restrição de número de serviços.

A ideia é que, não somente pessoas que já trabalham prestando serviços façam o cadastro de seus serviços na plataforma, mas também de qualquer pessoa que se sinta motivada a explorar suas habilidades e competências para exercer uma atividade econômica. Muitas pessoas podem aumentar sua renda e até descobrir novos *hobbies* e realizações profissionais com este recurso. Estará aberto a todos, sem burocracias.

### **Divulgação de unidades de atendimento de serviços**

A divulgação de serviços automaticamente inclui a divulgação de uma unidade de atendimento inicial. Deve ser possível, entretanto, a manutenção da unidade e inclusão de unidades extras para um serviço.

A unidade é composta de horário de atendimento, responsável pela unidade (que pode ser o dono do serviço ou uma terceira pessoa escolhida pelo dono), estado, e uma possível localização fixa, caso o usuário opte por unidade fixa, ao invés de móvel.

### **Pesquisa de serviços**

Os serviços poderão ser buscados por palavras-chave, por interação com um mapa ou por filtros avançados, como por faixa de preços, horários de atendimento, condições de pagamento e por reputação.

Este requisito trata do principal aspecto da aplicação: buscar serviços por localização. O usuário consumidor que buscará serviços, normalmente se interessará em buscar informações sobre serviços ao seu redor. Pessoas querem saber qual a melhor opção de almoço por perto, querem saber se um estabelecimento que já conhecem se encontra funcionando no momento, dentre outras diversas situações. Torna-se interessante, ainda mais, pois à medida que o usuário se locomover, o aplicativo naturalmente poderá exibir sua localização, e exibindo serviços ao seu redor, atualizando o contexto de serviços por perto.

Assim, o mapa contendo pontos que mostram serviços por perto (com endereços móveis ou fixos) é um poderoso mecanismo de encontro entre consumidores e prestadores de serviço.

### **Contratação de serviços pela plataforma**

Deverá ser possível contratar o serviço diretamente pela plataforma, para pagamento do serviço prestado com cartão de crédito. O prestador que julgar conveniente oferecer esta opção aos demais usuários da aplicação, poderá receber pedidos de atendimento de clientes interessados em seu serviço.

O usuário consumidor, no perfil do serviço, solicita atendimento. O usuário prestador do serviço, responsável pela unidade de atendimento do serviço, é notificado. Este decide atender ou não o cliente. Em caso de

aprovação do atendimento, o sistema exibe os valores a serem faturados pela prestação do serviço. O pagamento é realizado por meio de cartão de crédito, direto pela plataforma.

No cadastro do serviço, deverá haver uma parametrização que indicará que o aplicativo, no ato da solicitação de atendimento, controlará o encontro das partes – prestador e consumidor. Isto é útil, por exemplo, para serviços como de transporte particular, guincho de automóveis, trabalhadores que atendem em domicílio, dentre outros.

Ao contratar o serviço, o sistema exibirá perguntas ao consumidor para personalizar atividades da ferramenta em relação ao atendimento e pós-atendimento. Campos que terão ações específicas dentro da plataforma. Estes são descritos no Quadro 2.

**Quadro 2 – Definições de campos especiais de formulário pós-atendimento.**

Campo especial	Função
“Ser lembrado periodicamente de renovar o serviço”, discreto, opções: “Nunca”, “Diariamente”, “Semanalmente”, “Mensalmente”, “A cada 6 meses”.	O usuário consumidor do serviço receberá uma notificação periodicamente do aplicativo.
“Cpf na nota ?”, caixa de marcação.	Ao término do atendimento, a nota emitida inclui o cpf do consumidor
“O prestador de serviço poderá entrar em contato com você mesmo após conclusão do atendimento”, caixa de marcação.	Permitirá que o prestador de serviço possa enviar mensagens de texto ao consumidor em qualquer tempo após o termino do atendimento.
“Atendimento anônimo”, caixa de marcação.	A identificação do consumidor no registro histórico de atendimento não será armazenada. Somente prestador de serviço irá ser identificado na transação. O consumidor poderá

	apagar de sua lista de histórico de atendimentos o registro.
--	--

### **Divulgação de agenda de atendimentos**

Deve ser possível configurar uma agenda de atendimentos do serviço a ser divulgado. Com esta agenda configurada, tanto o prestador do serviço, como os consumidores em geral poderão agendar atendimentos – ou somente o próprio prestador, se assim preferir. No registro do atendimento são informados dia e hora para realização do atendimento.

Este requisito se encaixa nas necessidades, principalmente, de serviços que normalmente só ocorrem mediante agendamento. Como exemplo há aluguel de locais, salas, equipamentos, consultórios médicos, odontológicos, e muitos outros.

### **Consulta de atendimentos realizados**

O usuário deve poder consultar registros históricos de atendimentos realizados. Pelo prestador de serviço, pode consultar todos os rendimentos agendados e já realizados para seu serviço. Pelo consumidor, pode consultar todas as ocorrências em que foi atendido por qualquer serviço da plataforma, além de seus atualmente agendados. Os dados de atendimentos realizados são: serviço, unidade, data e hora, valor total pago, valor total da negociação e status (Agendado, Realizado, Cancelado).

### **Gerência de caixa**

Deve ser possível o usuário informar valores de entrada de atendimentos realizados para cada serviço que divulgar na plataforma. O usuário poderá usar o caixa de modo manual (informando valores) ou automática, quando os atendimentos são realizados na plataforma.

O usuário poderá, periodicamente, realizar fechamento de caixa, informando valor restante, data e hora e, claro, abertura de caixa, também com as datas e um valor inicial.

Aqui um ponto interessante é a integração com registro de atendimento. Cada lançamento no caixa pode estar associado a um atendimento realizado, tornando a recuperação da informação mais prática, quando consultado pelos registros de caixa, podendo facilmente descobrir que atendimento gerou cada valor.

### **Emissão de notas fiscais**

O sistema deve dispor de opção para gerar cupons fiscais de atendimentos realizados. Este recibo pode ser *customizado* pelo prestador do serviço, atribuindo ao documento ajustes textuais. Esta emissão é somente por via eletrônica, podendo ser gerado um pdf para exportação deste. Além disso, as notas fiscais geradas podem ser consultadas a qualquer momento por ambas as partes envolvidas: prestador do serviço e consumidor beneficiário do atendimento.

É importante ressaltar que esta funcionalidade não visa atender qualquer vínculo com a Receita Federal para geração de nota fiscal oficial, seguindo padrões do órgão. Servirá apenas de comprovante de realização do serviço realizado pela plataforma para aumentar a sensação de segurança dos consumidores, possibilitando que estes exportem a informação de que foram atendidos por determinado prestador, em determinada data, hora e com o valor total pago.

### **Configuração de clientes personalizados**

Deve ser possível, para prestadores de serviços, configurar um formulário de atendimento e/ou formulário de cadastro de clientes. O prestador poderá configurar que informações, de seu interesse, poderiam ser relevantes para coletar de seus clientes em dois momentos:

- Ao solicitar atendimento pela primeira vez, é exibido ao consumidor o preenchimento de um formulário de cadastro para aquele serviço.
- Sempre que solicitar atendimento para o serviço, o cliente preenche um formulário com informações relevantes ao prestador para complementar cada atendimento.

Os formulários serão modelados com a definição de um conjunto de campos, estes definidos como mostra o quadro 3.

**Quadro 3 – Dados de cada campo de formulários de cadastro e atendimento.**

<b>Nome do dado</b>	<b>Funcionamento</b>
Rótulo	Nome do campo que será percebido pelo usuário consumidor que fará uso do formulário.
Tipo de campo	Formato de dados que o campo permitirá a entrada do usuário consumidor. Os tipos serão: texto, número, discreto(opções definidas), data, valor/fração, caixa de marcação.
Obrigatoriedade	Se o campo deverá ser preenchido indispensavelmente pelo usuário de formulário.
Opções	Em caso de campos tipo “discreto”, será informado uma lista de valores fixos para que o usuário de formulário escolha uma, dentre as disponíveis.

O formulário de cadastro servirá para o propósito de coletar informações de clientes que, dado a natureza do tipo de serviço, podem solicitar novos atendimentos periodicamente, e que estas informações não mudam por atendimento. Exemplos de situações que se encaixam nesta necessidade, bem como possíveis confecções de formulários de cadastros são expostas no quadro 4.

**Quadro 4 – Exemplos de confecção de formulário de cadastro.**

<b>Serviço</b>	<b>Campos do formulário</b>
----------------	-----------------------------

	<b>Nome do campo</b>	<b>Tipo e definição do campo</b>
Moto taxista	<i>Endereço residencial</i>	Texto
	<i>Frequência de uso</i>	discreto, opções: <i>dias de semana, finais de semana, raramente.</i>
Professor de Instrumentos de corda	<i>Curso de interesse,</i>	discreto, opções: <i>violão, guitarra, contrabaixo.</i>
	<i>Idade</i>	número
Revendedora de cosméticos	<i>Telefone</i>	número
	“Telefone”	Número
	“marca preferida de cosméticos”	Discreto, opções: <i>marcaA, marcaB, marcaC</i>

Já o formulário de atendimento poderá ser útil para coletar informações que normalmente são diferentes para cada novo atendimento, dependendo do cliente, da situação e outros fatores. O quadro 5 exemplifica cenários de serviços que poderiam ser úteis aplicar este tipo de formulário.

**Quadro 5 – Exemplos de confecção de formulário de atendimento.**

<b>Serviço</b>	<b>Campos do formulário</b>	
	<b>Nome do campo</b>	<b>Tipo e definição do campo</b>

Aluguel de salas de reuniões e conferências	Deseja servir <i>coffe break</i>	caixa de marcação
	<i>Adicionar equipamento de Datashow</i>	caixa de marcação
Técnico em Refrigeração	<i>Marca da central</i>	discreto, opções: <i>Limpeza, Reparo, Instalação</i>
Consultório odontológico	Especialidade	discreto, opções: <i>Endontia, Cirurgia, Odontopediatria.</i>
	<i>Faixa Etária</i>	discreto, opções: <i>Criança, Adolescente, Adulto, Idoso.</i>
	<i>Convênio</i>	discreto, opções: <i>Particular1, Particular2, SUS.</i>
Pizzaria	<i>Sabor</i>	discreto, opções: <i>Mussarela, Portuguesa, Marguerita, Quatro Queijos.</i>

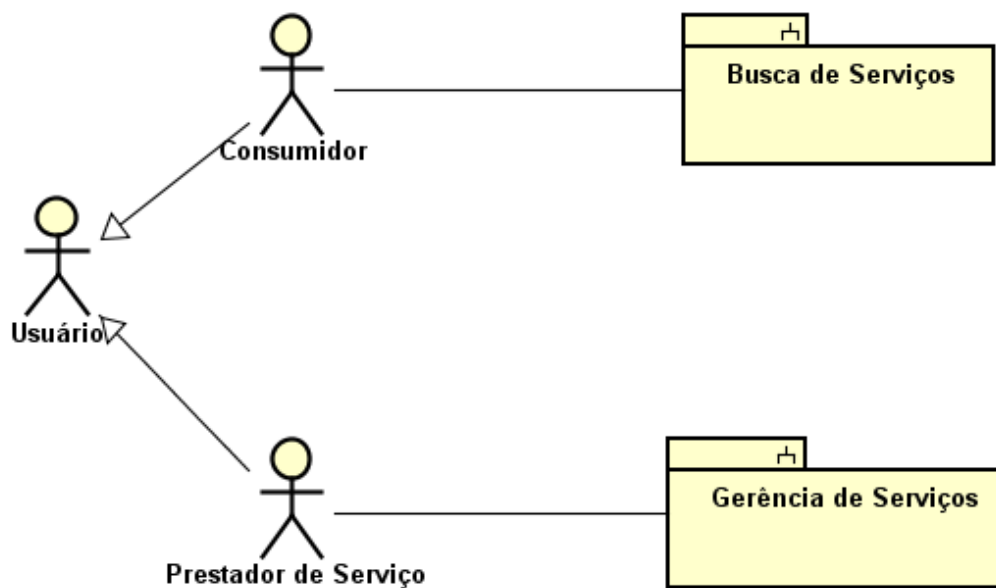
O formulário de atendimento, especificamente, poderá conter o funcionamento de “menu”, ou também conhecido como “carrinho de compras”. Ou seja, o usuário consumidor irá preencher o formulário, e as escolhas irão se acumulando e, no fim, o usuário poderá finalizar o pedido. O sistema exibirá o valor a ser pago. Neste caso, cada opção do formulário deverá ter um valor monetário associado. E, ainda, o usuário poderá informar a quantidade de cada item escolhido no formulário.

Vale ressaltar que os formulários serão construídos para cada serviço, de cada usuário na plataforma. O usuário poderá, com total liberdade, modelar uma interface feita sob medida, capaz de coletar dados de clientes com fidelidade às necessidades do serviço e do usuário responsável por ele. Entretanto, um facilitador existirá para melhorar o entendimento e aprendizado do recurso: modelos de formulário. O que os quadros 4 e 5 mostraram, são exemplos de modelos de formulários para cada tipo de serviço que possa existir na plataforma. Assim, quando o prestador de serviço for modelar seu formulário e, a exemplo, seu serviço é “Consultório odontológico”, um modelo predefinido de formulário para seu negócio será disponibilizado como exemplo. Se o usuário quiser ajustar algo do modelo ou começar com um formulário em branco, terá liberdade para tais ações.

#### 5.4 CASOS DE USO

O sistema é subdividido em dois subsistemas: Busca de serviços e Prestação de serviços. Em cada um dos subsistemas há um ator correspondente: consumidor e prestador de serviço.

Importante esclarecer que o usuário pode assumir os dois papéis na ferramenta, podendo ser consumidor buscar serviços ou solicitar atendimentos, e também prestador, ao divulgar algum negócio seu que prover aos demais usuários. A Figura 6 mostra o diagrama de subsistemas.

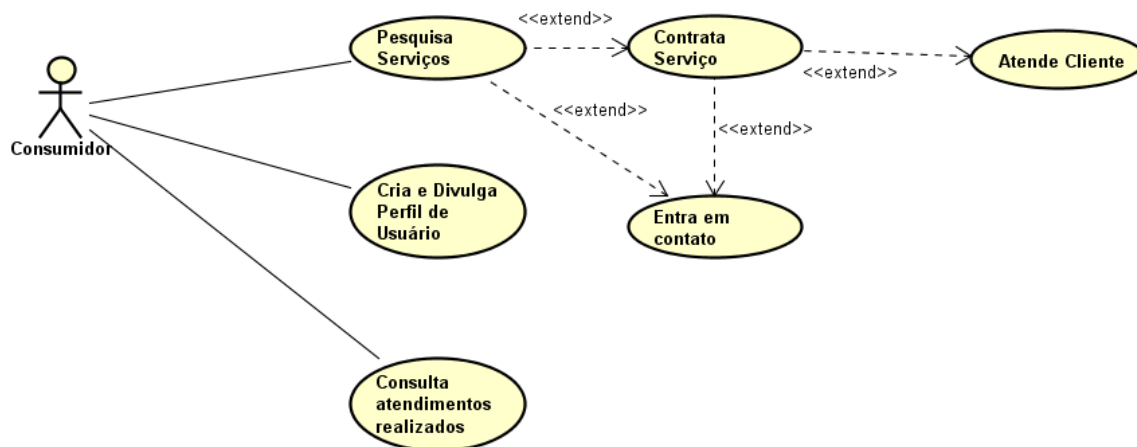


**Figura 6 – Diagrama de subsistemas.**

Após a definição das funcionalidades do sistema, procurou-se estabelecer caso de uso para cada aplicação.

Casos de uso, segundo PRESSMAN (2016), devem representar o software do ponto de visto do usuário. Logo, conclui-se que um caso de uso deve agregar valor ao usuário. Por exemplo, se neste projeto fosse definido um caso de uso como “Trocar senha de acesso”, este caso não agregaria valor ao usuário por si só. É uma funcionalidade válida para muitos sistemas, inclusive este, porém não deveria ganhar tanto destaque a ponto de se tornar um caso uso, afinal, o propósito de utilizar a ferramenta não tem a ver com trocar senha ou não, mas, sim, com atividades como buscar serviços, contratar serviços, entrar em contato com um prestador, e outros.

Portanto, os casos de uso, em concordância com a visão do usuário sobre o que um sistema como o proposto deverá funcionar, foram definidos criteriosamente para, juntos, tornarem-se uma síntese da proposta da aplicação.



**Figura 7 – Diagrama de casos de uso do subsistema Busca de Serviços**

A Figura 7 mostra os casos de uso do subsistema busca de serviços, em que o usuário, como papel de consumidor, irá interagir com o sistema. Percebe-se, portanto, que os casos de uso apresentados possuem relevância significativa para a visão do usuário consumidor, ao interagir com o sistema. Na Figura 8, os casos de uso do prestador são apresentados, levando em consideração a mesma ótica, quanto a casos de uso.

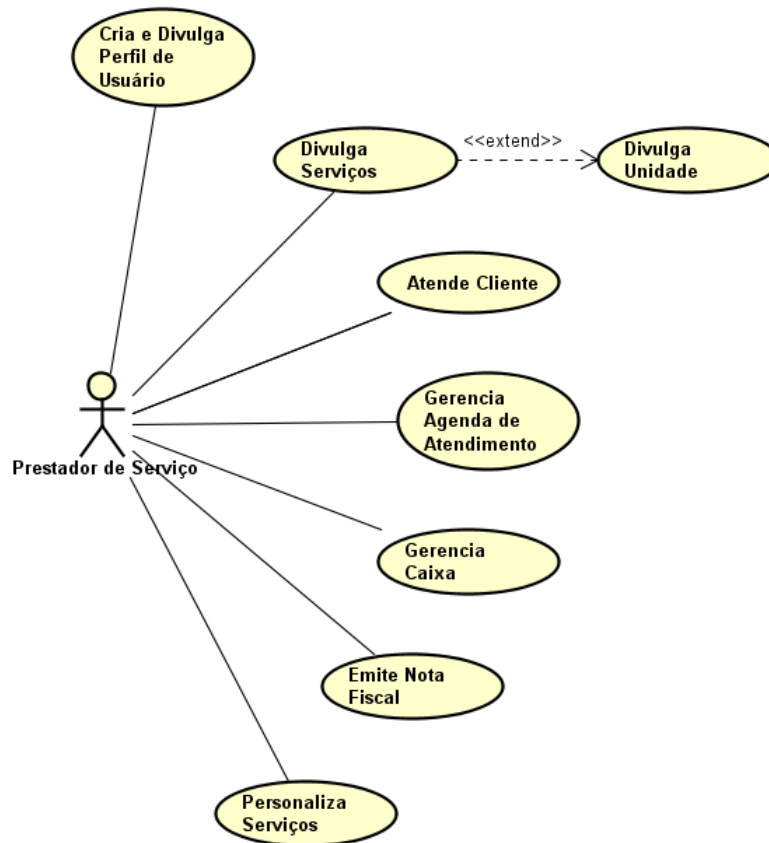


Figura 8 – Diagrama de casos de uso dos subsistemas prestação de serviços

## 5.5 REQUISITOS NÃO FUNCIONAIS

Para o sistema, é identificável, pela existência de requisitos como o de troca de mensagens instantânea, que o sistema deva possuir tempo de respostas consideravelmente rápidos, ao menos para os casos de uso em que se faça necessário uma comunicação em tempo real ente usuários. Para tanto, é declarado como requisito não funcional que o tempo de resposta ao utilizar recursos com a pesquisa de serviços (quando deve haver uma atualização constante das localizações móveis de unidades móveis) e contato, seja tão rápido que a mensagem, dado condições boas de conexão das partes envolvidas, chegue em no máximo um segundo.

Dentre outros requisitos não funcionais, pode-se citar a importância do aplicativo ter alta usabilidade, ou seja, que seja rápido de aprender, fácil de ser lembrado como de realizar as atividades nele, que seja agradável. Afinal, o público alvo do sistema será bem diversificado, incluindo pessoas com pouca

experiência em tecnologia. Assim, uma interface alinha a boas práticas de usabilidade é quase que obrigatório a este projeto, assim como tantos outros projetos de aplicativos móveis.

E, devido a opções de contratação e transação de pagamento com cartão de crédito pela plataforma, é necessário que a segurança dos dados seja assegurada, tanto na persistência de dados, como na troca de mensagens entre o aplicativo e as operadoras de cartão.

Por fim, deve ser garantido que o sistema, em sua construção possa ser feito de tal forma, que seja escalável, considerando o uso simultâneo de até milhares de usuários. O sistema deve suportar alta carga de troca de mensagens entre os usuários e o gerenciamento dos recursos persistidos em base de dados.

No próximo capítulo será definida a análise do sistema, identificando as entidades, os comportamentos do sistema, baseando-se nos requisitos e casos de uso explanados.

## 6 MODELO DE ANÁLISE

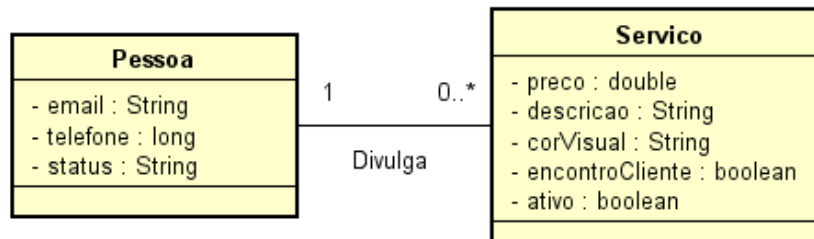
Após a etapa de especificação de requisitos, foi realizada a especificação de análise de sistema. “O objetivo do modelo de análise é fornecer uma descrição dos domínios informacional, funcional e comportamental necessários para um sistema baseado em computador.” (PRESSMAN e MAXIM, 2016, p. 154).

Aqui é estabelecido, isto posto, o levantamento das entidades do sistema, dicionário de dados de cada uma das entidades, além da especificação comportamental do software. Por fim são apresentadas considerações sobre os resultados desta etapa e como ela influencia no desenvolvimento das etapas seguintes.

### 6.1 LEVANTAMENTO DE ENTIDADES E SEUS RELACIONAMENTOS

Como primeira etapa do modelo de análise, procurou-se definir o modelo lógico, contendo o levantamento de entidades com seus relacionamentos, atributos de dados de cada entidade e, para completar, uma descrição de cada entidade e de todos seus atributos.

O ponto de partida para levantamento de entidades foi a pessoa, que reflete o cadastro de usuários do sistema. A partir da definição de pessoa como entidade, o segundo modelo de dados identificado foi o serviço. Estas entidades foram as primeiras a serem identificadas pelo fato de que a proposta central da ferramenta é divulgar serviços que pessoas prestam. Logo, naturalmente o processo de levantamento de entidades e relacionamentos iniciou-se por estes modelos. Dado esta definição, procurou-se logo estabelecer o relacionamento entre as entidades. Como especificado anteriormente, uma pessoa pode divulgar um ou mais serviços, mas também nenhum. Isso porque, vale lembrar, a pessoa é qualquer usuário, e teremos dois tipos: consumidor e prestador. A Figura 9 demonstra a especificação até este ponto.



**Figura 9 – Fragmento do diagrama de classes, demonstrando o primeiro passo da especificação de entidades**

Repare que o relacionamento definido não foi composição, agregação. Isto porque, por mais que não faça sentido o serviço existir sem a pessoa que o presta, registros não serão apagados do sistema. Como se trata de um serviço de interação entre pessoas, inclusive com transações de pagamento, o prestador de serviço que quisesse desvincular-se da plataforma, não iria apagar seus dados por completo. Para isso, o cadastro de pessoa contará com o campo status, o qual indicará se a pessoa está ativa ou inativa no sistema. Para assim, por exemplo, mesmo o prestador ou consumidor desligando permanentemente seu cadastro, estes dados iriam permanecer para prestação de contas, nos registros de atendimentos históricos. Assim será possível os envolvidos em um atendimento terem sempre seus dados disponíveis sobre quem prestou o serviço e quem foi atendido.

O usuário a se cadastrar pode ser uma pessoa física ou jurídica. É sabido que pessoas físicas possuem dados de sua natureza como CPF, data de nascimento, já pessoas jurídicas possuem informações como a razão social, o CNPJ. Porém, ambas as pessoas devem ter, em seus cadastros, dados de contato e acesso, como telefone e e-mail, além de outros. Pensado assim, este cenário foi colocado como um relacionamento entre entidades do tipo generalização, ou, o termo mais voltado a orientação de objetos: herança. Furtado (2010) menciona que este relacionamento entre entidades serve para evitar duplicidade de dados e operações. Assim foram definidas mais duas entidades: pessoa física e pessoa jurídica.

Ainda sobre pessoas, foram julgados alguns conjuntos de dados como entidades separadas, as quais compõem a entidade principal *pessoa*,

como os dados dos dois tipos de acesso e interesses da pessoa. Estes dados são opcionais, portanto fazem mais sentido ficarem separados do modelo de origem.

Por seguinte foram levantadas mais entidades que se relacionassem com as duas já especificadas. Por exemplo, serviços possuem uma ou mais unidades de atendimento, estas que podem ser do tipo fixa ou móvel.

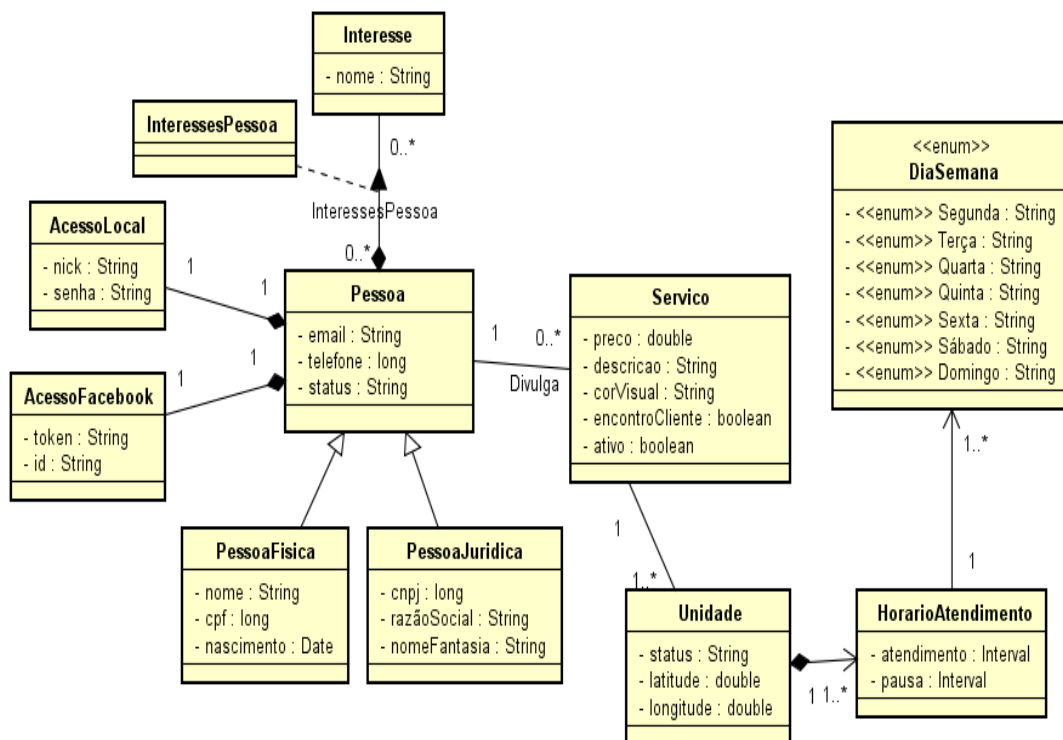
As unidades, inicialmente, foram especificadas para serem, também, entidades separadas, ambas com a generalização para entidade unidade. Entretanto, na etapa de definição do modelo de projeto, foram identificados alguns fatores que resultaram em uma revisão desta modelagem, tratando as unidades, por fim, como uma entidade só, mesmo que isso signifique possuir dados de localização desnecessários para quando estivermos tratando de unidades móveis – estas que não possuem paradeiro, não possuem uma localização geográfica determinada.

Os motivos que levaram a esta mudança foram os seguintes:

- O único atributo que diferencia os dois tipos de unidade é o de coordenadas de localização.
- Mesmo com a união da representação dos dois tipos, nenhum atributo como “tipo de unidade” foi necessário ser adicionado ao conjunto de atributos. Isso porque o atributo localização pode ser empregado para calcular, de certa forma, se uma unidade é fixa ou móvel. Se for unidade móvel significa não possuir dado de localização, então esta informação já está dita neste atributo – nulo ou preenchido.

Unidade, além de localização ou não, possui horários de atendimento – um ou vários. Horário, portanto, foi definido como uma entidade que compõe unidade. Um modelo de horário possui dados de faixa de hora de atendimento e pausa, além de um conjunto de dias da semana. Como os nomes de dias são constantes, foi definido uma entidade com o estereótipo *enum*, termo empregado em orientação à objetos para referenciar um conjunto de dados constantes, como este caso, tendo os valores de cada nome de dia. Assim, o horário foi relacionado com este *enum* dias da semana, com cardinalidade um para muitos.

Até aqui foram definidas entidades, atributos e relacionamentos que contemplam o essencial para o cadastro de pessoa e divulgação de serviços. A Figura 10 mostra o estado da definição neste ponto. Há a generalização de *pessoas físicas e jurídicas*, as entidades que compõe o cadastro de pessoa (*acesso local, acesso facebook, interesses de pessoa*). Note que uma pessoa pode ter vários interesses, que são palavras específicas para compor futuras formas de melhorar a experiência do usuário com pesquisas de serviços, dentro da ferramenta, mais próximas de seus interesses sobre serviços. Um termo textual de interesse pode ser utilizado para várias pessoas. Agora, também, é relacionado o *serviço* com a entidade *unidade* e sua composição com *horário de atendimento*.



**Figura 10 – Construção parcial do diagrama de classes contemplando o cadastro de pessoa e divulgação básica de serviços**

Prosseguindo com o levantamento de entidades, o próximo passo foi adicionar modelos que representassem a ocorrência de atendimentos de serviços. Quando se pensa em atendimento, neste cenário, algo como a seguinte expressão pode ser empregada: uma pessoa recebe atendimento de um ou mais serviços, e serviço pode atender um ou mais pessoas. Qual seria o lógico, portanto? Definir mais uma associação entre pessoa e serviço, com a

denominação “é atendida”, apontando para pessoa, transmitindo que uma pessoa é atendida por um ou mais serviços, com a cardinalidade muitos para muitos. Para completar teríamos uma entidade associativa denominada atendimento, contendo data, valor do atendimento, dentre outros atributos.

Entretanto, reespecificando melhor como o sistema iria funcionar, em pouco tempo percebeu-se que pessoas não são atendidas por um serviço, mas, sim, em uma unidade de atendimento, a qual é de um serviço. Com esta nova perspectiva, ganha-se a informação de em qual unidade foi realizado o atendimento para cada ocorrência.

E o prestador do serviço que atendeu ao chamado, será então a pessoa dona do serviço? Os requisitos mostram que cada unidade possui uma pessoa responsável, o que quer dizer não necessariamente ser a mesma pessoa que cadastrou o serviço. Assim, definiu-se um relacionamento com a entidade pessoa para unidade, representando quem é responsável por cada unidade de atendimento.

Ainda na unidade, foi adicionado um relacionamento com a entidade agenda, onde a unidade tem até uma unidade de atendimento. A agenda é opcional, portanto ficou separada. O prestador do serviço irá ativar este recurso para seu serviço ou não. Quando os consumidores ou prestadores resolverem agendar um atendimento, irão informar o dia, horários de início e término, e este agendamento possuirá um *status*. Este *status* servirá para controlar o andamento da solicitação, desde a solicitação, até o atendimento ter sido bem-sucedido ou não. Assim, configura-se a existência da entidade *horário agendado*, que interliga a agenda (a qual é de uma unidade, que tem um prestador responsável e é de um serviço) e a pessoa consumidora.

Agora, voltando à entidade *serviço*, temos seu tipo, que será uma entidade separada, uma vez que um serviço é de um tipo e, portanto, um tipo pode estar associado a múltiplos serviços. Ainda, o tipo de serviço é de uma área. Estas áreas podem servir para agrupar serviços, classificando-os e disponibilizando-as para filtro de serviços nas interfaces de busca por serviço.

O serviço, ainda, possui um formulário de atendimento e um formulário de cadastro, ambos opcionais. Foram definidas duas entidades para este propósito, as duas com realização para outra denominada *formulário*. Este formulário possui um ou vários campos. Estes campos possuem um rótulo, tamanho do futuro valor, tipo de campo e um atributo que sinaliza se será um campo opcional ou não no ato do preenchimento do formulário. Este tipo de campo, vale lembrar, quando é do tipo discreto, possuirá uma coleção de opções de o campo para o usuário do formulário selecionar uma opção, dentre as disponíveis. Opções, estas, que devem possuir um valor a ser apresentado ao usuário ao usar o formulário. E, por fim, estes campos possuirão uma ocorrência de preenchimento, que existirá mediante a ocorrência do uso de formulários de atendimento ou cadastro pelo usuário. Aqui temos, então, a adição de entidades valores de campos, preenchimento de formulários.

Para fechar, estes valores de campos são ligados à entidade de atendimento, para relacionar com cada atendimento o preenchimento completo de formulários de atendimento e cadastro do serviço que foi prestado. No fim, tem-se o caixa como entidade, o qual possui uma coleção de atendimentos associados. O caixa possui as datas de abertura, fechamento, e um valor inicial. O valor total do caixa não existe como atributo, já que pode-se calcular o total de valores de atendimentos associados ao caixa, em concordância com a 3ª regra formal de normalização de dados. “(...) se algum atributo pode ser calculado a partir dos atributos básicos armazenados, ele não deve fazer parte da relação” (FURTADO e COSTA JUNIOR, 2010, p. 295).

Todas essas entidades e relacionamentos comentados encontram-se, agora, na versão completa do diagrama de classes, o qual pode ser conferido na Figura 11.



## 6.2 COMPORTAMENTO DO SISTEMA

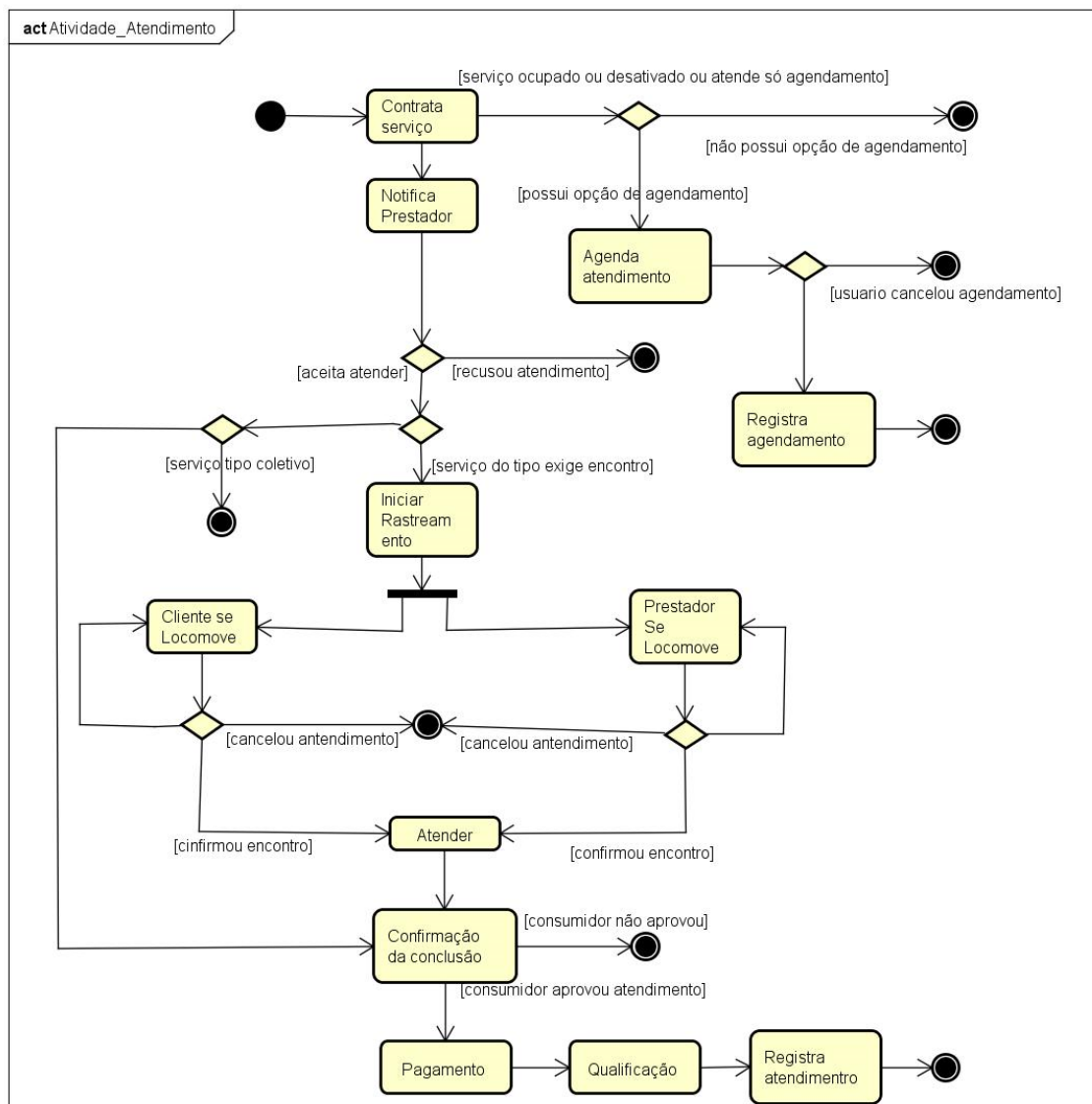
Definido o modelo de dados do sistema, o passo seguinte foi definir o modelo comportamental. Este modelo, segundo Pressman e Maxim (2016), sinaliza como o programa deve responder às ações externas, de usuários e demais atores.

Ainda, Pressman e Maxim (2016) orientam que as seguintes etapas devem existir nesta especificação do comportamento:

- Avaliar todos os casos de uso para entender completamente a sequência de interação.
- Identificar eventos que controlam a sequência de interação.
- Definir uma sequência lógica para cada caso de uso.
- Definir um diagrama de estado para o sistema.
- Examinar o modelo comportamental para garantir a consistência.

Assim, foram seguidos os passos para a construção deste modelo. A identificação de atividades de interação foi realizada. Uma atividade de destaque, rica em detalhes, é o atendimento ao consumidor – a prestação efetiva do serviço. Existem variantes interessantes, caminhos alternativos, que colocam esta atividade como a mais complexa do sistema. É onde realmente a ferramenta proposta se realiza por completo, pois atende a diversas situações de prestação de serviços. Este diagrama de atividade pode ser visto na Figura 12.

C



**Figura 12 – Digrama de atividade de atendimento.**

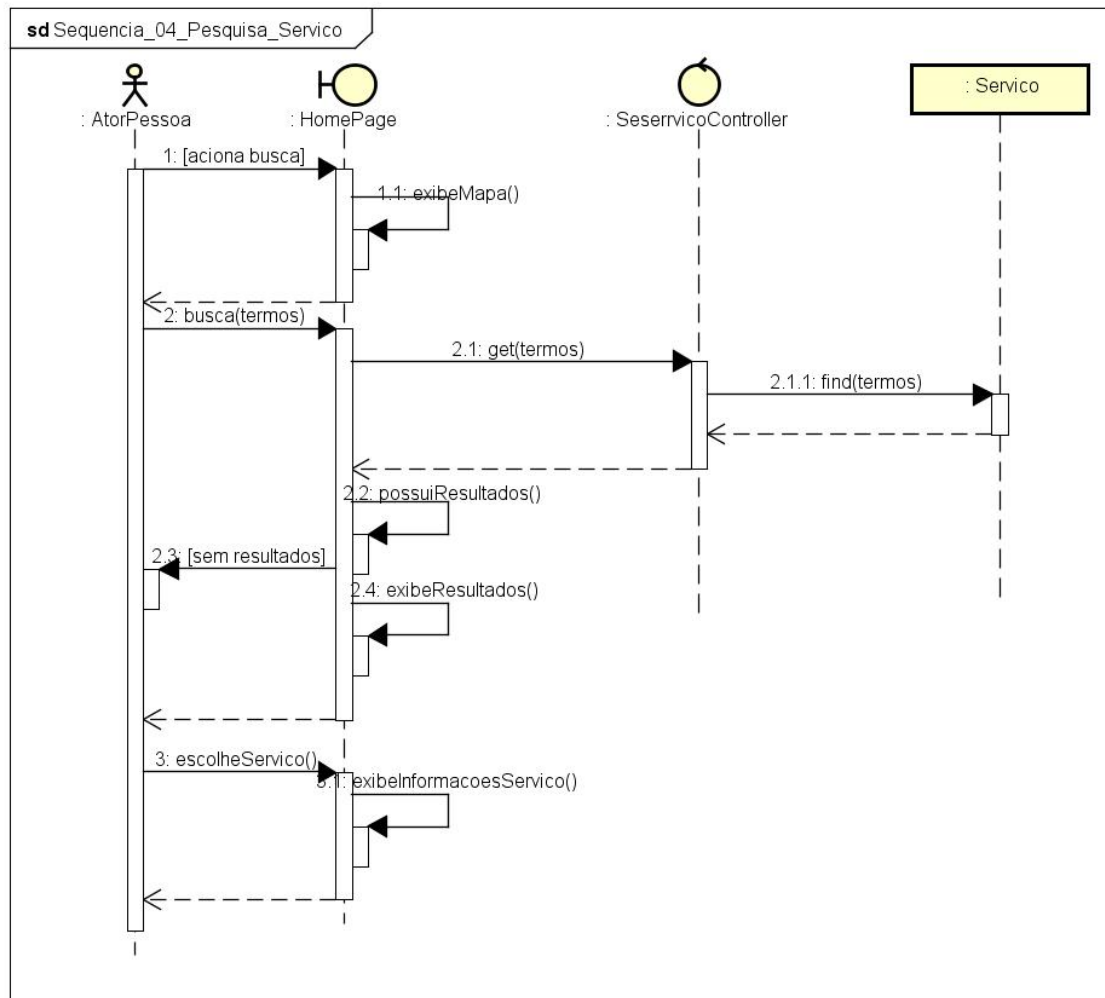
No diagrama, a condição *Serviço ocupado ou desativado* significa que a unidade do serviço está ocupada (não é serviço do tipo coletivo) ou a unidade está fechada, o sistema tentará fornecer ao usuário a opção de agendamento, caso o serviço esteja apto a isso. O usuário pode não querer agendar.

Já a condição *Serviço coletivo* indica que o serviço está configurado como atendimento coletivo. Assim, não há a ocorrência de atendimento, pagamento e qualificação. Depois de notificar o prestador, a atividade é encerrada.

Uma consideração importante é sobre o agendamento. Como especificado anteriormente, o agendamento de serviços poderá ser a única forma de contratar o serviço ou não. Antes de a atividade *atendimento* ocorrer, é necessário que a agenda esteja configurada para o agendamento ser opcional, ou para o serviço nem existir a existência do recurso de agenda. Assim, esta atividade será possível de ocorrer, para determinado serviço.

Como pôde ser observado no diagrama de atendimento, existem diversos caminhos condicionais, bifurcações, que levam a diferentes resultados e comportamentos do sistema. Aqui fica claro o que foi pregado capítulo 5, que fala sobre a ideia desta ferramenta, onde foi mencionado como este serviço poderia, e deveria ser dinâmico para atender às diversas necessidades de divulgação e atendimento de serviços.

Continuando com a especificação do modelo dinâmico do sistema, foram definidos os digramas de sequência para cada caso de uso definido na etapa de especificação de requisitos. “O diagrama de sequência mostra as interações entre os objetos, organizadas sequencialmente no tempo, para realizar uma funcionalidade do sistema” (FURTADO e COSTA JUNIOR, 2010, p. 214). Neste ponto é iniciado o processo de definição de classes de fronteira, classes de interface de usuário, começando a perceber o sistema com mais camadas de classes, cada uma com sua devida responsabilidade dentro do sistema. Este assunto será melhor tratado na etapa do modelo de projeto.



**Figura 13 – Diagrama de sequência de pesquisa de serviços.**

Os diagramas de atividade, quando bem definidos, conseguem expressar exatamente como a comunicação entre as classes irá ocorrer. Considera-se este artefato de análise bastante completo em relação aos objetivos propostos por ele. Assim, o diagrama de comunicação, o qual possui os mesmos propósitos, diferindo pela maneira de representar, não foram construídos. “Normalmente, pode-se escolher entre utilizar o diagrama de comunicação ou o diagrama de sequência, dependendo do enfoque que se deseja dar à representação da interação” (FURTADO e COSTA JUNIOR, 2010, p. 217). Ainda, segundo Furtado e Costa Júnior (2010), o diagrama faz a mesma coisa que de comunicação, com o acréscimo de demonstrar a ordenação temporal das mensagens trocadas entre os objetos.

Outro tipo de artefato produzido nesta etapa foi o diagrama de estados. Este diagrama, segundo Pressman e Maxim (2016), modela estados de

objetos, bem como suas ações quando ocorrem transições entre os estados. Furtado e Costa Júnior (2010) definem este artefato como um diagrama que complementa o diagrama o diagrama de classes, mostra todos os estados possíveis de objetos de certas classes e os eventos do sistema que provocam as mudanças de estados.

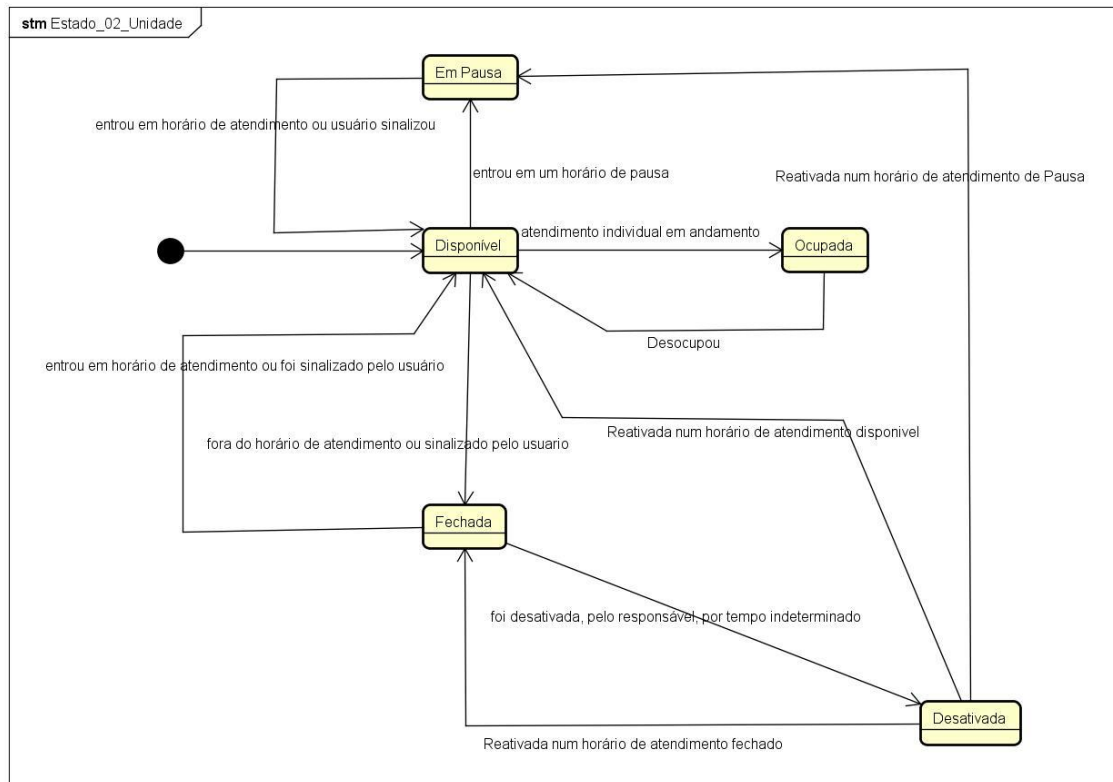


Figura 14 – Diagrama de estados da classe unidade.

A Figura 14 mostra um dos diagramas de estados definidos para o projeto, o da classe unidade. Para esta classe, há os estados *disponível*, *em pausa*, *fechada*, *desativada* e *ocupada*. O sistema deve analisar o horário de atendimento da unidade, em comparação com a hora do sistema. Caso esteja dentro do horário de atendimento, o estado se torna *disponível*. O mesmo ocorre para os estados *em pausa* e *fechada*. Ou seja, se a unidade está *disponível*, *em pausa* ou *fechada*, o sistema realiza a troca desses estados, automaticamente

Entretanto, há a opção de o usuário trocar o estado de *em pausa* para *disponível* e entre este e *fechada* manualmente. Neste caso, mesmo dentro de um horário de atendimento específico, o usuário responsável pela unidade poderá realizar as trocas de estado mencionadas.

Estes estados de unidade refletem na condição se o serviço poderá ser contratado ou não em determinado momento. Porém, nada impede de a unidade do serviço aparecer em buscas por serviços que não contenham filtro de busca por somente unidades disponíveis.

O estado *ocupada* reflete o momento em que a unidade está atendendo um consumidor, e este serviço não é do tipo atendimento coletivo. Ou seja, a unidade fica alocada, ocupada, com o atendimento realizado. Ao término ou cancelamento do atendimento, a unidade volta a ficar disponível.

Este, portanto, foi o modelo de análise definido para o projeto. Com os artefatos produzidos nesta etapa foram planejados e descritos cada modelo de dados, bem como seus atributos. Foram diagnosticados como a interação do sistema com o usuário deve ocorrer, e como a interação entre os objetos da aplicação devem funcionar. Após esta etapa, o próximo passo foi a definição do modelo de projeto, o qual consiste em definir um guia para a implementação do software. Este, portanto, é o tema do próximo capítulo.

## 7 MODELO DE PROJETO

Após definir o modelo de análise, o próximo passo seguido na construção da ferramenta foi a definição do modelo de projeto. Este artefato contém a projeção de classes, arquitetura, interfaces e componentes do sistema. Cada projeto destes citados possui um propósito específico e que, juntos, se traduzem um modelo a ser seguido para desenvolver dado sistema.

Para Pressman e Maxim (2016), é nesta etapa de desenvolvimento que se estabelece a qualidade do software a ser desenvolvido. Ainda, coloca em sua obra itens que garantem a qualidade do modelo de projeto e, assim, a qualidade do futuro software:

- Deve exibir uma arquitetura que seja composta por componentes e que possa ser implementado de forma evolucionária.
- Deve ser modular; ou seja, o software deve ser dividido logicamente em elementos ou subsistemas, de modo que seja fácil de manter e testar.
- Deve conter representações distintas de: dados, arquitetura, interfaces e componentes.
- Deve ser desenvolvido usando como referência as informações obtidas durante a etapa de análise de requisitos de software.

Ainda, Pressman e Maxim (2016, p. 226) afirmam que “o projeto de software é a última ação da engenharia de software na atividade de modelagem e prepara o cenário para a construção (geração de código e testes)”.

Portanto, esta etapa pode ser considerada crucial para o sucesso de qualquer software frente aos seus objetivos.

O projeto representa o software de várias formas diferentes. Primeiramente, a arquitetura do sistema ou do produto tem de ser representada. Em seguida, são modeladas as interfaces que conectam o software aos usuários, a outros sistemas e a dispositivos, bem como a seus próprios componentes internos. Por fim, os

componentes de software usados para construir o sistema são projetados. (PRESSMAN e MAXIM, 2016, p. 224).

Nesta seção serão tratados todos estes modelos de projeto do projeto de software. Ao final são definidos os resultados desta etapa para a construção da ferramenta *iservice*.

## 7.1 PROJETO DE ARQUITETURA

A primeira etapa de definição do modelo de projeto foi o projeto de arquitetura. Esta fase consiste em definir a estrutura necessária para a organização do software, desenvolvendo um meio de comunicação entre componentes – e subsistemas – da aplicação.

Para este projeto, foi empregado o estilo de arquitetura em camadas, o qual consiste em dividir o software em camadas de componentes, em que cada camada interage com suas vizinhas, provendo ou consumindo recursos para o funcionamento do sistema. A camada mais externa é a de interação com o usuário e a mais interna de interação com a máquina. A aplicação a ser desenvolvida, entretanto, assim como a maioria na atualidade, utiliza recursos já existentes para abstrair atividades de mais baixo nível, como a leitura e gravação de dados em um disco rígido, por exemplo.

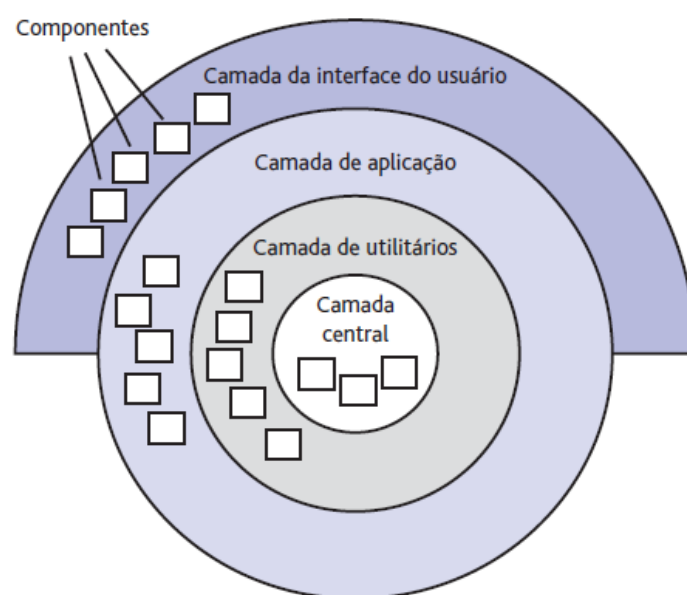


Figura 15 – Modelo de arquitetura em camadas (PRESSMAN e MAXIM, 2016)

Em sua obra, Pressman e Maxim (2016) mencionam que a arquitetura de projetos de aplicativos móveis geralmente é a em camadas, com as camadas de interface do usuário, camada de negócio e camada de dados. Além disso, existem dois conceitos importantes sobre projetos de aplicativos móveis e são importantes de serem analisados: clientes magros e clientes ricos – ou gordos.

Segundo Pressman e Maxim (2016), em um cliente magro apenas a interface do usuário é disposta no dispositivo do cliente, e as demais camadas ficam do lado do servidor da aplicação centralizado. Já cliente rico é caso em que pode ser atribuído ao cliente mais recursos das outras camadas além da interface.

Para este projeto foi compreendido que a utilização do estilo cliente magro fizesse mais sentido. Primeiro porque os requisitos demandam que os clientes da aplicação sejam altamente dependentes de outros clientes para possuir a informação necessária. Falando mais especificamente da busca por serviços, com o recurso de compartilhamento de localização geográfica dos usuários, bem como interação entre eles em tempo real, faz com que os clientes necessitem de conexão constante para troca de informações entre eles. Esta troca de informações deve ser administrada por uma aplicação servidora, uma *web service*, que interligue todos os clientes, afinal este compartilhando de informações de cada usuário não será privado, necessariamente. Em muitas ocasiões deve ser compartilhado a todos os usuários da informação.

Dado este cenário, entende-se, frisando, que cliente magro possa ser mais adaptado à realidade desta aplicação, uma vez que não faz sentido colocar toda a responsabilidade de processamento e regras em cada cliente, se, de qualquer forma, deve existir uma interface – API – que interconecte os clientes da aplicação.

Além disso, aplicativos móveis baseados na web têm se tornando cada vez mais comum. Semelhante a *webapps*, os aplicativos móveis podem ser, na verdade, apenas uma apresentação das informações e interação com o usuário, e o processamento e aplicação de regras, bem como o repositório de dados, ficando em um servidor na nuvem, ou chamada também de *API(Application Programming Interface)*.

A computação de serviço desobriga os desenvolvedores de aplicativos móveis a integrar código-fonte de serviço no cliente que está sendo executado em um dispositivo móvel. Em vez disso, o serviço é executado fora do servidor do provedor e é pouco acoplado aos aplicativos que o utilizam por meio de protocolos de troca de mensagens. Um serviço típico fornece uma interface de programas aplicativos (API, application programming interface) para permitir que seja tratado como uma caixa preta abstrata. (PRESSMAN e MAXIM, 2016, p. 406).

Assim, portanto, uma vez definido a necessidade de uma aplicação servidora que interligue os clientes, foram definidos dois subsistemas reais (em termos de implantação), da aplicação: *API* e *App*. Estes dois subsistemas irão se comunicar, trocando mensagens por meio da nuvem – internet. Além disso, o banco de dados da aplicação deverá estar conectado diretamente ao subsistema API, para gerenciar os dados de negócio a serem persistidos. Entretanto, não se faz necessário que o banco de dados e API estejam na mesma máquina instalados. Isso aumenta a flexibilidade de implantação do sistema na nuvem, tornando possível a abstração de locais de instalação para o subsistema API e do banco de dados. Uma visão dos subsistemas e banco de dados pode ser conferida na Figura 16.

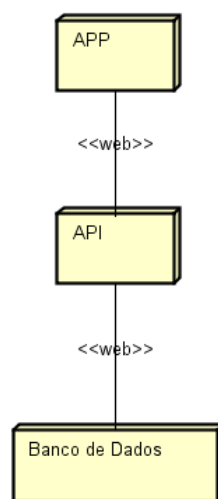
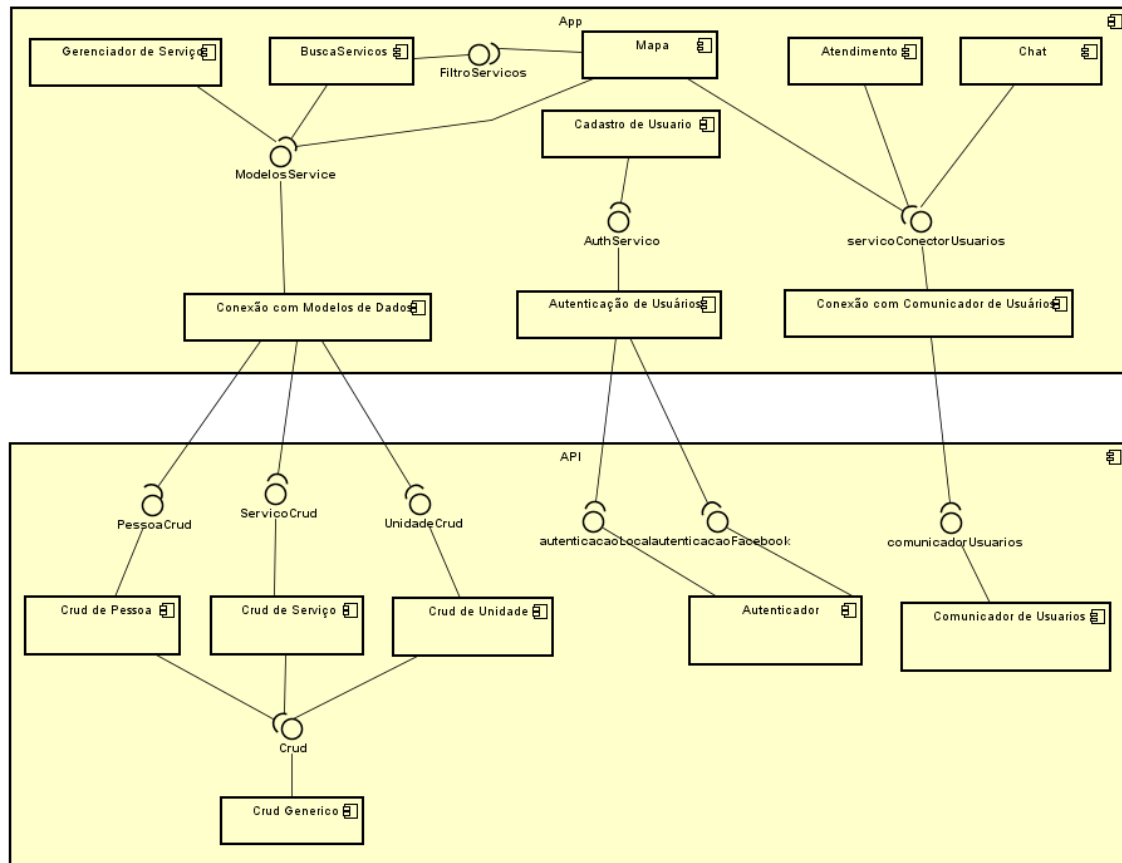


Figura 16 – Diagrama de implantação

Dado a definição do estilo de arquitetura e da divisão da aplicação em subsistemas e responsabilidades de armazenamento e processamento, o sistema foi analisado em termos de funcionalidades e recursos que deveriam existir para que todos os objetivos do sistema, em termos funcionais, fossem alcançados. Este levantamento foi a origem da definição de componentes da aplicação. A separação do sistema em subsistemas foi o primeiro passo para definir módulos da aplicação. Com o esclarecimento das duas partes do sistema, *API* e *App*, o esclarecimento sobre que componentes poderiam existir em cada um dos lados veio à tona. De um lado (*API*), componentes que gerenciam os dados de negócio persistidos no banco de dados, e outros que realizam a mediação, interconexão, entre usuários da aplicação. Do outro, componentes preocupados em receber comandos dos usuários e prover as necessidades deste em cada interface de usuário.

Com esta nova visão bem dividida do sistema, foi possível, ao final, chegar aos componentes da aplicação, como mostra a Figura 17. Neste diagrama é projetado a arquitetura do sistema com seus componentes e relações entre eles. É possível observar cada dependência entre componentes. O retângulo maior superior representa o aplicativo como um componente que encapsula os seus subcomponentes. O retângulo maior inferior representa a *API*, que, de igual forma, possui seus componentes para compor seu funcionamento.



**Figura 17 – Diagrama de componentes.**

Com a definição de que componentes devem existir, além das interfaces que cada iria prover aos demais componentes, o passo seguinte foi especificar cada componente detalhadamente, à nível de classes. Este é o assunto tratado nas seções seguintes (7.2 e 7.3).

## 7.2 PROJETO DE COMPONENTES

Após a definição da arquitetura, momento em que foram levantados os componentes da aplicação, foi realizado um processo de especificação detalhado de cada componente existente. Este trabalho resultou em um diagrama de classes de projeto, especificações de implementação em pseudocódigo e uma descrição de interfaces de cada componente.

Para cada componente foram definidos a função, os objetivos, questionamentos de como os objetivos poderiam ser alcançados e, então, especificado um diagrama de classes de projeto do componente. Durante este levantamento, foram removidos e substituídos alguns módulos, refinando o agrupamento dado a cada uma das fatias do sistema. A preocupação principal,

durante este processo, foi em definir uma correta interface pública do componente, de tal forma que este pudesse ser reutilizado sempre que outro componente do sistema necessitasse de suas funcionalidades. Além disso, foram levados em considerações padrões de projeto e boas práticas de engenharia de software. Estas considerações foram em cima dos seguintes conceitos:

### **Baixo acoplamento**

Deve existir o mínimo de interação – dependência – entre os módulos. Isso ajuda a realizar a manutenção e a própria criação do sistema, uma vez que a integração entre os componentes, quando simples, reduz as chances de ocorrer falhas de sistema quando os componentes forem integrados para compor o sistema macro, o sistema completo.

### **Alta coesão**

Os componentes definidos, cada um, agrupam funcionalidades semelhantes, ou que juntas possui o mesmo objetivo. Desta forma, foi possível definir uma quantidade equilibrada de componentes, sem que houvesse demasia de fatias do sistema desnecessárias, e que preservasse o conceito de uma unidade, neste caso componente, ter somente um propósito, uma responsabilidade. Como exemplo pode ser citado o componente *Comunicador de usuários*, o qual é composto de algumas classes, mas que trabalham juntas para implementar o funcionamento esperado por outros componentes ao utilizarem sua interface.

### **Inversão de dependência**

Os componentes da aplicação foram projetados de tal forma que nenhum deles dependa de uma classe concreta, que nenhum módulo precise conhecer detalhes de implementação (abstração baixa) sobre outro componente, pelo contrário. Um componente qualquer A depende de uma abstração do componente B, assim como, se B necessitar dos recursos do componente C, este irá se relacionar uma interface abstrata de C. Este mecanismo de organização e integração de componentes e classes é denominado inversão de dependências e ajuda na redução do acoplamento

entre classes – e conseqüentemente componentes – do sistema e melhora as condições de teste de software, como afirmam Pressman e Maxim (2016).

### 7.3 PROJETO DE CLASSES

No modelo de análise foram definidas classes de modelo de dados da ferramenta, como *pessoa* e *serviço*. O que essas classes representam são enxergadas pelo usuário, pois classificam dados de negócio. Quer dizer, o usuário irá lidar com o cadastro de pessoa, de serviço, atendimentos, enfim. Agora, na etapa de projetos, as classes do sistema são expandidas em números e em detalhes, com um nível de abstração mais baixo que as definidas até a etapa anterior.

O Projeto de classes visa completar a definição das classes do sistema, acrescentando as classes de fronteira (controladores) e classes de interface. As classes de projetos estão divididas entre as três camadas do estilo arquitetural MVC, que são elas: *view*, *controller*, *model*. Aqui serão chamadas de classes de fronteiras, controladoras e modelos, respectivamente. Entretanto, as classes de modelo, nos diagramas de classes de projeto, são imitadas por não terem sido mudadas desde então e, para melhor visualização das novas classes, evitando grande número de conexões nos novos diagramas de classes de projeto.

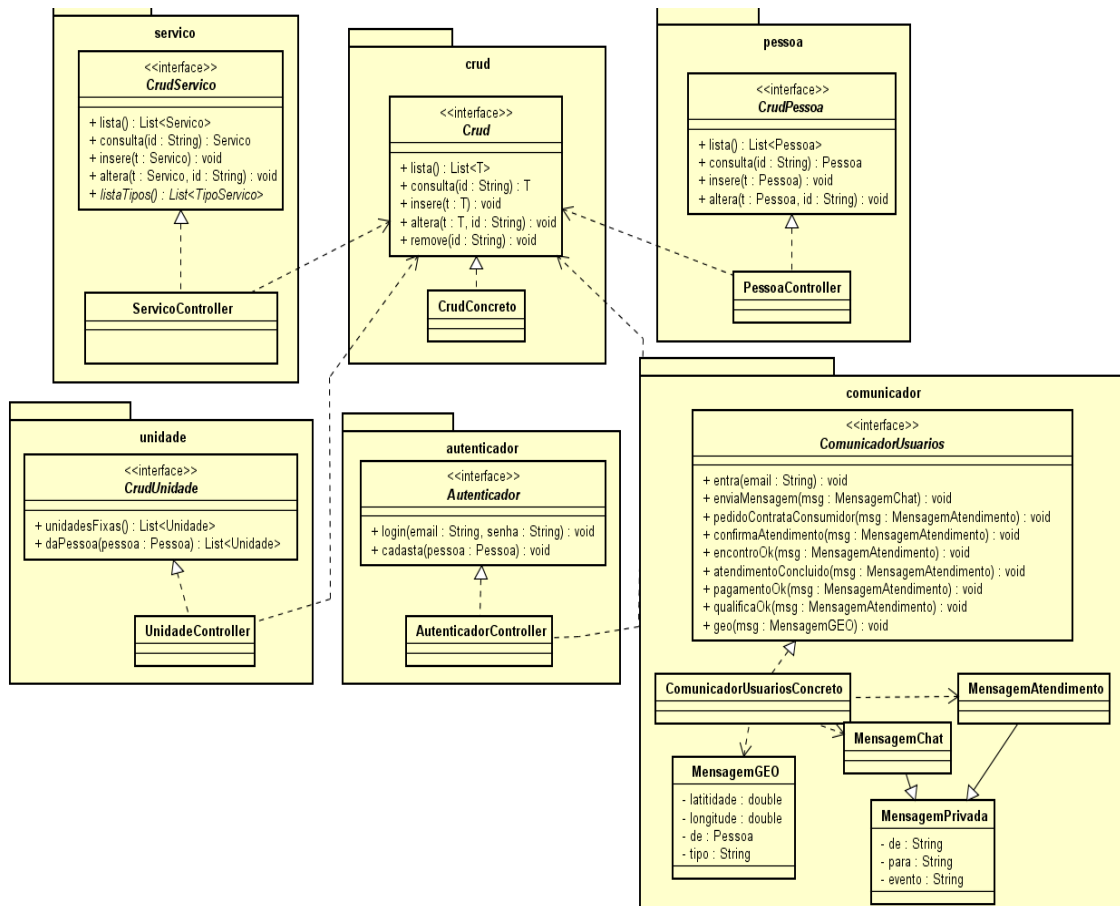


Figura 18 – Diagrama de classes de projeto do subsistema API

As figuras 18 e 19 mostram os diagramas de classes de projeto, estas que compõem cada componente do sistema – aqui os componentes são representados por pacotes. É possível observar que cada componente possui, no mínimo, uma classe abstrata (interface) e uma classe concreta – que implementa a interface. Ainda, neste diagrama, é possível identificar as dependências entre os componentes (setas pontilhadas com origem em um componente e destino em outro).

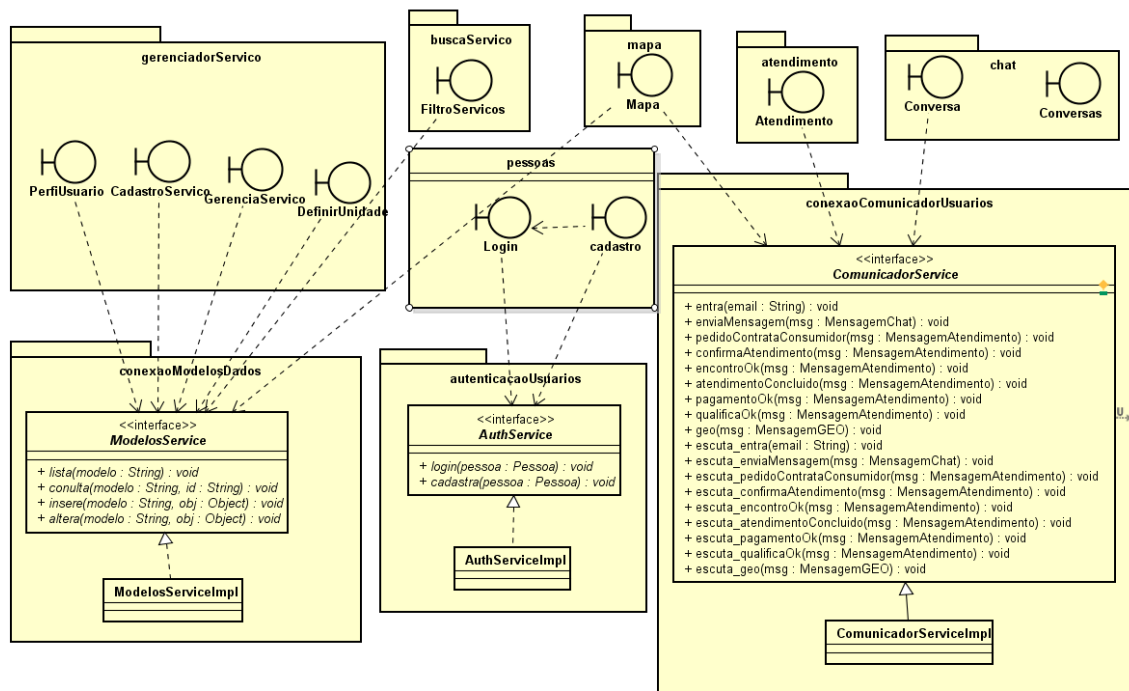


Figura 19 – Diagrama de classes de projeto do subsistema App

Prosseguindo com a especificação dos componentes, após levantar e definir a existência de cada classe que compõe os módulos, foram definidas as especificações de interfaces e classes concretas, detalhando cada um dos métodos de todas as classes de projeto. A Figura 20 mostra o detalhamento realizado para a interface de um dos componentes – *Crud de Serviço*.

<b>Método</b>	<b>URI</b>	<b>Corpo da Requisição</b>	<b>Especificação</b>
lista():lista<Pessoa>	/pessoa		Retorna uma lista de pessoas cadastradas no sistema
consulta(id)	/pessoa/{id}		Retorna um registro de pessoa cadastrada no Sistema que tenha o id especificado na URI.
insere(pessoa:Pessoa)	/pessoa	pessoa:Pessoa	Insere uma nova pessoa especificada no corpo da mensagem
altera(pessoa:Pessoa, id)	/pessoa/{id}	pessoa:Pessoa	Altera a pessoa informada no corpo da requisição, e que tenha o id informado na URI

**Figura 20 – Interface web do componente crud de pessoas**

Também, para cada classe concreta dos componentes, foram especificados pseudocódigos, que servem como um guia da lógica necessária para alcançar os objetivos de cada método de cada classe do sistema, para completa implementação das classes, alcançando objetivos dos componentes e, portanto, da aplicação. A Figura 21 exhibe um exemplo de especificação de lógica, da classe *AutenticadorConcreto*, do componente *Autenticador*.

<b>Classe:</b> AutenticadorConcreto	
<b>Método</b>	<b>Implementação</b>
login(email:string, senha:string)	<pre>var filtros = preparaFiltros(onde   pessoa.local.email == email E   pessoa.local.senha == criptogra(senha) ) se(this.crudPessoa.consulta(filtros) !=   nulo){   responde(200); }senao{   responde(401); }</pre>
cadastra(pessoa:Pessoa)	<pre>se(dados da pessoa estão corretos){   this.crudPessoa.insere(pessoa);   responde(201) }senao{   responde(400) }</pre>

Figura 21 – Quadro de especificação de pseudocódigo de cada método da classe *AutenticadorConcreto*

#### 7.4 PROJETO DE INTERFACES

Este é considerado um dos pontos que mais agrega qualidade a qualquer projeto de software que tenha como foco o ator humano e suas necessidades de interação com o sistema a utilizar. Neste momento é desenhado, elaborado, toda a forma de conversa entre a máquina, por meio do software pretendido, e o humano.

Dado a importância desta etapa de projeto, o projeto de interfaces, portanto, foi elaborado com devido cuidado e atenção, considerando a preocupação em atender corretamente as expectativas de um usuário frente aos seus objetivos de utilizar a aplicação proposta. Como declaram Pressman e Maxim (2016), o projeto de interfaces de usuário torna efetiva a comunicação entre o computador e o ser humano.

As interfaces dos usuários foram definidas seguindo o documento de requisitos e documento de análise deste projeto (Apêndices A e B). Analisando cada requisito, e suas visões nos casos de uso, chegou-se à conclusão das

interfaces de usuários da aplicação – as quais podem ser conferidas por completo no apêndice C, documento de projeto, seção projeto de interfaces.

Para especificação das interfaces do usuário, foi empregada a técnica de prototipagem, determinando modelos das telas do sistema, além de considerações sobre cada interface projetada. As seguintes informações foram levantadas e documentadas para cada modelo:

- *Descrição* – uma breve apresentação da tela, descrevendo o objetivo central da existência da interface, relacionando necessidades e requisitos do sistema.
- *Acionamentos* – enumeração de formas de navegar até a interface, seja por meio de ação do usuário em outra tela do sistema, ou por redirecionamento automático do sistema em razão de uma situação específica.
- *Links* – navegação da interface para outras interfaces, inclusive, também, comentando onde o link se encontra na interface, e se é ação do usuário ou redirecionamento do sistema. *Links* de uma interface compõem acionamentos das demais telas, assim como o inverso.
- *Observações* – informações adicionais, que foram necessárias de serem explicitadas, para correto entendimento sobre determinado aspecto da interação.

Considerando o fato de que o projeto é de aplicativo móvel, e que o usuário poderá, frequentemente, estar em movimento, ou locais conturbados, ou que, ainda, precise acionar rapidamente algumas opções principais do sistema, um *menu* de opções foi definido no rodapé da maioria das interfaces. Assim o usuário conseguirá, com apenas um comando, ter acesso às interfaces que lhe ajudariam nessas situações. Além disso, este menu serve como ponto de partida para tudo no sistema, para as outras interfaces que são acessadas pelo usuário. Assim, este *menu* se torna importantíssimo, o que precisou ser bem planejado para equilibrar as profundidades de caminhos entre o estado inicial da aplicação e as interfaces que não estão disponíveis diretamente neste menu. As opções do *menu* são:

- *Perfil de usuário* – útil para o usuário visualizar rapidamente seus dados, bem como os de serviço que presta e que já usufruiu, além de ser um atalho rápido para a interface que é considerada como a “tela inicial” do aplicativo – acionada após *login* de usuários.
- *Mapa de serviços* – leva para interface de mesmo nome. Esta é a principal tela de interesse para consumidores (observando que consumidor pode ser qualquer usuário, na verdade). Para um buscar um serviço, basta abrir a aplicação e acionar esta opção. Assim, rapidamente, o usuário dispõe de serviços a contratar.
- *Chat* – tela que lista as conversas em aberto do usuário com outros usuários. O usuário precisa saber quem está se comunicando com ele maneira eficiente, fazendo jus ao conceito de mensagens instantâneas. Logo, esta opção ficou disponível neste menu principal.

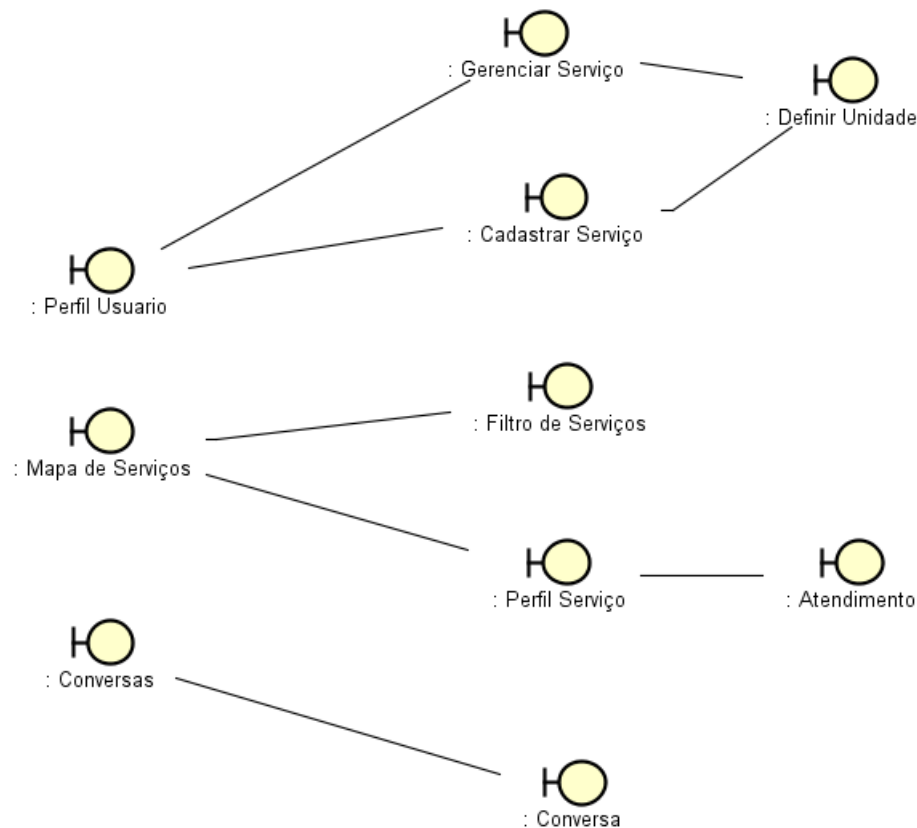
<b>Tela Perfil de pessoa</b>	
Descrição	Tela que contem o resumo do cadastro do usuário, além de uma lista de serviços que ele publicou na plataforma. É Considerada a “tela inicial” do app.
Acionamentos	<ul style="list-style-type: none"> <li>• Tela Login (Redirecionamento)</li> <li>• Botão “Perfil” no menu <i>tabs</i> inferior</li> <li>• Tela Atendimento (Redirecionamento)</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Mapa – botão “mapa” no menu <i>tabs</i> inferior.</li> <li>• Tela Conversas– botão “chat” no menu <i>tabs</i> inferior.</li> <li>• Tela cadastrar serviço - Botão adicionar serviço.</li> <li>• Tela gerenciar serviço – Botão gerenciar serviço</li> </ul>
Observações	

Figura 22 – Quadro de especificação da interface Perfil de pessoa.



**Figura 23 – Protótipo da interface Perfil de usuário, demonstrando menu de acesso rápido.**

Estas três opções no menu, logo, são inícios de trilhas de navegação. A Figura 24 mostra a navegabilidade geral entre as interfaces. Trata-se de um diagrama de comunicação, mas que também pode ser compreendido como um grafo direcionado, no qual as classes (interfaces) são vértices, e um link de uma interface a outra é um arco, com sua origem na extremidade esquerda, e o destino na direita. Como pode ser percebido, em termos de teoria de grafos, o caminho mais profundo possui tamanho igual à três – há um equilíbrio entre as profundidades, caminhos partidos do menu principal.



**Figura 24 – Diagrama de comunicação da navegabilidade entre interfaces do usuário.**

Para a definição visual das interfaces, assim como o comportamento do sistema mediante a ação do usuário sobre a interface, uma reflexão foi feita baseada na maneira como os usuários da aplicação poderiam interagir de maneira natural, semelhante como é na vida real. As sequências de passos para realizar as principais atividades foram definidas da seguinte forma:

- **Buscar serviços** – o usuário precisa buscar um serviço. O usuário entra no aplicativo, acessa alguma opção de busca por serviços, informa os parâmetros da pesquisa, visualiza os resultados e decide acionar ou não algum prestador de serviço.
- **Atender consumidor** – o usuário prestador de serviço, a qualquer momento, recebe uma notificação do aplicativo, mesmo que o usuário não esteja utilizando a aplicação no momento. O usuário percebe a notificação, toca na mensagem, o aplicativo abre e mostra as informações do consumidor. O

prestador decide atender ou não o consumidor acionando o botão correspondente.

Uma breve revisão a ser considerada, no projeto da interface de Mapa de serviços, é a de destacar, e perguntar logo ao usuário, o tipo de serviço que ele quer buscar. Essa reflexão é baseada na premissa de que, quase sempre, o usuário, ao pensar em buscar um serviço, pensa automaticamente em um tipo de serviço – ou melhor, a necessidade de buscar o serviço nasce junto à um tipo de serviço específico. Logo, considera-se como melhoria, na interface de busca por serviços, adicionar um campo acima do mapa de serviços, para que seja informado facilmente o tipo do serviço a ser disposto no mapa.

No atendimento do serviço, a interação entre os usuários prestador e consumidor torna a tela bastante dinâmica. Os dois usuários precisam interagir em conjunto para alcançarem seus objetivos e, enquanto um espera ação do outro, o sistema mantém o que aguarda a par da espera de ação do outro usuário, inclusive que ação é esta. Mais detalhes sobre esta interação poderão ser conferidos no próximo capítulo, *Desenvolvimento*, na subseção do desenvolvimento do aplicativo móvel.

## 7.5 PROJETO DE TESTES

Realizado o projeto de arquitetura, de componentes, de classes e interfaces, foi possível definir adequadamente um projeto de testes para a ferramenta.

Os testes de *software* são uma importante atividade no desenvolvimento de sistemas, uma vez que podem atestar de que a ferramenta em desenvolvimento está sendo implementada de maneira consistente, refletindo os requisitos funcionais e não funcionais do projeto. De acordo com Pressman e Maxim (2016), os testes objetivam encontrar o maior número possível de erros. Ainda, por eles, são documentados testes de software, dentre os quais são, aqui, destacados:

- Teste de unidade – testes para pequenas partes da implementação de uma solução em sistemas. Em projetos orientados a objetos, unidade é equivalente a cada método

disponível em cada classe do sistema. É feito um teste, portanto, para saber se determinado método cumpre o que promete (sua assinatura e especificação no modelo de projeto).

- Teste de integração – técnica útil para realizar testes de interação entre componentes de arquitetura do sistema. Com este teste é possível verificar se as interfaces providas pelos componentes estão realizando o papel pretendido.
- Teste de validação – para verificar se o produto funciona como esperado pelo cliente, pelos usuários. Ou seja, o teste de validação pode recorrer à especificação de requisitos definida anteriormente para constatar que o produto está condizente com os requisitos funcionais da aplicação.
- Teste de sistema – teste para verificar se o *software* funciona como se espera quando inserido em um contexto externo, como em uma instalação em um servidor, e passa a ser utilizado. Aqui podem ser feitos teste de estresse, de desempenho, dentre outros.

As estratégias de teste levantadas para o projeto considerados relevantes para os objetivos do desenvolvimento da primeira versão são: teste de unidade, de integração, validação e de sistema. O quadro 6 exhibe o planejamento para cada tipo de teste projetado.

**Quadro 6 – Esquema de testes de software projetados**

Teste de Unitário	Realização dos testes de funcionalidades esperadas de cada componente (unidade) do sistema, executando os métodos das interfaces dos componentes sobre diferentes aspectos, e verificando a resposta, se é de acordo com o que se espera.
Teste de Integração	Realizar testes de integração entre os componentes, se a comunicação definida entre cada par de componente que necessite se

	comunicar, é realizada de maneira correta, de acordo com o esperado.
Teste de validação	O sistema deve ser submetido aos interessados pela aplicação, para atestar de que os requisitos foram atendidos, com o sistema já pronto, ou partes dele.
Teste de sistema	Os seguintes testes de sistema devem ser considerados: <ul style="list-style-type: none"> <li>• Teste de recuperação.</li> <li>• Teste de segurança.</li> <li>• Teste de estresse.</li> </ul>

Alguns tipos de teste não foram incluídos no projeto de teste, pelo menos por enquanto. É importante esclarecer que, de acordo com os objetivos e o que foi planejado para este projeto, até o momento, é implementar um produto minimamente viável, de forma a concretizar uma primeira versão do produto capaz de ser demonstrada, verificada. E isso torna alguns testes desnecessários, no momento, já que o projeto ainda está distante de ter uma versão considerada suficiente para entrar no mercado, entrar em produção. Portanto, quando os tipos de testes ausentes neste projeto serão acrescentados e planejados no modelo de projeto quando se julgar necessário, em novas etapas do desenvolvimento.

Assim, o modelo de projeto foi definido, considerando os quesitos necessários a um desenvolvimento orientado a este modelo, de tal forma que a qualidade do aplicativo (a fidelidade aos requisitos) seja garantida. No próximo capítulo o desenvolvimento – implementação – do sistema é elucidado, expondo as tecnologias e ferramentas empregadas e os detalhes de implementação de todo o projeto, bem como os testes de software trabalhados, relacionando com o modelo de projeto definido.

## 8 DESENVOLVIMENTO

Neste capítulo são explanados os principais pontos da etapa de implementação da ferramenta, bem como a realização de testes. É relatado que requisitos foram implementados, se parciais ou completos, a priorização e o adiamento de implementações de cada um deles.

As ferramentas e as linguagens de programação que foram utilizadas são comentadas, bem como padrões de projeto e paradigmas. Ao fim, são feitas considerações a respeito do prosseguimento da implementação da ferramenta.

### 8.1 FERRAMENTAS E LINGAGUENS

O mercado de linguagens de programação vem mudando consideravelmente nos últimos anos. Hoje existe uma diversidade de linguagens considerável e ferramentas diferentes, com focos e metodologias semelhantes, permitindo às equipes de desenvolvimento escolher a que melhor se adequar ao projeto, seja por motivos técnicos ou estratégicos – como o fato de a equipe de desenvolvedores disponível ter experiência com determinada tecnologia.

Entretanto, esse crescimento divide a comunidade da área, criando um cenário de pequenas e enormes bolhas de adeptos a linguagens e padrões. Se por um lado existem mais opções para trabalhar, por outro a decisão de qual ferramentas empregar se torna uma tarefa mais árdua.

Para este projeto, foram considerados os seguintes aspectos, na ordem disposta, para a escolha das ferramentas empregadas:

- Afinidade com as restrições de software, requisitos não funcionais e funcionais.
- Experiência do desenvolvedor do trabalho.
- Integração entre linguagens que forem empregadas
- Comunidade de desenvolvedores relativamente abrangente para apoio e suporte no desenvolvimento.

Com estes critérios, chegou-se à escolha de cada tecnologia para utilizar no projeto. Nesta seção são abordadas as tecnologias de computação e software empregadas para a implementação, assim como a implantação.

### 8.1.1 Javascript

Conhecida até poucos anos como a como a linguagem de programação interpretada que roda em navegadores *web*, hoje se transformou em algo a mais que isso. Esta linguagem foi portada para funcionar também em aplicações servidoras, por meio de máquinas virtuais de interpretação da linguagem que foram desenvolvidas, como o NodeJS.

A linguagem, antes, tinha o propósito de dar vida às páginas *web*, colocando funcionamento e conteúdo dinâmico no lado do cliente, manipulando a estrutura da página de forma a melhorar a experiência dos usuários, respondendo a interações dos usuários e realizando atividades como abrir uma janela flutuante, gerar uma lista de registros em tabela HTML, enfim.

Com a adoção da linguagem no lado servidor, em pouco tempo de amadurecimento da ideia, surgiram profissionais e até grandes empresas que adotaram o desenvolvimento de softwares completos com esta linguagem (CAMPOS, 2016) . E isto é justificável. Capan (2016) relata as seguintes vantagens de se adotar a linguagem para referidos fins:

- Por ser *javascript*, para muitos projetos isso pode significar a utilização da mesma linguagem para as camadas servidor e cliente. Ou seja, necessidade de aprendizado de uma única linguagem, capaz de trabalhar tanto no lado cliente como servidor (e até no banco de dados com os do tipo orientado a documentos como o *MongoDB*).
- Escalável para grande número de clientes conectados em um servidor com esta tecnologia.
- Linguagem não bloqueante e de chamadas assíncronas. Ideal para aplicações que requerem grandes quantidades de operações de leitura e gravação.

Além disso, a comunicação do tipo *push* é facilitada nesta plataforma, na qual o servidor envia mensagens aos clientes também, além da habitual comunicação do cliente para o servidor (BEVERLOO, THOMSON, *et al.*, 2017). Facilidade justificada pelo fato da arquitetura ser não bloqueante.

Levando em consideração estes critérios, e as linguagens de programação que o desenvolvedor do trabalho considerava ter domínio mínimo para desenvolver um sistema como o *iservice*, esta linguagem foi escolhida. Dentre os aspectos considerados, destaca-se o de eficiência em comunicação de tempo real, essencial para esta aplicação.

### 8.1.2 Máquina Virtual *NodeJS*

Uma vez definida a linguagem de programação *javascript* para trabalhar também no servidor, este uso requer uma máquina virtual que execute *javascript*. *NodeJS* é a principal do mercado, com uma grande comunidade de desenvolvedores participantes do desenvolvimento e manutenção da máquina, e também desenvolvedores de software que estão utilizando a ferramenta.

### 8.1.3 Aplicativos móveis nativos e híbridos

O desenvolvimento de aplicativos móveis, em relação à implementação do programa móvel, pode ser construído por meio de dois mecanismos: nativo ou híbrido. A diferença está no próprio nome, ou seja, nativos são aqueles desenvolvidos com a linguagem especificada para determinado sistema operacional móvel (Android, iOS). Já o híbrido mescla as tecnologias da linguagem nativa e *web*, possibilitando a implementação do app multiplataforma, no qual apenas um código fonte desenvolvido é utilizado para compilar em várias plataformas móveis.

Dito isto, entende-se que o desenvolvimento de aplicativos híbridos é mais rápido e barato, pois, ao contrário do nativo, não requer um desenvolvimento, do zero, para cada sistema móvel.

Entretanto, é importante dizer que o desenvolvimento nativo é melhor em relação aos aspectos de segurança, eficiência no processamento, e dispõe dos recursos nativos da maneira natural, como foi projetado pelos fabricantes das plataformas móveis. Para projetos em que o orçamento pode ser suficiente

para a implementação nativa, considera-se que esta seja a melhor escolha. Em outras palavras, o desenvolvimento híbrido de aplicativos móveis é uma solução ágil e econômica, ou para projetos que não exijam grande capacidade de processamento, ou uma saída para implementação de protótipo de um projeto.

Como este trabalho pode ser considerado uma startup, ainda sem recursos de possíveis investidores, o desenvolvimento foi feito com a metodologia híbrida, possibilitando o resultado mais rápido e econômico.

Entretanto, considera-se a possibilidade de à medida que o projeto ganhe recursos financeiros, seja pelo sucesso de uso considerável, seja por apoio de investidores, o desenvolvimento nativo seja iniciado para compor uma nova versão mais completa e robusta da solução.

#### 8.1.4 Ionic Framework

Uma vez decidido o desenvolvimento do app de forma híbrida, ferramentas que possibilitem este desenvolvimento foram buscadas.

Neste contexto de *frameworks* híbridos, há o destaque para o *Cordova*, que foi um projeto iniciado pela *Adobe*, mas posteriormente doado para a *Apache Foundation* (LINO, 2015).

A ferramenta *Apache Cordova* é capaz de gerar um aplicativo híbrido apenas fornecendo um projeto web, com arquivos *HTML*, *CSS*, *javascript*. Quer dizer, um projeto web que poderia ser executado em um navegador, pode se transformar em um *app* móvel.

Neste cenário, existem *frameworks* que adicionam uma camada a mais de abstração para esse desenvolvimento, fazendo uso do *Cordova* para gerar o *app* e disponibilizando recursos extras, facilitadores do desenvolvimento híbrido para os desenvolvedores. Uma destas ferramentas de abstração é o *Ionic framework*, que é largamente utilizado pela comunidade deste ramo, e possui uma documentação rica em detalhes. Esta ferramenta, logo, foi adotada para o projeto, por pesquisas sobre seu sucesso no mercado de ferramentas do gênero, pelas facilidades específicas que ela provê para a construção de um aplicativo.

A base de desenvolvimento híbrido, com *Apache Cordova* e ferramentas que o utilizam como o *Ionic*, é com a linguagem *javascript* - neste momento ocorre a coincidência das linguagens de programação dos subsistemas API (servidor) e App(cliente).

#### 8.1.5 Banco de Dados MongoDB

*MongoDB* é um sistema de gerenciamento de banco de dados (SGBD) não relacional, orientado a documentos. Bancos de dados deste gênero não possuem tabelas, mas coleção de documentos.

Este tipo de banco de dados possui como vantagem a escalabilidade melhor em relação aos bancos de dados relacionais. Tal vantagem é explicada por Steppat (2009) pelo fato de que os não relacionais não trabalham com a estrutura de dados fortemente definida. Não há uma rigorosa validação de consistência dos dados a serem armazenados, tornando o custo de processamento para persistência de informações ligeiramente mais escalável.

O projeto *iservice* possui a premissa de ganhar um público alvo de grandes escalas, com muitas buscas por serviços. Além disso, um tipo de pesquisa de extrema relevância no projeto é o de aproximação geográfica. O *mongodb*, em particular, possui ótimas funções nativas que possibilitam uma busca eficaz de informações de localização.

Para concluir, este SGBD, assim como vários do mesmo gênero, utiliza estrutura de dados *JSON (JavaScript Objeta Notation)* e comandos *javascript*. Assim, são estabelecidas as três partes do sistema para serem implementadas e trabalha-se na mesma linguagem base.

## 8.2 IMPLEMENTAÇÃO DA API

O desenvolvimento, implementação, do projeto iniciou-se pelo subsistema API, que compete ao lado servidor da aplicação. Inicialmente foram definidos os modelos de dados, baseados no diagrama de classes e no dicionário de dados levantados na etapa de análise de sistemas. Por seguinte foram implementadas as rotas de comunicação - camada *controller* - com o subsistema *App*, e, por fim, definido o componente de comunicador de usuários.

### 8.2.1 Modelos de dados

O ponto de partida da implementação do projeto como um todo foi implementar objetos configuração dos modelos de dados, de forma a refletir o diagrama de classes, assim como os detalhes contidos no dicionário de dados, como o tamanho de cada campo, a obrigatoriedade e a relação entre os objetos.

Por estarmos trabalhando com bancos de dados não relacional, os quais não definem regras sobre a padronização de registros de dada coleção, foi utilizado a ferramenta *mongoose*, uma middleware de comunicações com o banco *MongoDB*, para definir uma interface de dados a serem persistidos. Por meio desta ferramenta, é possível, portanto, definir um esquema de dados para cada coleção de documentos a serem persistido, sem que isso signifique reduzir a escalabilidade da persistência. Estes objetos de configuração dos esquemas podem são equivalentes a classes entidades, classes PO(*Persistent Object*).

Os dados, que são armazenados no banco de dados, como mencionado, são armazenados em formato JSON. A Figura 24 mostra um exemplo de consulta de dados no repositório do MongoDB, exibindo de maneira amigável os objetos nessa estrutura de dados.

```
{
  "_id" : ObjectId("59b5f19c83b0f21520b9b73a"),
  "local" : {
    "email" : "bira.simoes.miranda@gmail.com",
    "password" : "$2a$08$KLue3nNPeS6kpEXVzy3SU.cEIuwr5BxmvIqvMw4VWVl3DLtI5GbTO"
  },
  "servicos" : null,
  "interesses" : [
    "Musica",
    "Boteco",
    "Cerveja",
    "Feira",
    "Baile da Saudade",
    "Sertanejo"
  ],
  "contato" : {
    "telefone" : "9199998888",
    "email" : "bira.simoes.miranda@gmail.com"
  },
  "status" : "ATIVO",
  "fisica" : {
    "nome" : "Bira Simões",
    "nascimento" : ISODate("1991-01-01T00:00:00.000Z"),
    "cpf" : "1234567890"
  },
  "__v" : 0
}
```

Figura 25 - Consulta de registro do banco de dados.

### 8.2.2 Controladores

A camada *controller* foi desenvolvida em seguida para prosseguir o desenvolvimento. Esta precisou ser muito bem definida, uma vez que é através dela que toda comunicação entre a API e os clientes, e entre clientes, ocorrerá.

Foram levantadas as necessidades de comunicação, de forma a dispor as corretas portas de entrada e saída de informação no servidor. Nem toda ação entre os dados de uma coleção existirá na aplicação. Um Exemplo é a operação de exclusão de dado registro do banco de dados, não haverá essa possibilidade, necessidade. Outra: não haverá cadastro e alteração de tipos de serviços, estes serão gerenciados pelos administradores do sistema e manipulados em ambiente de desenvolvimento, não necessitando, portando, definir uma interface de comunicação com os clientes para tal operação, assim como já foi especificado no modelo de projetos.

As operações de controle são parte da implementação dos seguintes componentes do modelo de projeto: *servicoCrud*, *unidadeCrud*, *pessoaCrud*, *comunicador de usuários* – mais especificamente as interfaces *web* destes componentes.

Nesta camada de controle, são utilizados dois protocolos de rede: *HTTP* e *websocket*, sendo o segundo utilizado no comunicador de usuários e o primeiro nos demais componentes citados.

Para os componentes trabalhados com *HTTP*, foi seguido parte do padrão de comunicações de aplicações em sistemas distribuídos *RESTful*. As principais características deste padrão, muito bem resumidas por MONTEIRO (2015), são:

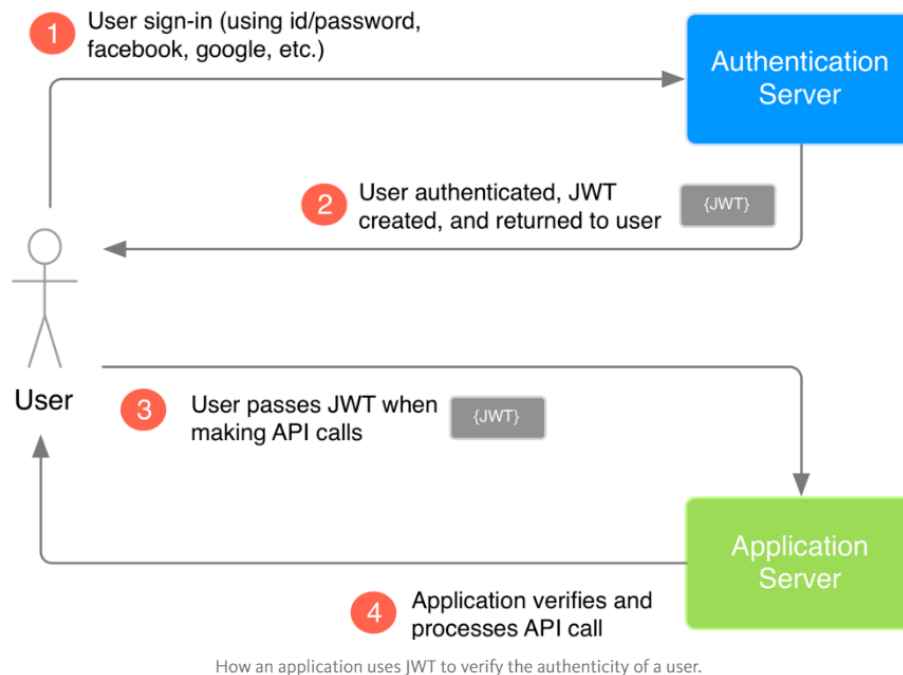
- Baseado em recursos (Resource Based) – URI de acesso.
- Manipulação de recursos utilizando representações (*Manipulation of Resources Through Representations*) – tipo de ação sobre os dados a depende do verbo *HTTP* da mensagem.
- Mensagens autodescritivas – cada mensagem é autossuficiente de informações necessárias para seu processamento no recebimento.

Para esta camada, é importante citar, que as implementações de quesitos de segurança da informação foram estabelecidas. Praticamente todos os recursos que precisam estar disponibilizados aos clientes da aplicação, necessitam de autenticação de usuário, exceto as de cadastro de usuário e a própria requisição de autenticação. O estilo de autenticação de usuários escolhido foi por meio da técnica de *JSON Web Token*(JWT).

Este estilo funciona da seguinte forma:

- Um usuário, ao realizar seu *login*, recebe como resposta da autenticação, quando bem-sucedida, um texto sem significado relativo – valor codificado.
- Este conteúdo da resposta é, na verdade, uma representação em JSON, de alguns ou todos os dados do usuário, codificados.
- O cliente possui a responsabilidade de armazenar sua chave (*token*) com segurança, pois o servidor não armazena *token* nem sessões de usuários que se autenticaram.
- Sempre que o cliente precisar se comunicar com servidor, e esta comunicação for para uma rota que exige autenticação, o cliente envia no cabeçalho da requisição sua chave de acesso.

A Figura 26 exhibe este fluxo demonstrado por Stecky (2016).



**Figura 26 – como uma aplicação usa JWT para verificar a autenticidade de um usuário. Stecky(2016)**

Esta forma de autenticar usuários é utilizada por grandes empresas de softwares, como *facebook* e *google*. Ela pode ser utilizada como um meio para garantir a confiança entre um aplicativo e os usuários (STECKY, 2016).

### 8.2.3 Conexão WebSocket para respostas em Tempo Real

O componente *Comunicador de usuários*, especificado na etapa de projeto de sistema, foi implementado utilizando a tecnologia *websocket*.

Trata-se de um protocolo de comunicação em redes com canal de comunicação em dois sentidos (*two-way communication*), o qual possibilita servidores enviarem mensagens a um cliente, mesmo este tendo feito nenhum pedido anteriormente (FETTE e MELNIKOV, 2011). Quer dizer, pode ser considerado, fazendo uma analogia ao padrão de comunicação cliente servidor, que o servidor assume um papel de cliente, e o cliente assume um papel de servidor, de forma a tornar possível esta comunicação inversa, no sentido apenas de envio de mensagem (e não de tratamento e resposta). Assim, é possível fornecer atualizações sobre informações de negócio da aplicação a todos os clientes conectados.

Este mecanismo de comunicação é ótimo para implementação de conversas entre usuários (*chat*), notificações de modo geral ao usuário, sem que, para isso, o cliente precise ficar realizando verificações, junto ao servidor, se há mensagens novas de interesse para serem puxadas, do servidor para o cliente. Com isso, tem-se naturalmente o ganho de performance, por não precisar realizar uma sobrecarga de rede para tais verificações desnecessárias.

### 8.3 IMPLEMENTAÇÃO DO APLICATIVO MÓVEL

Para a implementação do aplicativo móvel, o passo inicial foi definir as classes de modelo também no lado cliente, para estruturar melhor as informações que serão manipuladas nas interfaces e consumidas da API. As definições dos atributos dessas classes são exatamente iguais aos modelos de dados definidos no servidor. Assim temos a mesma representação em classes, e instâncias em objetos nos dois subsistemas.

Por seguinte foi realizado o desenvolvimento do cadastro de usuários – onde tudo realmente começa, no fluxo de uso da aplicação. Nesta interface do sistema, procurou-se, desde já, estabelecer critérios para preenchimento dos formulários do aplicativo. A primeira medida foi validar campos obrigatórios, de forma que só seja possível submeter as informações após correto preenchimento destes dados. A Figura 27 exibe o cadastro de pessoa com estas regras sendo aplicadas.

← IService - Cadastro

Contato

Email

Campo Obrigatório

Telefone

999132323

Login

Email

teste

Endereço de Email Inválido

Senha

Campo Obrigatório

Confirmação de Senha

\*\*\*\*\*

Senhas Não conferem

Figura 27 – Formulário de cadastro de pessoa com regras de validação aplicada aos campos.

Dando prosseguimento, após a definição efetiva do cadastro de pessoas pelo aplicativo, o segundo passo foi definir o processo de autenticação. Inicialmente foi implementado a autenticação local, ou seja, aquela em que o usuário informa o seu e-mail e senha cadastrados. Aqui, define-se o que faltava da implementação do componente de projeto *autenticação de usuários*, o qual foi iniciado ao definir o cadastro de pessoas.

No processo de autenticação de usuários, vale lembrar, que o cliente precisa armazenar sua chave de acesso ao realizar o processo de *login*, já que o servidor não irá armazenar sessões de usuários, o que é trivial. Esta chave pode ser persistida no dispositivo móvel em um banco de dados local do aplicativo, recuperando esta chave sempre que necessitar realizar uma requisição a API. Para otimizar, inclusive, esta chave é armazenada na memória principal, uma única vez, e sempre recuperada deste local, enquanto o aplicativo

estiver ativo, evitando sobrecarga de leituras em disco – do banco de dados do dispositivo. A Figura 28 ilustra parcialmente uma ocorrência de sucesso de autenticação da API, contendo, no corpo da resposta, um objeto JSON com dados do usuário autenticado e sua chave de acesso (JWT).

```
{
  "token": "JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfYWQ0i0iI10WI5ZjVhYjk5NDE2NTAwMDQzMWY3MTg1LCJpbnRlcmVzc2VzIjpbIk11c2ljYSIsIiBCb3RlY28iLCIgQmF0dXFlZSIsIiBGbG9yaWN1bHR1cmEiLCIgUGl6emEiLCIgW9nYSIsIiBSZWdnYWUiLCIgQ2luZW1hIiwiaWF0IjoxNTA1MzYwNzI5LCJleHAiOjE1MDUzNzA4MDI9.a-eCyU48IKbBDVA9G5R7Zb8wSjvGNm0FC4Kp5AOr_R0",
  "user": {
    "_id": "59b9f5ab994165000431f718",
    "_v": 0,
    "local": {
      "email": "ewertom.moraes@gmail.com",
      "password": "$2a$08$0sZNAiTx6HSfiem0NXC02.zlqOX8E0wY6Th/7puKkH7gmDXzaPTDm"
    }
  },
  "servicos": null,
}
```

Figura 28 – Resposta da API quando autenticação de usuário é bem sucedida

Com o ingresso do usuário devidamente implementado, foi dado início à implementação da interface *perfil de pessoa*. Esta tela é o ponto de partida de qualquer usuário do aplicativo, após realizar seu *login*, ou quando da autenticação já efetivada uma vez, ao abrir a ferramenta.

Nesta tela são exibidos os dados do usuário, os serviços que ele publicou, e os menu principal, na parte inferior.

Por meio desta tela, também, é possível chegar ao cadastro de serviço do usuário. Esta interface definida no projeto de interfaces propõe o cadastro em uma tela única, com todos os campos de serviços e de unidades a serem adicionadas. Porém, durante o desenvolvimento, foi proposto o que se considera melhoria neste processo de interação com o usuário neste cadastro: formulário em forma de *wizard*, ou seja, o formulário ser estruturado de forma segregada e ordenada, guiando o usuário em uma sequência lógica e esperada de informação dos dados para um cadastro de serviço correto e com tudo que seja necessário.

Para o formulário deste cadastro, antes do cadastro de fato iniciar, é perguntado ao usuário qual tipo de serviço ele quer cadastrar. Após esta pergunta é colocada a primeira etapa do cadastro, de fato, onde o usuário irá

decidir qual tipo de unidade de atendimento ele deseja cadastrar inicialmente para o serviço. Isso porque todo serviço precisa de ao menos uma unidade de atendimento associada. E, no cadastro passo a passo isto fica transparente ao usuário, uma vez que são feitas perguntas a ele, de forma que não fica claro que ele está fazendo mais de um tipo de cadastro ao mesmo tempo – serviço e unidade – fica com a sensação de ser um cadastro único. A Figura 28 mostra a primeira pergunta ao usuário mencionada.



**Figura 29 – Primeira etapa do cadastro de serviço.**

Ao passar da primeira etapa do cadastro de serviço, o usuário recebe a segunda etapa na tela, que é o cadastro de unidade já devidamente direcionado para o tipo de unidade que ele informou, ao responder a pergunta anterior (Figura 29). Caso tenha escolhido que é uma sala ou ponto fixo, uma opção a mais para informar localização geográfica é exibida para ele neste segundo passo.

Cadastro de Unidade

Responsável

Eu mesmo

Outra Pessoa

Localização

Móvel - onde eu for, divulgar minha localização

Local ou Ponto Fixo

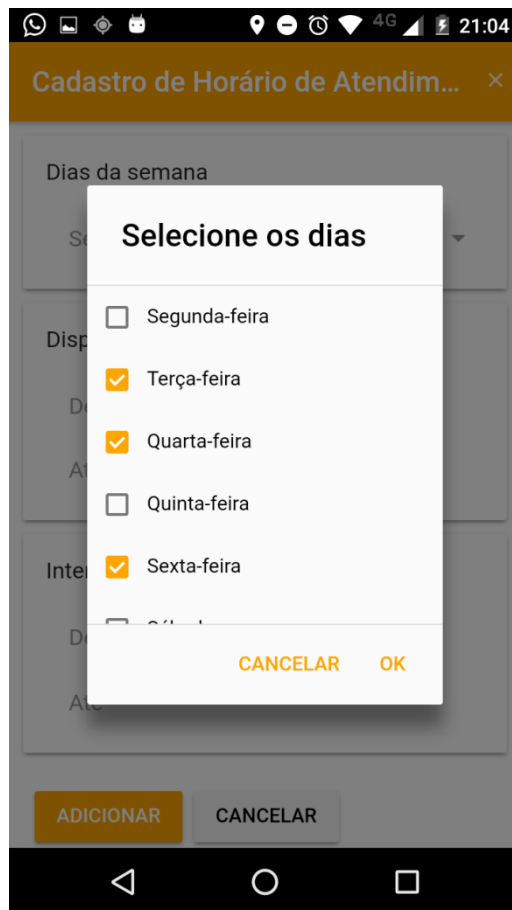
Localização

Pesquise um endereço...

Nome da Unidade

**Figura 30 – Interface Definir unidade**

Ainda, no cadastro de unidade, o cadastramento de horário de atendimento da unidade foi implementado seguindo detalhes especificados. O usuário informa um ou mais dias da semana, horários de atendimento e de pausa (Figura 31).



**Figura 31 – cadastro de horário de atendimento, no momento da seleção dos dias da semana.**

Após informar os dados da unidade, como o local, horário de atendimento, responsável pela unidade, o passo seguinte é informar dados do serviço como a faixa de preços e a descrição textual pessoal do prestador sobre seu anúncio (Figura 32). Após esses dados serem informados, o cadastro é efetivado.

Serviço

Tipo Lanches e ...

Preço mínimo  
2,00

Preço máximo  
15,00

Descrição  
Sirvo lanches de diversos tipos.  
Sempre estou aqui na praça da Bíblia,  
durante a noite. Temos sucos e  
refrigerantes também.

O serviço pode ser contratado  
pelo iservice?  
Os clientes irão pagar por  
cartão de crédito

3 / 3

Figura 32 – Terceiro passo do cadastro de serviço.

Uma parte de destaque na implementação do produto foi o componente de mapa. Este possui características interessantes e mais aprofundadas como a consideração do contexto do usuário (agir de acordo com a localização da pessoa), a troca de mensagens instantâneas de geolocalização entre usuários (por *websocket*) e a integração com recursos nativos do dispositivo(GPS).

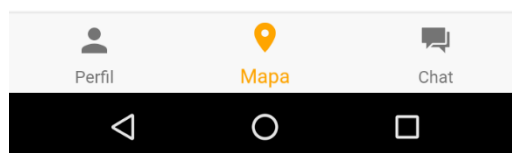
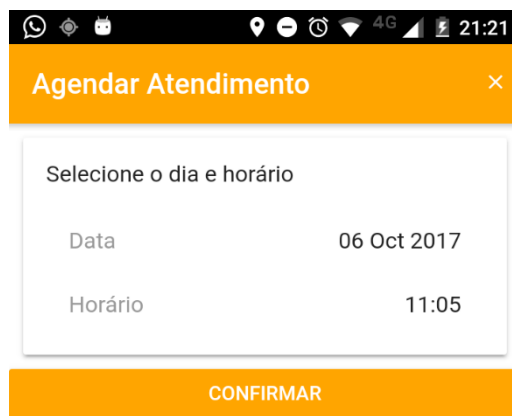
Para exibição de um mapa ao usuário, a ferramenta empregada foi a da *Google – maps*. Trata-se de uma API pública para ser utilizada em aplicações como o *iservice* para exibição de um mapa geográfico de praticamente toda a superfície terrestre. Nesta ferramenta da empresa, uma das formas de dispor o mapa é visualizando os contornos das ruas com o nome delas, nomes de lugares, de distritos oficiais – bairros, municípios, estados. Toda esta visualização está disponível na ferramenta e considerou-se perfeito adotar ela para compor a visualização do mapa da aplicação em construção.

Sobre o mapa, então, são plotados pontos (ou marcadores), que sinalizam a localização de unidades de serviços, tanto móveis como fixas. Essas unidades são apresentadas em duas cores: verde (unidades de atendimento disponíveis) e amarelo (unidades não disponíveis).

Ainda, associado ao mapa, existe a interface filtros de serviços, a qual, como projetada anteriormente, deve dispor opções de filtro que, quando aplicadas, realizam uma requisição para a API, em busca de serviços com os critérios de filtro empregados. Este resultado então é aplicado à tela de mapa, a qual é aberta para o usuário, agora com somente os resultados filtrados disponíveis.

Quando o usuário aciona, no mapa, uma unidade de atendimento, é exibida a interface *perfil do serviço*. Nela é possível acionar a contratação do serviço, agendar atendimento futuro e entrar em contato com o prestador responsável pela unidade. Entretanto, vale lembrar que essas opções, de acordo com a especificação, são dinâmicas por tipo de serviço e outras condições como o horário de atendimento. Assim, dependendo das condições de cada situação, ao acionar o perfil do serviço, as opções são dispostas ou não.

Quando a opção de agendamento acionada, é exibido uma tela extra para informar dia e horários desejados pelo consumidor. A Figura 33 mostra este diálogo com as perguntas de agendamento.



**Figura 33 – Diálogo para agendar atendimento.**

Ao acionar a opção contratar, é então exibida a interface de atendimento. Para desenvolvimento desta funcionalidade, foi necessário, assim como no mapa, utilizar o recurso de *websocket* para contemplar o funcionamento esperado do componente de projeto de atendimento. Como projetado, este componente interage com o de *comunicador de usuários*, da API, para realizar a interação entre prestador de serviço e consumidor por meio de notificações de um para o outro, passando pelo servidor, de igual forma feito para o compartilhamento de localização em tempo real no para atualizar o mapa de serviços.

Logo, no atendimento foi implementado todo o processo de interação, de tal forma que um usuário espere a ação do outro para continuar a dar prosseguimento ao atendimento. Esta comunicação entre usuários funciona exatamente como ilustrado no diagrama de atividades de atendimento, disponível na seção 6.2. Aqui, a Figura 34 exibe a implementação desta interface

de usuário do aplicativo, em um momento exato que o consumidor envolvido no atendimento aguarda o prestador de serviço confirmar o encontro.



**Figura 34 – Andamento do atendimento. Consumidor aguardando confirmação de encontro pelo prestador.**

E, para o componente *chat*, a troca de mensagens instantâneas também foi implementada seguindo o protocolo de comunicação *websocket*, já que este é a base de implementação do componente *comunicador de usuários*, da API, que é dependência do componente *chat*, do aplicativo. A interface de usuário, aqui, foi das mais simples. A primeira tela exibe a lista de conversas em aberto que o usuário iniciou com demais usuários, e a interface de conversa, Figura 35, exibe a lista de mensagens já trocadas entre os dois usuários.



Figura 35 – Interface Conversa

E, para concluir, para atender à comunicação entre as interfaces que exibem e cadastram dados de pessoa, serviços e unidade, o componente de projeto *conexão com modelos de dados* foi implementado de tal forma que conseguisse se comunicar com qualquer um dos controladores de modelos de dados da API, contemplando o projeto de componentes.

#### 8.4 TESTES DE SOFTWARE

Os testes de software realizados contemplaram o projeto de testes definido na etapa de projeto de sistemas. Foram implementados alguns dos testes de integração e também de aceitação da aplicação, até o momento deste trabalho.

#### 8.4.1 Testes Automatizados

Para implementação dos testes de integração (e futuramente dos testes unitários), técnicas de testes automatizados foram consideradas para este projeto.

Teste automatizado é o ato de criar um código fonte que possibilite realizar as checagens que os casos de teste necessitam, sem que qualquer verificação humana seja necessária. Isso quer dizer, uma vez construído o código responsável por chamar uma rotina a ser testada, passando os parâmetros corretos, e verificando a resposta – se é de acordo ou não com o esperado pelo projeto de testes – então este trabalho provavelmente nunca será refeito. Ou seja, faz com que o ato de uma pessoa realizar o teste manualmente, sempre, toda vez que precisar submeter o software à nova bateria de testes, seja dispensado, pois foi automatizado. E, quando necessário realizar alguma atualização no caso de teste do projeto, reflete isso no código responsável por testar.

Estes testes automatizados, por exemplo, podem ser executados automaticamente toda vez que um recurso novo for adicionado ao repositório de código-fonte do projeto. Se houver algum teste que falhou (não houve a resposta programada para ser a esperada), imediatamente a equipe de desenvolvedores saberá do ocorrido. Isso traz ao ambiente de desenvolvimento e do projeto, muita agilidade na descoberta de problemas em componentes da aplicação, no que se diz respeito aos testes unitários e de integração.

Um exemplo arbitrário de teste unitário automatizado, em pseudocódigo, pode ser conferido na Figura 36.

```

testa_se_atualiza_saldo(){
    saldo_atual = 1000;
    valor_saque = 200;

    conta.saldo = 1000;
    conta.realizaSaque(200);
    se(conta.saldo != (saldo_atual - valor_saque)){
        avisaTesteFalhou();
    }senão{
        avisaTesteSucesso();
    }
}

```

Figura 36 – exemplo de teste unitário automatizado

#### 8.4.2 Testando chamadas à API

Para os testes de deste projeto, foram iniciados, primeiramente, testes de integração que simulassem a comunicação entre componentes do *app* que necessitam se comunicar com os componentes da API. Isso quer dizer que são interfaces web, que a chamada de rotinas dos componentes alvo se dá através da web, com requisições HTTP e *websocket*.

A API deste projeto, assim como a maioria do gênero, possui uma visão macro sobre o que se espera ao utilizar suas funcionalidades. Uma API na web deve ser testada, principalmente, a sua integração, sua interface web, que possibilite inserir e consumir recursos dela.

Assim, foram construídos testes automatizados para realizar estes testes de chamadas à API, respeitando os casos de testes definidos no modelo de projeto de sistemas. A Figura 37 exhibe a implementação de um dos casos de teste de forma automatizada. Este teste verifica o caso em que, chamar o recurso de consultar os serviços de uma pessoa informada, do componente *serviço crud*, deve ocorrer com sucesso. Faz três checagens automatizadas, após receber a resposta da requisição: que não foram gerados erros na requisição (linha 5), que o *status http* da resposta foi igual a 200 (linha 6) e que deveria existir um corpo de resposta requisição válido (linha 7).

```
1  chai.request(app)
2    .get('/api/servico/pessoa/' + _idPessoa)
3    .set('Authorization', token)
4    .end(function (err, res) {
5      should.not.exist(err);
6      res.should.have.status(200);
7      should.exist(res.body);
8      done();
9    });
10  });
```

Figura 37 – Caso de teste automatizado para consultar serviços de uma pessoa.

#### 8.4.3 Testando a Interface de Usuário

O teste de interface do usuário foi definido para ser manual, com o uso de emuladores de dispositivos móveis e dispositivos reais. A interface dos usuários é testada tanto do ponto de vista funcional, quanto não funcionais, como critérios de facilidade de uso, de memorização, dentre outros.

### 8.5 CONSIDERAÇÕES SOBRE O DESENVOLVIMENTO DE SOFTWARE REALIZADO

O software, até momento presente deste trabalho, está em fase de desenvolvimento, não existindo, ainda, a completude de todos os requisitos. Para alguns requisitos considera-se que a implementação completada, até aqui, outros atendidos parcialmente, e outros que ainda não foram iniciados. A priorização de implementação das funcionalidades foi a de garantir um mínimo produto viável (MVP), capaz de poder ser submetido a testes funcionais e de usabilidade por pessoas que representem o público alvo do sistema, e para servir como protótipo a ser demonstrado para futuros investidores e demais pessoas interessadas em participar do projeto, do desenvolvimento e do uso dele.

No próximo capítulo será abordado um teste de usabilidade que foi realizado, utilizando o produto no estado em que foi reportado seu desenvolvimento até aqui, atendendo a requisitos básicos da aplicação, como o cadastro de pessoas, de serviços, a busca e contratação de serviços.

## 9 AVALIAÇÃO DE USABILIDADE

Para atestar que a ferramenta em construção possibilita uma boa experiência aos usuários, validando questões como facilidade de uso, memorização de como fazer atividades na aplicação, dentre outras, foi realizado uma avaliação de usabilidade de software, com parceria voluntária de algumas pessoas que representassem o público alvo.

Para realização do teste, neste capítulo, o objetivo do teste é estabelecido e a estratégia seguida detalhada. Posteriormente é relatado como o recrutamento dos usuários participantes foi feito, e quais os critérios foram levados em consideração. A aplicação da avaliação é discutida e, no fim, os resultados obtidos são analisados e interpretados para propor futuras melhorias no projeto de interfaces do usuário, além de outros requisitos não funcionais.

### 9.1 OBJETIVO

Para este projeto, o objetivo de realizar uma avaliação de usabilidade é saber, com uma versão mínima do produto, se ela seria boa e agradável de usar, mesmo estando ainda incompleta. É bom esclarecer que neste momento, em um estágio em que a ferramenta já está especificada e desenvolvida em parte, não é mais o objetivo descobrir se as pessoas estariam interessadas em utilizar a ferramenta, ou se ela é útil e necessária, de fato – esses questionamentos já foram realizados anteriormente com o público alvo. Agora, o objetivo é exclusivamente buscar a melhoria de usabilidade da aplicação, para levantar pontos de melhoria em pelo menos os seguintes critérios:

- Facilidade de aprendizado.
- Facilidade de memorização das atividades aprendidas.
- Facilidade de uso.
- Satisfação, se gera boa experiência, se é agradável.
- Comunicação entre o usuário e o sistema.
- Tempo de resposta ao usuário.

## 9.2 ESTRATÉGIA

Uma vez definidos os objetivos, foi planejado como o teste de usabilidade poderia ocorrer. Assim, foi elaborada uma estratégia, planejamento, a ser seguido para realizar os testes em tempo hábil, e que pudesse registrar por mais de um meio a interação entre o sistema e o usuário. Assim, ficou definido que a estratégia seria:

- Definir um roteiro de uso da aplicação.
- Definir considerações que devem ser passadas aos usuários, informando as limitações atuais do sistema, para evitar confusão entre o que pode ser um gargalo de usabilidade, e uma limitação do aplicativo por não atender, efetivamente, certa funcionalidade.
- Confeccionar um questionário para ser respondido pelos usuários logo após a avaliação ter sido executada por eles – a confecção do questionário está disponível na íntegra como apêndice E deste trabalho
- Controlar o tempo de uso do aplicativo cronometrado em 10 minutos para todos os usuários que irão participar, de forma a não dar condições diferentes de avaliação para os usuários participantes.
- Realizar a gravação da tela do dispositivo, sempre que possível, dos testes realizados, para ajudar na análise e interpretação dos resultados dos testes.

## 9.3 RECRUTAMENTO DE USUÁRIOS

Os usuários recrutados foram selecionados diante da disponibilidade de pessoas em participarem deste evento, além de levar em consideração critérios representativos do público-alvo da ferramenta. Os critérios considerados e suas justificativas foram os seguintes:

- Nível de experiência com tecnologia – deve haver pessoas experientes e não experientes em tecnologia, especificamente de *smartphones*. As pessoas com pouca experiência no uso de

aplicativos poderiam sinalizar os piores casos de interação, quanto a facilidade de aprendizado, por exemplo.

- Prestadores de serviços firmados – deve ter, entre os usuários, prestadores que já exerçam o seu ofício de prestador, que tenham como principal atividade a prestação de serviços que divulgariam na plataforma.

#### 9.4 APLICAÇÃO DA AVALIAÇÃO

Para a aplicabilidade dos testes, foi decidido realizar em locais que fossem confortáveis aos usuários, objetivando naturalidade no uso, sem que parecesse um ambiente controlado, reduzindo a liberdade e naturalidade da pessoa ao utilizar aplicação.

As aplicações das avaliações foram feitas baseadas nas etapas:

- Explicar a visão geral sobre o aplicativo
- Explicar que funcionalidades deveriam ser utilizadas.
- Esclarecer que os dados dentro da aplicação, relacionado ao cadastro de prestadores e seus serviços, são fictícios e estão ali para servir apenas como demonstração. Que o usuário não leve em consideração o conteúdo destes dados, para não atrapalhar os objetivos do teste.
- Disponibilizar o aplicativo, quando possível, no dispositivo do próprio usuário, para aumentar a naturalidade, deixando a pessoa usar seu aparelho, a qual está acostumada.
- Utilizar a gravação de vídeo da tela, quando possível.
- Cronometrar o uso em exatamente dez minutos.
- Deixar o usuário utilizar por conta própria, colocando-o no controle da situação, dando liberdade de uso.
- Após o término de dez minutos da uso, realizar as perguntas do questionário de usabilidade para o usuário.

Conforme os testes iam sendo realizados, os resultados eram analisados. As formas de coletar resultados dos testes foram o vídeo de gravação da tela do usuário, o formulário preenchido após o teste e as expressões faciais e verbais que o usuário tinha durante o teste.

## 9.5 COLETA DE RESULTADOS DAS GRAVAÇÕES DE TELA

Após realização dos testes em que foi possível gravar a tela do usuário, estes vídeos foram assistidos e analisados. Foi possível identificar alguns problemas de interação com facilidade por esta técnica de coleta.

O quadro 7 relata o resultado de cada interface disponível na aplicação em relação aos itens de problema identificados nelas, que puderam ser identificadas com os vídeos registrados e os comentários durante o teste.

**Quadro 7 – Problemas identificados nos registros em vídeo dos testes de usabilidade.**

<b>Interfaces</b>	<b>Problema levantado</b>	<b>Frequência</b>
<i>Tela de apresentação</i>	No último <i>slide</i> entender que deveria deslizar a tela (como é nos demais <i>slides</i> anteriores), quando precisar acionar o botão “continuar”	Baixa
<i>Cadastro de usuário</i>	Dificuldade de entender que campos são obrigatórios de serem preenchidos. Alguns usuários ficaram confusos, pois não existe nenhuma mensagem ou símbolo que alerte o que deve ser preenchido.	Alta
<i>Cadastro de usuário,</i> <i>Cadastro de unidade</i>	Cautela na hora de preencher os campos de data ou hora. Os usuários fazem isso com certa demora.	Alta
<i>Filtro de serviços</i>	Atraso para acionar o botão <i>aplicar filtro</i> . Possivelmente os usuários não sabem o que fazer após utilizar os filtros.	Média
<i>Atendimento</i>	Dificuldade de os usuários entenderem seu papel, o que devem fazer, durante o processo de atendimento.	Média

Sobre a tela de apresentação, foi percebido durante os vídeos assistidos posteriormente, em que é possível ver os casos de toque na tela. Dois usuários deslizaram corretamente desde o primeiro slide (que são 4), e no último, em que é disposto um botão para continuar, eles tentaram deslizar novamente. A expectativa deles, ao que parece, era que a maneira de continuar a interação com o ato de deslizar novamente, como antes. Porém houve esta quebra de expectativa deles, a qual foi claramente diagnosticada.

No cadastro de usuário, o problema de entender quais campos eram obrigatórios para realizar a atividade, foi diagnosticado tanto no vídeo, como nas expressões durante o teste. No vídeo, houve um usuário que passou um tempo para enxergar qual campo ainda estava faltando, para poder ativar o uso do botão de cadastro. Após entender onde estava o único campo em branco, preencheu e efetivou o cadastro. A solução, que é a mais comum de ser adotada, é colocar um símbolo como um asterisco ao lado de cada campo obrigatório e informar quais campos estão faltando ao tentar acionar a confirmação de cadastro.

Esta última ocorrência até abre margem para a interpretação de outro problema: os campos estão destacados na interface suficiente de tal forma que possam ser enxergados com facilidade? Esta demora em achar qual campo estava faltando induz isso. Talvez seja necessário um destaque maior dos campos de digitação nos formulários. Muitas das partes da interface possuem fundo branco. A diferença, quem sabe, não seja tão clara entre os diferentes tipos de componentes visuais das interfaces. Poderia ser buscado, junto a um profissional de *design*, que meios podem ser empregados para resolver este tipo de deficiência nas interfaces.

Sobre a cautela e demora dos usuários em preencher campos de data e hora, o motivo deve ser pelo fato do elemento visual de escolha de data e hora, é com ação de deslizar a tela. O usuário desliza na direção vertical a escolha de dia, mês e ano para datas e hora e minuto para horários. Esse deslizamento, provido pelos componentes utilizados na implementação das interfaces, é

sensível demais. Assim, se o usuário utilizar força demais para deslizar, pode passar da opção que queira selecionar.

Entretanto, antes de colocar como um problema desta aplicação, o que foi percebido é que esta cautela dos usuários ocorre desde o início do uso destes componentes. Isso pode querer dizer que a pessoa, só de entender como é a interação com este componente, já se previne e usa com cuidado, devagar. Este elemento de seleção de datas e horas é um padrão adotado em muitos *frameworks* de construção de interfaces de usuários. E isto é um ponto interessante, pois ao adotar ferramentas de trabalho largamente utilizadas no mercado, o projeto herdou este problema. Uma solução seria encontrar um componente melhor, que se enquadre com mais facilidade na escolha destes tipos de dado, ou, se possível, modificar o atual para melhorar a usabilidade dele neste projeto.

No atendimento, a dificuldade dos usuários de entenderem o passo a passo, pode ser explicado pelo fato de que ainda não foi desenvolvido um módulo de interface de ajuda para atividades complexas, como esta. Um recurso de ajuda poderia exibir marcações, dicas, setas, na interface de forma a ajudar os usuários a entenderem o funcionamento, ensinando-os detalhes nos primeiros usos da ferramenta. Esta técnica de aprendizado de usuários, como se fosse uma aula, diretamente na interface do usuário é utilizada em muitos produtos de *software* móveis e *webapps*.

Ainda no atendimento, foi questionado por alguns usuários por que a mensagem do sistema para dar *feedback* ao usuário, enquanto ele espera a ação do outro usuário, é colocada em um “botão opaco”, e não de outra forma mais convencional para dar uma mensagem. Este diálogo com o usuário pelo sistema, de fato, poderia estar apresentado de outra forma, dando mais atenção a ele, de modo que o usuário se prenda no que é necessário para entender o papel dele em todos os momentos, seja quando ele precisar acionar o botão que avança a etapa de responsabilidade dele, seja quando ele precisar aguardar o outro usuário fazer alguma atividade.

## 9.6 COLETA DE RESULTADOS DO QUESTIONÁRIO

O questionário aplicado recolheu a opinião dos usuários logo após a realização dos testes. Estes resultados possibilitaram mensurar a qualidade da interação com o usuário sobre os aspectos considerados relevantes para qualquer sistema de informação. Estes dados resultantes mostram que, para a maioria dos casos, as pessoas deram notas positivas para as questões apresentadas em que se pedia uma pontuação de um à cinco sobre determinado aspecto de usabilidade. Outras perguntas, em que se pedia uma classificação da frequência de um dado problema de interação, houve um equilíbrio, houveram mais casos de resposta *média*, dentre as demais opções *baixa* e *alta*. A seguir os resultados considerações de cada pergunta são descritos.

### Pergunta sobre a nota para a rapidez de aprendizado

O objetivo desta pergunta era entender se os usuários sentiam que estavam assimilando rapidamente como realizar as atividades no aplicativo. Quer dizer se, em pouco tempo, é possível assimilar o conhecimento necessário para realizar um cadastro, uma pesquisa, entender como funciona um fluxo de atendimento, enfim. Aqui é possível notar que as notas foram positivas, ou seja, consideraram que a ferramenta é relativamente rápida de ser aprendida.

#### Rapidez de aprendizado

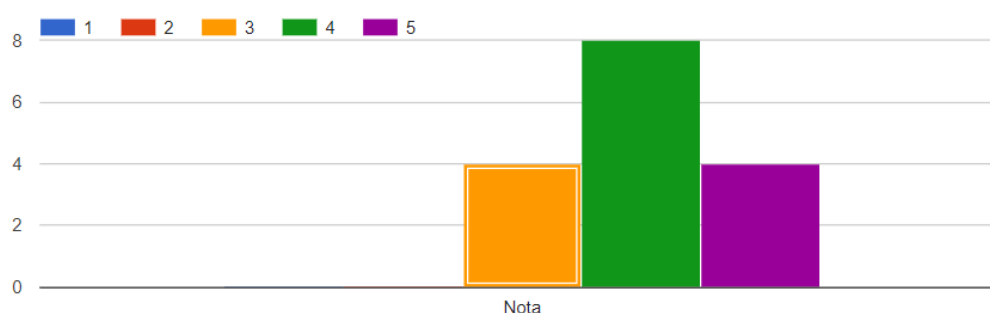


Figura 38 – resultados da pergunta sobre a rapidez de aprendizado

### Pergunta sobre nota para facilidade de uso, depois de aprendido

Esta questão pede para os usuários refletirem se o aplicativo depois do primeiro uso fica mais fácil de utiliza. Ajuda a entender se o usuário acha que, em casos do primeiro uso ter sido um pouco difícil, o aplicativo seria mais fácil em utilizações futuras. Se a ferramenta é difícil só no primeiro contato (ou nem isso), ou se sempre será difícil.

Relacionando com a resposta da pergunta anterior a esta, reforça o entendimento que, pela diminuição das respostas de nota três, os usuários realmente acham que poderia haver uma melhora ao utilizar mais de uma vez.

#### Facilidade de uso, depois de aprendido

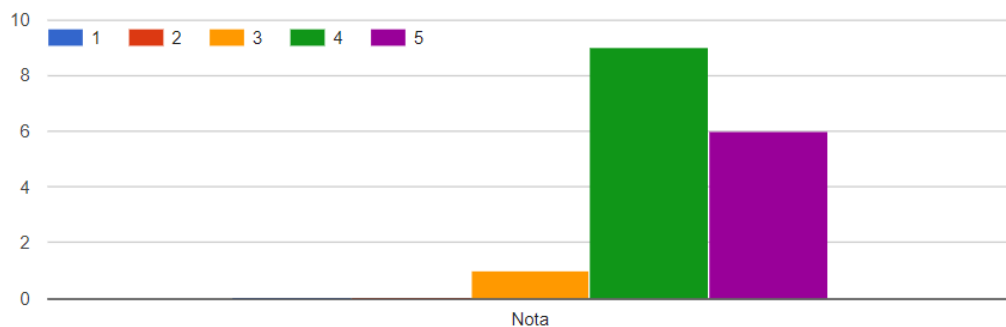


Figura 39 – resultados da pergunta sobre facilidade de uso após aprendizado

#### Pergunta sobre nota para capacidade de memorização do uso

Aqui, o usuário precisava refletir sobre quão o aplicativo e sua interface poderiam ser lembrados mais tarde. Se achavam que depois de um período sem uso, lembrariam como de realizar atividades que haviam acabado de aprender. A pergunta e a resposta se assemelham à anterior, porém são colocadas de duas perspectivas diferentes para os usuários. O resultado é satisfatório.

O quão você acha que o uso pode ser fácil de ser memorizado / lembrado ?

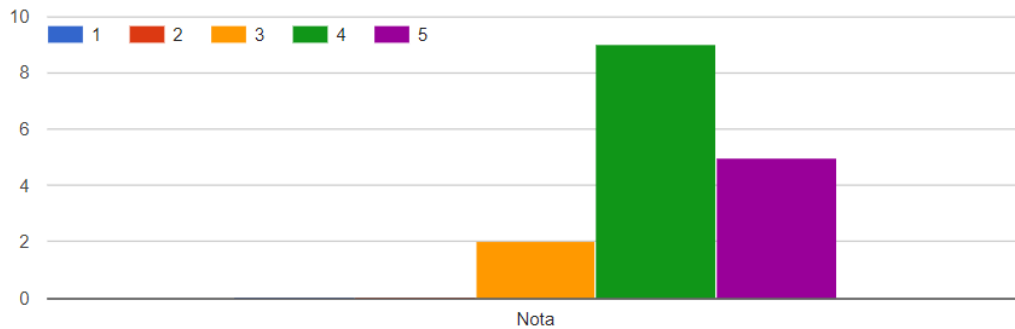


Figura 40 – resultados da pergunta sobre nota para capacidade de memorização do uso

### Pergunta sobre nota do quanto o aplicativo é agradável

O usuário informa uma nota sobre a visão de satisfação do uso, de maneira geral. Utilizando o termo “agradável”, foi reforçado que esta pergunta pede a resposta para o que usuário sentiu ao utilizar o aplicativo, se ele se sentiu bem ao fazer uso da ferramenta, o quão bem foi. Vale lembrar que os usuários, a maioria, estavam em ambientes naturais do cotidiano deles, sem artifícios como câmeras, gravadores, ou um local que não conheciam. Esse fator deve influenciar, para que respostas negativas não sejam dadas influenciadas pelas condições do ambiente em que o usuário está inserido, no momento do teste.

O aplicativo é agradável, bom de ser utilizado

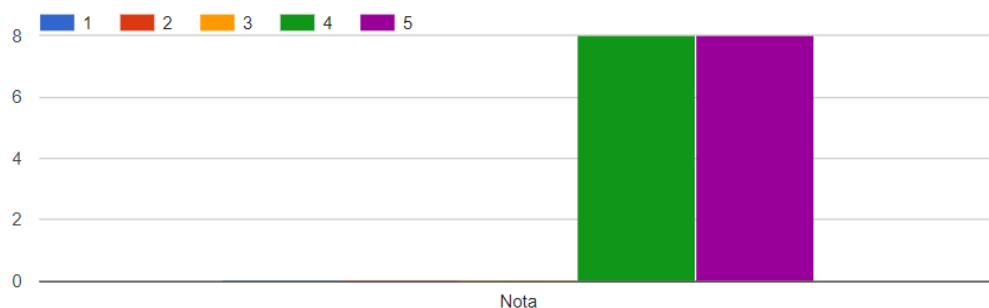


Figura 41 – resultados da pergunta sobre nota do quanto o aplicativo é agradável

### Pergunta sobre nota quanto a rapidez de executar as tarefas

Ainda com esquema de perguntas por notas, o usuário responde se o aplicativo é rápido para executar as tarefas – se não apresenta demoras ao carregar as informações, sobre o tempo de resposta para as ações. Este questionamento é pertinente, pois o fator desempenho sempre é pauta em conversas com usuários, de diferentes projetos conhecidos.

#### O aplicativo é rápido para executar as tarefas

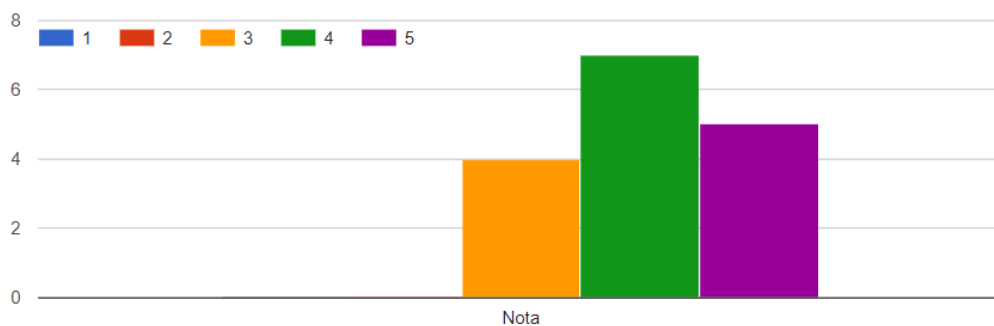


Figura 42 – resultados da pergunta sobre nota quanto a rapidez de executar as tarefas

#### Pergunta sobre não encontrar alguma opção esperada

Sabe-se que esta é uma questão recorrente em muitos casos de uso de sistemas. O usuário, frequentemente, sempre aplica sua visão à um *software*, imaginando opções e funcionalidades que poderiam ter, de acordo com a sua expectativa. Entretanto, ainda houveram ocorrências de usuários que acharam não sentir ausência de nenhum item nas interfaces, houve um equilíbrio nas respostas – mesmo o aplicativo ainda estando incompleto.

De qualquer forma, é notável que algumas opções ainda faltam em várias interfaces disponíveis para o teste, pois a ferramenta ainda está em estado de desenvolvimento, com recursos implementados por completo, e outros parciais. Portanto, é esperado que conforme o desenvolvimento avance, ocorrências de ausência de opções, falta de cobertura de expectativas do usuário na interface, sejam amezinhadas.

Frequência de vezes que não encontrou uma opção que esperava encontrar, como um link, uma mensagem de ajuda, um botão

16 respostas

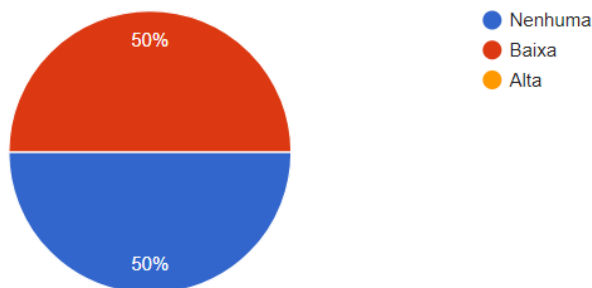


Figura 43 – resultados da pergunta sobre não encontrar alguma opção esperada

### Pergunta sobre não saber o que fazer em algum momento da interação

Esta pergunta foi inspirada na técnica de teste de comunicabilidade, a qual consiste em realizar a gravação do usuário utilizando o sistema, e rotular depois suas expressões, onde um dos rótulos se chama “e agora”, quando o usuário reagir de tal forma. (PRATES e BARBOSA, 2003).

Frequência de vezes que não soube o que fazer. Algo como um pensamento : "E agora?"

16 respostas

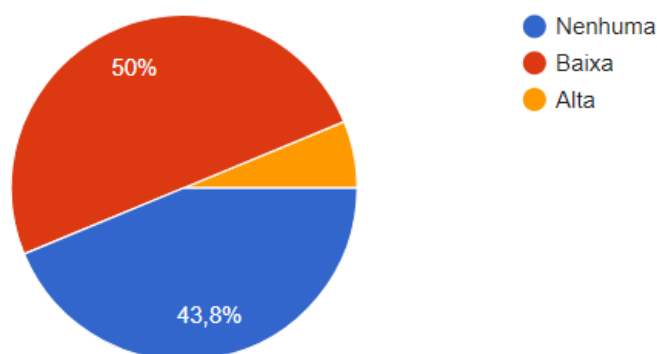


Figura 44 – resultados da pergunta sobre não saber o que fazer em algum momento da interação

### Pergunta sobre sentimento de ter ficado sem resposta do sistema

Quando um sistema não responde ao usuário sobre uma exceção que ocorreu, ou sobre uma mensagem de sucesso de operação, ou algo similar, o usuário pode ficar se perguntando o que aconteceu após executar uma tarefa ou usar algum componente da tela. Para saber sobre esses casos, esta pergunta se mostrou pertinente.

### Frequência de vezes que você sentiu ter ficado sem resposta do sistema sobre algo que aconteceu?

16 respostas

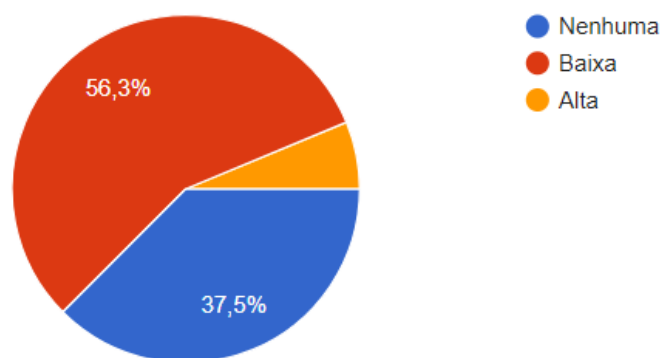


Figura 45 – resultados da pergunta sobre sentimento de ter ficado sem resposta do sistema

## 9.7 CONCLUSÕES DA AVALIAÇÃO

Com a avaliação realizada, foi possível fechar um ciclo de desenvolvimento do projeto, coletando *feedback* dos usuários sobre o incremento desenvolvido até o momento da avaliação. A medida que novos incrementos de desenvolvimento sejam realizados, é pertinente sempre fazer novas avaliações como esta, objetivando garantir a melhoria contínua no quesito usabilidade.

Os resultados apontaram os gargalos e as falhas de interação existentes. É preciso, portanto, orientar o desenvolvimento para atender às melhorias de usabilidade detectadas, para que o sistema não seja apenas funcional, mas agradável a quem mais interessa – os usuários.

## 10 CONCLUSÕES

Este trabalho possibilitou desenvolver um software de maneira efetiva, partindo de uma concepção inicial, quando apenas uma ideia de solução para um problema em tecnologia existia, até chegar ao produto concreto, tornando-se um projeto de sistemas de informação com todas as etapas e considerações mínimas para construção de um produto de sistemas, como orienta a comunidade científica de engenharia de software.

Por meio de um questionário de campo realizado foi possível, mesmo com uma amostra pequena, mas representativa do público alvo, constatar que muitas pessoas partilham do mesmo anseio que autor levantou, sobre como a prestação de serviços poderia ser abordada, se tratando de sistemas de informação, mais especificamente de um aplicativo móvel.

Com as técnicas empregadas para a especificação de requisitos, foi possível chegar a um pequeno universo de todas as vertentes e possibilidades que a ideia poderia realizar, afunilando para as principais, capazes de prover um produto para diferentes áreas de serviços. Os casos de uso levantados, juntos, trabalharam para chegar a um objetivo: divulgação de serviços que pudessem ser prestados de tal forma que pudessem ser consultados, contratados, agendados ou acionados por meio de um aplicativo móvel, e que o contexto de localização geográfica fosse levado em consideração para o usuário.

O modelo de projeto de sistemas desenvolvido possibilitou definir um roteiro, um guia para uma implementação do software com a segurança maior de alcançar qualidade, ou seja, que o software desenvolvido seguisse os requisitos levantados nas etapas de especialização e análise de sistemas, e que respeitasse as boas práticas de engenharia e arquitetura de software, viabilizando uma manutenção eficaz do sistema.

A implementação do software realizada foi enxergada como uma instância do modelo de projeto. Ou seja, seguindo o modelo, existiriam diversas maneiras de implementar o projeto, diferentes instâncias, considerando ferramentas, linguagens diferentes, que pudessem ser descobertas e analisadas para atender também as necessidades, mesmo com pequenas diferenças. Daqui

se tira a melhor lição sobre tudo que foi trabalhado em engenharia de software neste trabalho: o modelo de projeto de sistemas é extremamente importante para qualquer projeto de software, uma vez que este assegura a qualidade do produto a ser criado, interligando toda a abstração que os requisitos de software possuem naturalmente, e todas as considerações para tornar este universo de ideias um produto palpável, um software, atuando como um guia de implementação o qual possa ser desenvolvido de diversas formas, mas alcançando os mesmos objetivos. Vale lembrar que a comunicação entre diferentes áreas envolvidas em um projeto de software, sempre foi um martírio para qualquer projeto desta área. O papel do modelo de projeto, como interlocutor entre abstração de alto e baixo nível, possibilita maior segurança em transmitir, portanto, as necessidades do cliente para o código, de fato.

Com o teste de usabilidade realizado com usuários que representassem cada tipo de usuário do público alvo, utilizando o aplicativo ainda em desenvolvimento, foi possível levantar gargalos de interação, além do nível de satisfação dos usuários, suas expectativas e anseios sobre a ferramenta. Mais ainda, informalmente, foi possível dialogar com estas pessoas e perceber o real interesse no aplicativo, tanto para buscar um serviço quanto para publicar um. O que se percebeu foi a constatação da visão inicial, ainda na concepção, de que muitas pessoas irão atuar como consumidor e prestador na plataforma, mesmo aquelas que atualmente ainda não se titulam como prestadoras, mas querem aproveitar seu tempo realizando algumas atividades extras.

Para concluir, é tangenciada a possibilidade de este projeto contribuir para a melhoria de qualidade de vida do público alvo, uma vez que este projeto de sistemas de informação possui a pretensão de eliminar dificuldades que temos hoje, mas que já poderíamos não ter mais, dado as tecnologias disponíveis que existem no mercado para construção de uma solução como esta. Que as pessoas possam se comunicar ainda mais quando o assunto for serviços, quando qualquer necessidade relacionada a esta palavra possa ser facilitada pela tecnologia. Afinal, sabemos que a principal motivação de buscar novas tecnologias é melhorar a vida das pessoas, ou de um conjunto de pessoas, de todas as formas possíveis, garantindo mais qualidade a todos. E isto, certamente, é a principal motivação deste projeto.

## REFERÊNCIAS

BEVERLOO, P. et al. Push API. **W3**, 2017. Disponível em: <<https://www.w3.org/TR/push-api/>>. Acesso em: 13 set. 2017.

CAMPOS, F. F. D. Por que Node.js? **VIZIR**, 2016. Disponível em: <<http://vizir.com.br/2016/04/por-que-node-js/>>. Acesso em: 13 set. 2017.

CAPAN, T. Why The Hell Would I Use Node.js? A Case-by-Case Tutorial. **Toptal**, 2016. Disponível em: <<https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>>. Acesso em: 08 set. 2017.

FETTE, I.; MELNIKOV, A. RFC 6455 - The WebSocket Protocol. **Internet Engineering Task Force (IETF)**, 2011. Disponível em: <<https://tools.ietf.org/html/rfc6455>>. Acesso em: 13 set. 2017.

FITZSIMMONS, J. A. **Administração de Serviços. Operações, Estratégia e Tecnologia da Informação**. 7ª. ed. [S.l.]: AMGH, 2014.

FURTADO, A. B.; COSTA JUNIOR, J. V. **PRÁTICA DE ANÁLISE E PROJETO DE SISTEMAS**. Belém: abfurtado.com.br, 2010.

GOOGLE meu negócio. **Google meu negócio**, 2017. Disponível em: <<https://www.google.com.br/intl/pt-BR/business/>>. Acesso em: 28 jul. 2017.

Hi Job. **Hi Job**, 2017. Disponível em: <<http://hijob.com.br>>. Acesso em: 12 Agosto 2017.

IBGE. IBGE produz novos indicadores de mercado de trabalho. **IBGE**, 2016. Disponível em: <<http://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/9915-ibge-produz-novos-indicadores-de-mercado-de-trabalho.html>>. Acesso em: 8 Agosto 2017.

JONES, M.; BRADLEY, J.; SAKIMURA, N. RFC 7719 - JSON Web Token (JWT). **Internet Engineering Task Force**, 2015. Disponível em: <<https://tools.ietf.org/html/rfc7519>>. Acesso em: 10 set. 2017.

JORNAL Nacional. **Jornal Nacional**, 2017. Disponível em: <<http://g1.globo.com/jornal-nacional/noticia/2017/07/crise-economica-pode-ser-chance-para-mudanca-de-carreira.html>>. Acesso em: 8 ago. 2017.

LARUCCIA, M. Prof Dr Mauro Laruccia. **Mauro Laruccia**, 2009. Disponível em: <<http://www.maurolaruccia.adm.br/wordpress/?p=108>>. Acesso em: 30 jul. 2017.

LINO, M. O que é Apache Cordova? **Programação Full Stack**, 2015. Disponível em: <<http://blog.marianalino.com.br/o-que-e-apache-cordova/>>. Acesso em: 12 set. 2017.

MINISTÉRIO DA INDÚSTRIA, C. E. E. S. A importância do setor terciário. **mdic**, 2017. Disponível em: <<http://www.mdic.gov.br/index.php/comercio-servicos/a-secretaria-de-comercio-e-servicos-scs/402-a-importancia-do-setor-terciario>>. Acesso em: 7 ago. 2017.

MONTEIRO, F. <https://imasters.com.br/front-end/entendendo-como-funciona-o-restful-de-verdade/?trace=1519021197>. **IMasters**, 2015. Disponível em: <<https://imasters.com.br/front-end/entendendo-como-funciona-o-restful-de-verdade/?trace=1519021197>>. Acesso em: 07 set. 2017.

OSTERWALDER, A. **Business Model Generation**. [S.l.]: Starlin Alta Editora, 2010.

PRATES, O. R.; BARBOSA, D. J. S. PUC RIO. **http://www-di.inf.puc-rio.br/~simone/files/JAI2003\_avaliacao\_s.pdf**, 2003. Disponível em: <[http://www-di.inf.puc-rio.br/~simone/files/JAI2003\\_avaliacao\\_s.pdf](http://www-di.inf.puc-rio.br/~simone/files/JAI2003_avaliacao_s.pdf)>.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software. Uma abordagem profissional**. 8ª. ed. [S.l.]: bookman, 2016.

REIS, E. **A Startup enxuta**. [S.l.]: Texto Editores, 2011.

SANTOS, A. O. Confederação Nacional do Comércio de Bens, Serviços e Turismo. **CNC**, 2017. Disponível em: <<http://cnc.org.br/central-do-conhecimento/artigos-do-presidente/economia/avancos-tecnologicos-e-sobrevivencia>>. Acesso em: 06 Agosto 2017.

SIMBRACS. Comércio e Serviços. **SIMBRACS**, n. 2<sup>a</sup>, 2013.

SOUZA, D. Diferença entre os patterns PO, POJO, BO, DTO e VO. **Dev media**, 2017. Disponível em: <<http://www.devmedia.com.br/diferenca-entre-os-patterns-po-pojo-bo-dto-e-vo/28162>>. Acesso em: 13 set. 2017.

STECKY, M. E. 5 Easy Steps to Understanding JSON Web Tokens (JWT). **Vandum software**, 2016. Disponível em: <<https://medium.com/vandum-software/5-easy-steps-to-understanding-json-web-tokens-jwt-1164c0adfcec>>. Acesso em: 20 jul. 2017.

STEPPAT, N. Bancos de dados não relacionais e o movimento NoSQL. **Caelum**, 2009. Disponível em: <<http://blog.caelum.com.br/bancos-de-dados-nao-relacionais-e-o-movimento-nosql/>>. Acesso em: 07 set. 2017.

TORRES, J. **Guia d Startup**. [S.l.]: Casa do Código, 2013.

XAVIER, J. Aplicativo 'Mission 7' facilita acesso a serviços diversos em Manaus. **Em Tempo**, 2017. Disponível em: <<http://www.emtempo.com.br/aplicativo-mission-7-facilita-acesso-a-servicos-diversos-em-manaus/>>. Acesso em: 12 ago. 2017.

## APÊNDICE A – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS

# PROJETO ISERVICE

## DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS

<b>Histórico de Modificações</b>		
<b>Data</b>	<b>Autor</b>	<b>Descrição</b>
05/05/2017	Ewertom Moraes	Criação do documento.
16/05/2017	Ewertom Moraes	Definida primeira versão do documento completo.
26/06/2017	Ewertom Moraes	Revisão de Casos de Uso Atende Cliente, Divulga Unidade. Remoção de Ator Unidade

### 1. Introdução

Este documento objetiva apresentar e descrever o projeto de software denominado IService, expondo os objetivos do software, definindo seu escopo e detalhando as funcionalidades pretendidas. Para descrever as funcionalidades do sistema, a técnica de modelagem de casos de uso será utilizada.

### 2. Ambiente do Sistema

Prestadores de serviço de qualquer tipo de serviço atendem clientes em um endereço fixo, por atendimento ambulante e em domicílio. Geralmente aceitam mais de uma forma de pagamento. O prestador mantém contato com clientes por telefone, divulga um cartão de visita, possui um site ou endereço eletrônico qualquer.

Além disso, pessoas comuns, que não se titulam prestadores de serviço, mas que possuem experiência em dado serviço, podem exercer uma atividade qualquer remunerada (técnica ou não). Essas pessoas podem divulgar suas habilidades e técnicas de alguma forma atualmente, ou não.

O sistema entra no contexto de divulgação, contratação e de gerência do serviço, onde essas três etapas podem ser utilizadas de maneira completa ou parcial.

### 3. Objetivos do produto

IService será uma aplicação com a finalidade de englobar a comunicação entre prestadores de serviço e clientes para contato, divulgação, contratação de quaisquer tipos

de serviço, com transação de pagamento pela aplicação. Pessoas físicas e jurídicas poderão construir e divulgar seu perfil, informando quais serviços disponibiliza.

Os interessados poderão procurar quem os possa atender por palavras, proximidade por localização geográfica, disponibilidade de horário, reputação, etc. Quando um cliente adquirir o serviço, poderá dar uma nota para o mesmo, semelhante funciona em sites de lojas virtuais, com produtos.

A aplicação possuirá um módulo avançado para pessoas que desejam disponibilizar o serviço por agendamento, permitindo que os próprios clientes façam o agendamento, e até personalizar a contratação de serviços, configurando campos específicos de cliente que será atendido, por serviço.

#### **4. Benefícios do Sistema**

- sistema deve contemplar as seguintes funcionalidades:
- Criação e Divulgação de Perfil de Usuário, com opção de cadastro por meio de formulário ou utilizando vínculo com rede social Facebook.
- Divulgação de Serviços do usuário, informando área do serviço, tipo, forma de atendimento, preço, entre outras informações.
- Divulgação de Unidades de Serviço (com endereço fixo ou móvel - localização em tempo real)
- Pesquisa de Serviços por nome, tipo, preço, localização, forma de atendimento, disponibilidade, reputação.
- Contratação de Serviços com opção de pagamento pela plataforma com cartão de crédito, pontuando a qualidade do atendimento, solicitação de agendamento de serviço pelos clientes e gerência da agenda de atendimentos. Registro da ocorrência do atendimento.
- Atendimento de Cliente pela plataforma, quando solicitado, registrando a locomoção do prestador/cliente até o encontro para o atendimento quando o serviço for do tipo que precisa do encontro das partes.
- Divulgação de Agenda de Atendimento.
- Consulta de Atendimentos Realizados por diferentes critérios para confecção de relatórios.
- Gerência de caixa, com operações de abertura e fechamento do mesmo, inclusive por serviço disponibilizado pelo usuário.
- Emissão de nota fiscal para serviços prestados.
- Configuração de Clientes Personalizados, com a modelagem de um formulário que deve ser preenchido por clientes que queiram um serviço pela primeira vez, e modelagem de informações adicionais no ato das contratações do serviço, para personalizar as informações de cliente e registros de atendimentos de acordo com a necessidade de cada usuário e/ou serviço.

#### **5. Diagramas de casos de uso**

O sistema possui basicamente dois perfis de usuários: prestador de serviço e consumidor. Estes dois perfis são, na verdade, estados que usuários podem assumir em determinado momento, dependendo do contexto da situação corrente na aplicação. Um usuário poderá assumir o papel de prestador de serviço em dado momento, e em outro momento assumir

papel de consumidor e, portanto, ambos são considerados estados de um único ator: usuário.

Assim, são definidos estes dois perfis como dois tipos de atores do sistema, para que a especificação por diagramas de casos de uso seja mais clara quanto a interação e relação dos usuários com cada caso.

O sistema a ser desenvolvido é dividido em dois subsistemas básicos: Busca de serviços, gerência de serviços.

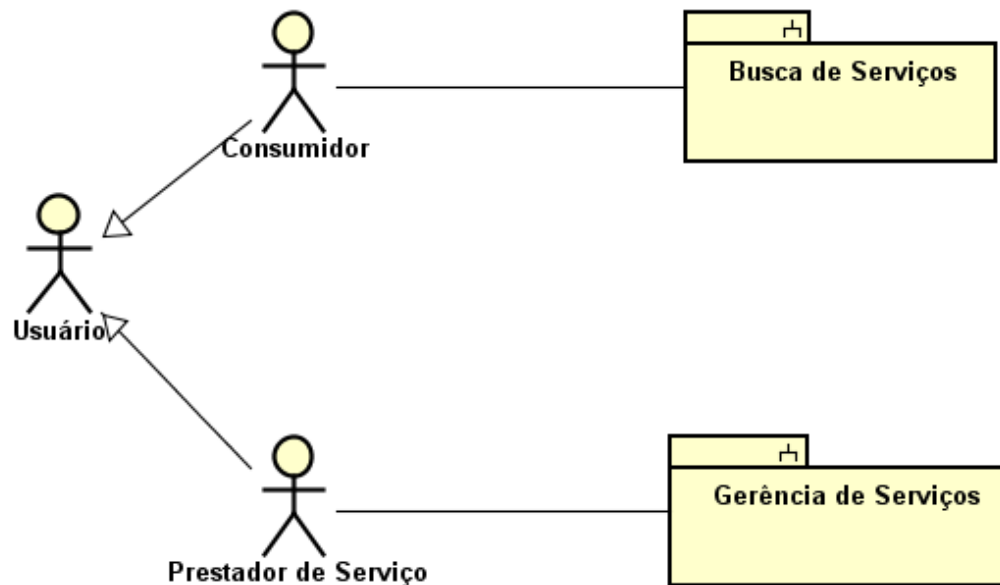


Figura 1. Diagrama de Subsistemas

### 5.1. Busca de Serviços

Este subsistema é responsável por prover funcionalidades de busca e contratação de serviços para os usuários. Aqui, o usuário assume o papel de Consumidor, e, portanto, será referenciado com este termo.

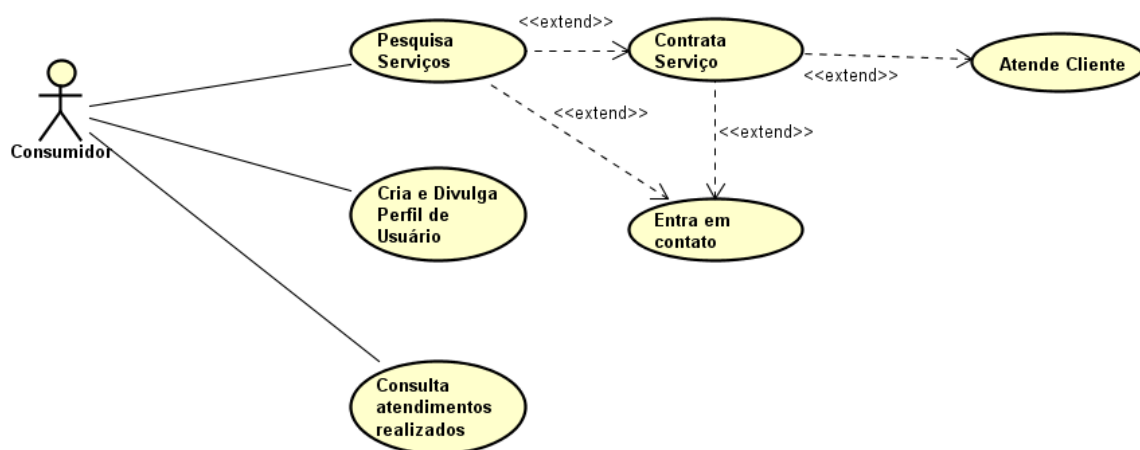


Figura 2. Diagrama de Casos de Uso do subsistema Busca de Serviços

## 5.2. Gerência de Serviços

Este subsistema é responsável por prover funcionalidades de gerenciamento de serviços que o usuário divulga. Nele é possível divulgar serviços, atender consumidores do serviço divulgado e gerenciar o serviço.

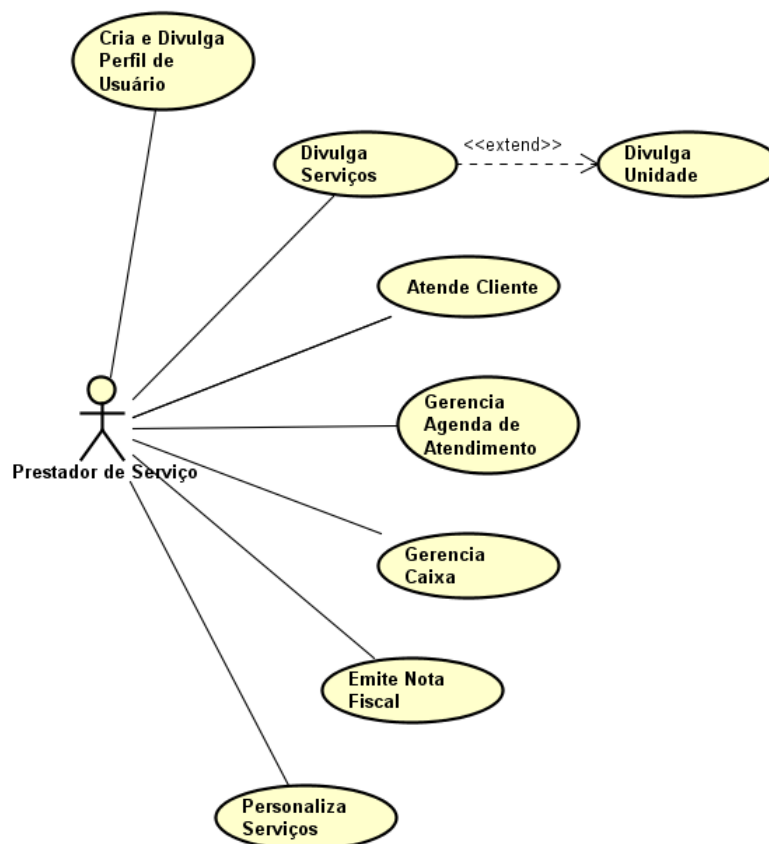


Figura 3. Diagrama de casos de uso do subsistema Gerência de Serviços

## 6. Descrição de casos de uso

Nesta seção são descritos os casos de uso do Sistema IService, descrevendo-os em linhas gerais, especificando os atores que realizam, pre-condições de realização do caso, um fluxo principal, um ou mais fluxos alternativos, exceções (para tratamento de falhas) e pós condições.

### 6.1. Cria e Divulga Perfil de Usuário

<b>Descrição</b>	Pessoas físicas e jurídicas se cadastram, com campos relevantes para cada tipo de pessoa (Física ou Jurídica). O cadastro pode ser realizado com o uso de um formulário para digitar informações da pessoa ou com vínculo ao Facebook.
<b>Atores</b>	Consumidor, Prestador de Serviço
<b>Precondições</b>	O usuário deve estar com o aplicativo aberto e sem login ativo.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário escolhe se deseja se cadastrar de forma padrão, informando seus dados pessoais ou se quer usar sua conta do <i>facebook</i>.</li> <li>2. O sistema processa as informações cadastrais.</li> <li>3. O sistema avisa o usuário que o cadastro foi efetuado com sucesso.</li> </ol>
<b>Fluxo Alternativo 1 Cadastro Padrão</b>	<ol style="list-style-type: none"> <li>A. no passo 1, o usuário escolhe a opção de Cadastro padrão.</li> <li>B. o usuário informa os seguintes dados: nome, e-mail, telefone, senha, lista de interesses.</li> <li>C. o usuário clica em “Me cadastrar”.</li> </ol>
<b>Fluxo Alternativo 2 Cadastro com Facebook</b>	<ol style="list-style-type: none"> <li>A. no passo 1, o usuário escolhe a opção de cadastro com Facebook.</li> <li>B. o usuário é redirecionado para uma página de autorização do <i>facebook</i>.</li> <li>C. O usuário confirma e autoriza seu cadastro usando uma conta do <i>facebook</i>.</li> </ol>
<b>Pós-condições</b>	O usuário é cadastrado no sistema e é direcionado para a página de Perfil de Usuário.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. no fluxo alternativo 1, passo A, o usuário informa os dados em formato inadequado. Ao clicar em Me cadastrar, o sistema alerta o usuário dos campos que necessitam de ajuste e retorna ao passo A.</li> <li>2. no fluxo alternativo 2, no passo B, o usuário não possui em seu dispositivo uma conta ativa do <i>facebook</i>. O sistema alerta e redireciona o usuário para uma página de autenticação oficial do <i>facebook</i>. Caso não faça login, o caso de uso é encerrado. Com o login do <i>facebook</i> efetivado, o caso de uso é concluído.</li> </ol>

### 6.2. Pesquisa de Serviços

<b>Descrição</b>	O usuário pesquisa serviços que pretende conhecer ou contratar por meio de palavras chave ou visualizando em um mapa.
<b>Atores</b>	Consumidor
<b>Precondições</b>	O usuário deve estar autenticado
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário aciona a opção de Buscar Serviços.</li> <li>2. O sistema exibe um campo de pesquisa e um mapa.</li> <li>3. O usuário utiliza o campo de pesquisa para informar uma palavra chave para buscar pessoas que prestam serviços.</li> <li>4. O sistema processa a pesquisa e exibe resultados que satisfaçam o termo utilizado, listando serviços que contenham o termo em quaisquer campos relacionados ao cadastro de serviços no sistema.</li> <li>5. O usuário escolhe um dos resultados da pesquisa.</li> <li>6. O sistema exibe as informações do serviço e do prestador do serviço.</li> <li>7. Eventualmente o usuário pode acionar a opção contratar serviço ou entrar em contato.</li> </ol>
<b>Fluxo Alternativo 1</b> <b>Pesquisa por mapa</b>	<ol style="list-style-type: none"> <li>A. No passo 1, o usuário utiliza o mapa disponível para visualizar Serviços disponíveis.</li> <li>B. O usuário locomove o mapa para uma determinada localidade desejada.</li> <li>C. o usuário seleciona um dos serviços disponíveis no mapa.</li> <li>D. O fluxo principal prossegue do passo 6.</li> </ol>
<b>Fluxo Alternativo 2</b> <b>Pesquisa avançada</b>	<ol style="list-style-type: none"> <li>A. No passo 1, o usuário aciona a opção de pesquisa avançada.</li> <li>B. O sistema exibe uma tela com campos de filtro: tipo de serviço, faixa de preço, distância máxima entre localização do usuário e local do serviço, ramo do serviço.</li> <li>C. O usuário informa os valores que deseja para filtrar.</li> <li>D. O fluxo principal prossegue do passo 4.</li> </ol>
<b>Pós-condições</b>	O sistema exibe as informações do serviço e do prestador do serviço. Eventualmente o caso de uso Contrata serviço ou o Entra em contato é acionado.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. No passo 3 do fluxo principal, os termos e condições de busca não satisfazem qualquer serviço cadastrado. O sistema exibe uma mensagem alertando que não foi possível encontrar serviços.</li> <li>2. No fluxo alternativo 1, o usuário não encontra qualquer serviço disponível no trecho de mapa que está visualizando. O usuário define uma nova forma de buscar pelo mapa ou o caso de uso é interrompido.</li> <li>3. No fluxo alternativo 2, o usuário informa valores que não satisfazem qualquer serviço cadastrado. O usuário é alertado sobre isso. Este define novos critérios de busca ou o caso de uso é interrompido.</li> </ol>

### 6.3. Contrata Serviços

<b>Descrição</b>	O consumidor contrata serviços de prestadores de serviço (pessoas físicas e jurídicas) cadastrados no sistema. A realização do serviço pode ser para o instante da contratação ou agendado.
<b>Atores</b>	Consumidor
<b>Precondições</b>	O usuário deve estar autenticado e o caso de uso Pesquisa Serviço foi executado e, neste, o usuário selecionou a opção Contratar serviço.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O prestador do serviço é notificado sobre a solicitação de atendimento.</li> <li>2. O caso de uso Atende cliente é executado.</li> <li>3. O usuário dá uma nota ao atendimento.</li> </ol>
<b>Fluxo Alternativo 1 Serviço Agendado</b>	<ol style="list-style-type: none"> <li>A. no passo 1, após o usuário acionar a opção Contratar Serviço, o prestador de serviço é um prestador de serviço configurou o serviço para funcionar mediante agendamento. O sistema então exibe uma agenda.</li> <li>B. O consumidor seleciona um dia e hora disponível para ser atendido e aciona a opção Agendar Serviço.</li> <li>C. O prestador do serviço é notificado sobre o agendamento de atendimento.</li> </ol>
<b>Fluxo Alternativo 2 Serviço não dispõe contratação pela plataforma.</b>	<ol style="list-style-type: none"> <li>A. no passo 1, o serviço é do tipo divulgação - não possui opção de contratação pela plataforma e então o Sistema alerta o consumidor da restrição.</li> <li>B. O usuário seleciona opção entrar em contato com o prestador do serviço (aciona caso de uso “Entra em contato”) ou aciona opção cancelar para encerrar o caso de uso.</li> </ol>
<b>Pós-condições</b>	O prestador do serviço é notificado sobre a solicitação ou agendamento de atendimento, um registro de atendimento realizado é registrado com status iniciado.
<b>Casos de Uso Envolvidos</b>	<ul style="list-style-type: none"> <li>● Pesquisa Serviços</li> <li>● Atende Cliente</li> <li>● Entra em Contato</li> </ul>
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. No passo 1 do fluxo principal, o prestador de serviço não confirmou o atendimento, então o caso de uso é interrompido.</li> </ol>

#### 6.4. Divulgar Serviços.

<b>Descrição</b>	O prestador de serviço cadastra, altera, lista e exclui seus serviços que queira divulgar, disponibilizar na plataforma.
<b>Atores</b>	Prestador de Serviço
<b>Precondições</b>	O usuário deve estar autenticado e ter acessado a opção “Perfil”
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário clica na opção “Meus serviços”</li> <li>2. Na tela “Meus serviços”, o usuário clica em adicionar novo</li> </ol>

	<p>serviço.</p> <p>3. O Sistema exibe um formulário de cadastro simples.</p> <p>4. O usuário informa os dados do serviço a ser divulgado.</p> <p>5. O caso de uso Divulga Unidade é executado.</p> <p>6. O usuário escolhe a opção “Salvar”.</p>
<b>Fluxo Alternativo 1</b> <b>Cadastro Avançado</b>	<p>A. No passo 3, o usuário usa a opção para realizar cadastro avançado.</p> <p>B. O Sistema exibe outro formulário de cadastro, com opções mais avançadas (Cadastro de múltiplas unidades, cadastro de formulários de atendimento e cadastro do serviço, dentre outras opções).</p> <p>C. O passo 6 do fluxo principal é executado.</p>
<b>Fluxo Alternativo 2</b> <b>Alteração de serviço</b>	<p>A. no passo 2, o usuário seleciona um serviço já existente dentre os listados para alterar os dados do mesmo.</p> <p>B. O usuário altera os dados de interesse.</p> <p>C. O usuário clica em “Salvar”.</p>
<b>Fluxo Alternativo 3</b> <b>Exclusão de serviço</b>	<p>A. no passo 2, o usuário seleciona um serviço já existente dentre os listados para exclusão do mesmo.</p> <p>B. O sistema pede a confirmação do usuário quanto a exclusão do serviço.</p> <p>C. O usuário confirma a operação.</p>
<b>Fluxo Alternativo 4</b> <b>Consulta de serviço</b>	<p>A. no passo 2, o usuário seleciona um serviço já existente dentre os listados para consulta do mesmo.</p>
<b>Pós-condições</b>	O Inclusão ou Modificação em dado serviço é processada pelo sistema. O usuário é redirecionado para a tela “Meus serviços”.
<b>Exceções</b>	<p>1. No fluxo principal, passo 3, o usuário informa dados de forma inadequada e clica em salvar. O sistema alerta o usuário dos campos que precisam de ajuste e volta ao passo 3. O usuário corrige os dados para prosseguir ou escolhe a opção cancelar para encerrar o caso de uso.</p> <p>2. No fluxo alternativo 3, passo B, o usuário informa dados de forma inadequada e clica em salvar. O sistema alerta o usuário dos campos que precisam de ajuste e volta ao passo B. O usuário corrige os dados para prosseguir ou escolhe a opção cancelar para encerrar o caso de uso.</p>

## 6.5. Divulga Unidade

<b>Descrição</b>	O usuário cadastra, altera, lista e exclui unidades de atendimento de seus serviço.
<b>Atores</b>	Usuário

<b>Precondições</b>	O usuário deve estar autenticado e, no cadastro de serviço, ter acessado a opção de cadastro unidade.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a opção “Unidade”</li> <li>2. O sistema exibe uma tela adicional de Unidades de atendimento do serviço.</li> <li>3. O usuário clica em “Adicionar uma unidade”</li> <li>4. O usuário informa os dados da unidade(nome, pessoa responsável pelo atendimento, tipo de unidade, horário de atendimento).</li> <li>5. O usuário escolhe a opção “Salvar”.</li> </ol>
<b>Fluxo Alternativo 1</b> <b>Tipo de Unidade Fixa</b>	<ol style="list-style-type: none"> <li>A. No passo 4, o usuário escolhe como tipo unidade fixa.</li> <li>B. O sistema exibe um campo adicional associado com um mapa para que seja informado a localização da unidade.</li> <li>C. O usuário informa o local.</li> <li>D. O usuário escolhe a opção Salvar.</li> </ol>
<b>Fluxo Alternativo 2</b> <b>Tipo de Unidade Móvel</b>	<ol style="list-style-type: none"> <li>A. No passo 4, o usuário escolhe como tipo de unidade móvel.</li> <li>B. O sistema exibe uma mensagem alertando que será utilizada a localização do responsável da unidade para divulgar a localização móvel da unidade.</li> <li>C. O usuário confirma o entendimento da condição.</li> <li>D. O usuário escolhe a opção salvar</li> </ol>
<b>Fluxo Alternativo 3</b> <b>Alteração de unidade</b>	<ol style="list-style-type: none"> <li>A. no passo 3, o usuário seleciona uma unidade já existente dentre as listadas para consulta.</li> <li>B. O usuário aciona opção de editar</li> <li>C. O usuário altera os dados de interesse.</li> <li>D. O usuário clica em “Salvar”.</li> </ol>
<b>Fluxo Alternativo 4</b> <b>Desativar unidade.</b>	<ol style="list-style-type: none"> <li>A. no passo 3, o usuário seleciona uma unidade já existente dentre as listadas para consulta.</li> <li>B. O sistema pede a confirmação do usuário quanto a desativação da unidade.</li> <li>C. O usuário confirma a operação.</li> </ol>
<b>Fluxo Alternativo 5</b> <b>Consulta de unidade</b>	<ol style="list-style-type: none"> <li>A. no passo 3, o usuário seleciona uma unidade já existente dentre as listadas para consulta.</li> </ol>
<b>Pós-condições</b>	O Inclusão ou Modificação em dada unidade é processada pelo sistema. O usuário é redirecionado para a tela “Meus serviços”.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. No fluxo alternativo 3, passo C, o usuário informa dados de forma inadequada e clica em salvar. O sistema alerta o usuário dos campos que precisam de ajuste e volta ao passo C. O usuário corrige os dados para prosseguir ou escolhe a opção cancelar para interromper o caso de uso.</li> <li>2. No fluxo principal, passo 5, o usuário informou dados de forma inadequada. O sistema alerta o usuário dos campos que precisam de ajuste e volta ao passo 4. O usuário corrige os dados para prosseguir ou escolhe a opção cancelar para interromper o caso de uso.</li> </ol>

## 6.6. Atende Cliente

<b>Descrição</b>	O prestador de serviço que divulga serviços marcados para contratação na plataforma eventualmente recebe solicitação de atendimento por consumidores. O prestador recebe uma notificação do sistema, decide por atender (ou não), o atendimento ocorre e, no final, o prestador finaliza o atendimento.
<b>Atores</b>	Prestador de Serviço
<b>Precondições</b>	O prestador de serviço deve estar autenticado e o caso de uso Contrata Serviço foi previamente executado por um consumidor.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema emite uma notificação ao prestador de serviço de que há um consumidor interessado em ser atendido, exibindo dados do consumidor.</li> <li>2. O prestador confirma que vai atender.</li> <li>3. Após o término do atendimento, o prestador do serviço confirma a conclusão do atendimento.</li> </ol>
<b>Fluxo Alternativo 1</b> <b>Serviço Exige Encontro ao Cliente</b>	<ol style="list-style-type: none"> <li>A. O serviço cadastrado está marcado para exigir encontro ao cliente. Então no passo 2, após confirmação, o sistema exibe um mapa sinalizando a localização do usuário prestador e do usuário cliente.</li> <li>B. O usuário prestador (e/ou o usuário cliente) se locomove para o encontro acontecer.</li> <li>C. Os usuários chegam próximos um do outro. O sistema então emite um diálogo para ambos pedindo a confirmação do encontro.</li> <li>D. Ambos os usuários confirmam o encontro.</li> <li>E. O passo 3 do fluxo principal é executado.</li> </ol>
<b>Pós-condições</b>	O atendimento do serviço foi concluído. Um registro de ocorrência de atendimento foi armazenado para futuras consultas. Transações de pagamento com crédito são realizadas, se for o caso de pagamento desta modalidade.
<b>Casos de Uso Envolvidos</b>	<ul style="list-style-type: none"> <li>• Contrata serviço</li> </ul>
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. No passo 2 do fluxo principal, o prestador de serviço opta por não atender o cliente. O caso de uso é interrompido.</li> <li>2. no passo D, do fluxo alternativo 1, um dos usuários não confirma o encontro. O caso de uso é interrompido.</li> <li>3. no Passo 3, do fluxo principal, o prestador do serviço não confirma a conclusão do serviço. O caso de uso é interrompido.</li> </ol>

## 6.7. Entra em contato

<b>Descrição</b>	O consumidor entra em contato com outro prestador de serviço por
------------------	--

	meio de mensagens de texto - ou simplesmente por “chat”.
<b>Atores</b>	Consumidor e Prestador de Serviço.
<b>Precondições</b>	O consumidor deve estar autenticado e uma das seguintes condições foram contempladas: <ol style="list-style-type: none"> <li>1. O caso de uso Pesquisa Serviço foi executado e, neste, o usuário selecionou a opção Entrar em Contato.</li> <li>2. O consumidor ou prestador acionou o histórico de chats e selecionou uma conversa existente.</li> </ol>
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema exibe a tela de Conversa.</li> <li>2. O usuário digita mensagens que deseja enviar ao usuário destinatário e aciona o botão Enviar.</li> <li>3. O usuário destinatário é notificado da mensagem.</li> <li>4. O usuário destinatário responde as mensagens.</li> <li>5. Caso um dos usuários queiram enviar mais mensagens, o caso de uso retorna ao passo 3, senão é encerrado.</li> </ol>
<b>Pós-condições</b>	Não se aplica.
<b>Casos de Uso Envolvidos</b>	<ul style="list-style-type: none"> <li>• Pesquisa Serviços</li> </ul>
<b>Exceções</b>	Não se aplica

## 6.8. Consulta Atendimentos Realizados

<b>Descrição</b>	O usuário consulta as ocorrências de atendimentos já realizados, por serviço, por unidade, por data e outros critérios.
<b>Atores</b>	Usuário
<b>Precondições</b>	O usuário deve estar autenticado
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a opção Histórico de Atendimentos.</li> <li>2. O sistema exibe a lista de serviços que o usuário possui divulgados.</li> <li>3. O usuário seleciona o serviço de interesse para consulta.</li> <li>4. O sistema exibe a lista de atendimentos já realizados com o serviço selecionado</li> </ol>
<b>Pós-condições</b>	O sistema exibe uma lista de atendimentos já realizados pelo usuário para aquele serviço com os campos: cliente, valor pago, forma de pagamento, data e hora
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. No passo 1, após acessar a opção Histórico de Atendimentos, o usuário não possui serviços divulgados. O sistema exibe uma mensagem alertando a ausência de serviços e o caso de uso é interrompido.</li> <li>2. No passo 3, após selecionar um serviço para consulta, o usuário não possui atendimentos realizados para aquele serviço. O sistema o alerta disto e o caso de uso é interrompido.</li> </ol>

## 6.9. Gerencia Agenda de Atendimento

<b>Descrição</b>	O usuário prestador de serviço oferta uma agenda de atendimentos do seu serviço para controle pelo próprio usuário prestador para que os clientes possam agendar dia e horário que pretendem usufruir do serviço.
<b>Atores</b>	Usuário Premium
<b>Precondições</b>	O usuário prestador deve estar autenticado
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a opção “Meus serviços”</li> <li>2. O usuário seleciona o serviço de interesse para configurar a agenda</li> <li>3. O sistema exibe uma agenda (Calendário), com os dias do mês corrente.</li> <li>4. O usuário usa a opção “Ofertar Período”</li> <li>5. O sistema exibe um formulário com dois campos: data início e data término, horários disponíveis em cada dia da semana.</li> <li>6. O usuário informa os campos.</li> <li>7. O usuário clica em salvar.</li> </ol>
<b>Fluxo Alternativo 1</b> <b>Bloquear Período</b>	<ol style="list-style-type: none"> <li>A. No passo 4, o usuário deseja bloquear um período para não permitir agendamento pelos clientes. Então ele usa outra opção “Bloquear Período”.</li> <li>B. O sistema exibe um formulário com dois campos: data início e data término.</li> <li>C. O usuário informa os dois campos.</li> <li>D. O usuário clica em Bloquear.</li> </ol>
<b>Fluxo Alternativo 2</b> <b>Consulta de Agendamentos</b>	<ol style="list-style-type: none"> <li>A. No passo 4, o usuário deseja apenas consultar todos os agendamentos realizados, então utiliza a opção “Listar Agendamentos”.</li> <li>B. O sistema exibe uma listagem de atendimentos realizados.</li> </ol>
<b>Pós-condições</b>	As informações na agenda do serviço, quando alteradas, são salvas no sistema.
<b>Casos de Uso Envolvidos</b>	Não se aplica
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. No fluxo alternativo 1, o usuário escolheu um período totalmente não divulgado para bloquear. O sistema exibe uma mensagem alertando que não é possível bloquear um período que não está ofertado.</li> </ol>

## 6.10. Gerencia Caixa

<b>Descrição</b>	O usuário controla fluxos de caixa de cada serviço que oferta ou de todos os serviços conjuntos, realizando abertura e fechamento de caixa a qualquer momento.
------------------	--

<b>Atores</b>	Usuário Premium
<b>Precondições</b>	O usuário deve estar autenticado
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a opção “Caixa”</li> <li>2. O sistema exibe o valor do caixa atual, data e hora de abertura, data e hora de fechamento(se fechado) e estado atual do caixa: aberto ou fechado.</li> <li>3. O usuário seleciona a opção “abrir caixa”.</li> <li>4. O sistema exibe um campo para informar o valor inicial do caixa.</li> <li>5. O usuário informa o valor inicial e confirma a abertura.</li> <li>6. O sistema realiza a abertura do caixa e redireciona o usuário para a tela de caixa com seu estado como aberto.</li> </ol>
<b>Fluxo Alternativo 1</b> <b>Fechar Caixa</b>	<ol style="list-style-type: none"> <li>A. No passo 3, o caixa está fechado e por isso o sistema exibe apenas a opção fechar caixa. O usuário seleciona esta opção.</li> <li>B. O sistema exibe as transações realizadas desde a última abertura de caixa, discriminado os dados de cada uma e exibindo o valor total.</li> <li>C. O usuário confirma o fechamento do caixa.</li> <li>D. O sistema redireciona o usuário para a tela do Caixa, exibindo o estado do caixa como fechado.</li> </ol>
<b>Pós-condições</b>	O sistema armazena a informação de abertura/fechamento do caixa.
<b>Casos de Uso Envolvidos</b>	Não se aplica.
<b>Exceções</b>	<ol style="list-style-type: none"> <li>1. No fluxo principal, passo 5, o usuário não informa o valor e aciona a abertura. O sistema exibe um alerta solicitando que o usuário preencha o campo. O usuário informa o valor para prosseguir ou usa a opção Cancelar para interromper o caso de uso.</li> <li>2. No fluxo alternativo 1, passo C, o usuário aciona a opção Cancelar, então o caso de uso é interrompido.</li> </ol>

### 6.11. Emite Nota Fiscal

<b>Descrição</b>	O sistema emite notas fiscais informando os dados para a nota ou selecionando um registro de atendimento de serviço já realizado.
<b>Atores</b>	Usuário
<b>Precondições</b>	O usuário deve estar autenticado
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário seleciona a opção “Nota Fiscal”</li> <li>2. O sistema exibe um formulário com campos para serem digitados: serviço(s) ofertado(s), preço, data, cliente.</li> <li>3. O usuário preenche as informações e clica em emitir nota fiscal.</li> <li>4. O sistema aciona a opção de impressão, com opção de salvar em arquivo PDF.</li> </ol>

<b>Pós-condições</b>	O sistema emitiu a nota fiscal para o sistema de impressão do dispositivo do usuário ou gerou um arquivo PDF da nota fiscal.
<b>Casos de Uso Envolvidos</b>	Não se aplica
<b>Exceções</b>	1. No passo 2, o usuário não preencheu todas as informações obrigatórias para gerar a nota fiscal. O sistema alerta o usuário do que ainda precisa ser preenchido. O usuário preenche o que falta para prosseguir ou usa a opção cancelar para interromper o caso de uso.

## 6.12. Personaliza Serviços

<b>Descrição</b>	O usuário prestador deseja personalizar(modelar) os clientes que farão uso de seus serviços e coletar informações adicionais. O usuário configura um formulário que deve ser preenchido por seus clientes no primeiro uso dos serviços que disponibiliza e outro no ato de cada contratação dos serviços.
<b>Atores</b>	Usuário
<b>Precondições</b>	O usuário prestador deve estar autenticado, ter acessado a opção “Meus serviços” e selecionado um serviço existente.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O usuário aciona a opção “Personalizar Serviço”.</li> <li>2. O sistema exibe uma tela com opções: criar pré cadastro de clientes, opções para contratação de serviços.</li> <li>3. O usuário utiliza a opção “criar pré cadastro de clientes”.</li> <li>4. O sistema exibe uma tela para que seja desenhado um formulário de pré cadastro de clientes do serviço.</li> <li>5. O usuário modela o formulário de pré cadastro.</li> <li>6. O usuário clica em salvar.</li> </ol>
<b>Fluxo Alternativo 1 Configuração de formulário de contratação do serviço</b>	<ol style="list-style-type: none"> <li>A. No passo 5, o usuário quer criar configurar um formulário que pedirá informações do cliente no ato da contratação do serviço, então ele aciona a opção “configurar formulário de contratação do serviço”.</li> <li>B. O sistema exibe uma tela para que seja modelado um formulário de contratação do serviço.</li> <li>C. O usuário modela o formulário.</li> <li>D. O usuário clica em Salvar.</li> </ol>
<b>Pós-condições</b>	Não se aplica.
<b>Casos de Uso Envolvidos</b>	Não se aplica.
<b>Exceções</b>	Não se aplica.

## 7. Requisitos não funcionais

### 7.1 Segurança

O sistema deve proteger as informações dos usuários, principalmente no que se diz respeito a dados de cartões de crédito cadastrado e senha de acesso a plataforma, com técnicas de criptografia e armazenamento seguro e confiável dos bancos de dados. Para ter acesso a plataforma, todo usuário precisa informar seus dados de acesso (login e senha) ou autorizar a rede social Facebook a usar sua conta para ter acesso a aplicação.

## **7.2. Desempenho**

As operações mais básicas como busca de serviços, criação de serviço, entrar em contato, devem ter desempenho rápido, com respostas em até 2 segundos para cada interação com o usuário. Informações muito utilizadas pelo usuário devem ser armazenadas no dispositivo do mesmo para evitar requisições repetitivas e desnecessárias ao servidor da aplicação, aumentando o desempenho na utilização destes recursos.

## **7.3 Interface**

O subsistema aplicação deve possuir interface moderna, responsiva e de fácil utilização para usuários de todos os tipos, até para os mais leigos. Interfaces devem estar de acordo com os interesses e objetivos principais dos usuários, com atalhos de recursos mais recorrentes, facilitando a interação.

## **7.4 Portabilidade**

A aplicação móvel deve estar disponível em multiplataformas - *android* e *IOS*.

## **8. Restrições de software**

A aplicação deve ser desenvolvida com as tecnologias mais convencionais para aplicações móveis multiplataformas (código fonte único), padrões de projetos que permitam a total flexibilidade da aplicação servidora para servir diversos tipos de aplicações cliente.

A linguagem de programação a ser utilizada no servidor e clientes é única: javascript. Essa mesma linguagem se estende ao banco de dados, que será orientado a documentos, o que facilita a integração das camadas do sistema como um todo, otimizando tráfego de dados entre as camadas.

**APÊNDICE B – DOCUMENTO DE ESPECIFICAÇÃO DE ANÁLISE****PROJETO ISERVICE  
DOCUMENTO DE ESPECIFICAÇÃO DE ANÁLISE**

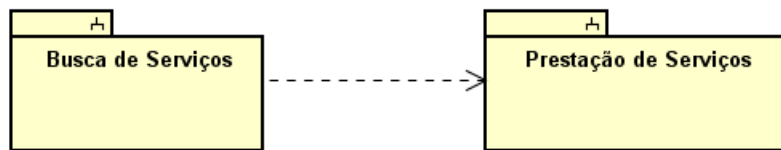
<b>Histórico de Modificações</b>		
<b>Data</b>	<b>Autor</b>	<b>Descrição</b>
28/05/2017	Ewertom Moraes	Definição de Lista de Classes e seus Atributos.
15/06/2017	Ewertom Moraes	Definição do Diagrama de classes
30/06/2017	Ewertom Moraes	Definição de Diagramas de Sequência e de Estados

**1. Introdução**

Este documento objetiva definir a análise do projeto de software IService. Esta análise será do tipo orientada a Objetos. Serão definidas classes de objetos, interações e estados no ciclo de vidas destes objetos, com técnicas de modelagem UML na versão 2.0, além do dicionário de dados que existirão na aplicação.

**2. Modelo Lógico****2.1 Diagrama de Pacotes**

A Aplicação conterá dois subsistemas: busca de serviço, prestação de serviços. Esses subsistemas conversam entre si, sendo um dependente do outro. O subsistema busca de serviços é dependente do de prestação de serviço, pois o primeiro requer uma funcionalidade do segundo – caso de uso atende cliente.

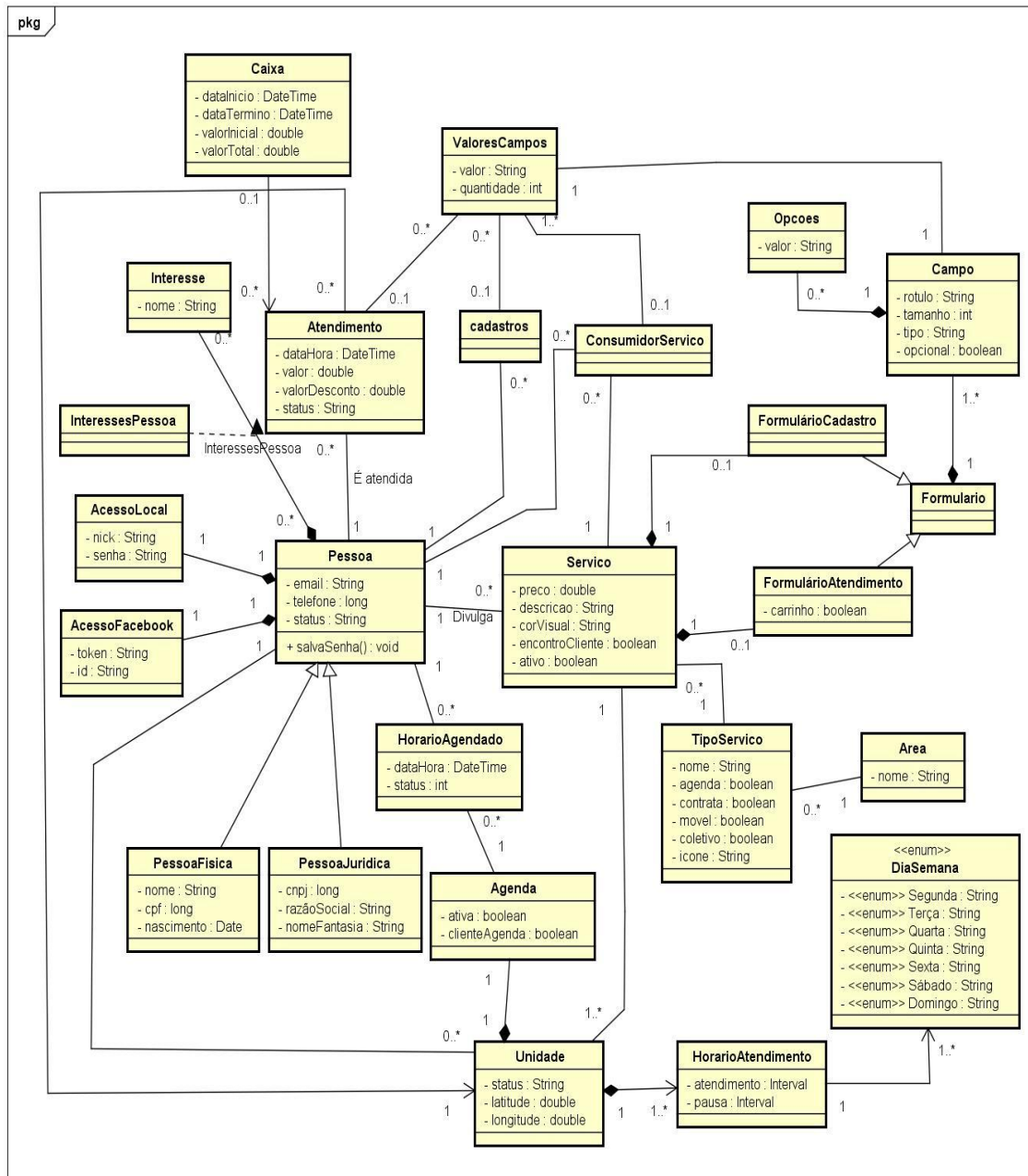


## 2.1 Diagrama de classes

O diagrama de classe define o modelo de dados de cada Objeto dentro do sistema, e seus relacionamentos.

As classes aqui definidas são do modelo de negócio da aplicação, ou seja, aquelas que representam dados de interesse, que contemplem os requisitos, a representação e modelagem para persistência de dados que contemplem os casos de uso e funcionalidades levantadas na etapa de especificação de requisitos.

Complementando, aqui não são definidos métodos no diagrama do tipo: cadastrar, alterar, deletar, listar. Esses métodos serão comuns a qualquer modelo de dados - classe de negócio. Portanto, são omitidos no diagrama. Ainda, outros métodos que contemplarão regras de negócio dos modelos (como validar se uma pessoa está disponível ou não para atender cliente), serão parte de outras classes complementares de cada modelo que se faça necessário implementar regras de negócio específicas, o que constará na etapa de projeto. Ao contrário dos métodos, atributos de negócio, nesta etapa, foram todos devidamente declarados no diagrama.



### 3. Dicionário de dados

O dicionário de dados exibe os atributos que cada classe de modelo, do diagrama de classes, deve ter.

Algumas marcações são utilizadas no dicionário, contendo significados específicos. O quadro abaixo explica cada um destes:

Símbolo	Significado
@	Identificador do registro na coleção
[A, B, C]	Escolher dentre uma das opções separadas por vírgula

{NomeCampo}	Nome do campo entre Chaves . Identificam que o campo possui múltiplas ocorrências. Uma lista de valores do atributo.
-------------	---

### Tipo

Os tipos de dados podem ser um dos listados na tabela abaixo, ou uma outra classe existente. Neste último caso, sinalizando que o tipo do atributo é do tipo da classe definida (Composição, Agregação).

Símbolo do Tipo	Significado
C	Caracteres. Pode ser uma mistura de qualquer tipo de caracteres, letras, números símbolos.
N	Numérico. Números inteiros ou reais. O seu tamanho, quando necessário, será definido entre parênteses, no formato (Casas Inteiras, Casas Decimais)
D	Data
DT	Data e Hora
T	Hora
NomeDeClasse	atributo do tipo da classe definida.
B	Valor booleano, aceitando valores Verdadeiro e Falso

PessoaFisica				
Descrição da Coleção	Registro de pessoas físicas.			
Nome	Descrição	Tipo	Opcional	Informações Complementares
@id	Identificador do registro na coleção	C		gerado automaticamente
nome	nome da pessoa física	C		
cpf	número do cpf	N	X	
nascimento	data de nascimento	D	X	
email	endereço de email	C		
telefone	número de telefone com ddd e	N	X	

	identificador de país			
acessoLocal	Dados de acesso local usados para realizar login na aplicação informando login e senha.	AcessoLocal		
acessoFacebook	dados de acesso que o facebook fornece para identificar o perfil da pessoa no facebook.	AcessoFacebook	X	

<b>PessoaJuridica</b>				
<b>Descrição da Coleção</b>	<b>Registro de Pessoas Jurídicas.</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
razaoSocial	nome da pessoa física	C		
nomeFantasia	Nome fantasia da pessoa jurídica	C		
cnpj	número do cnpj	N		
email	endereço de email	C		
telefone	número de telefone com ddd e identificador de país	N	X	
acessoLocal	Dados de acesso local usados para realizar login na aplicação informando login e senha.	AcessoLocal		
acessoFacebook	dados de acesso que o facebook fornece para identificar o perfil da pessoa no facebook.	AcessoFacebook	X	

<b>Servico</b>				
<b>Descrição da Coleção</b>	São registros de Divulgação de Serviços que Pessoas Físicas e Jurídicas podem cadastrar. Este serviço, em resumo, possui um tipo de serviço, um valor cobrado, uma descrição e unidades de atendimento.			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
peessoa	peessoa que presta o serviço (física ou jurídica)	Pessoa		
preco	Preço do Serviço	N(11,2)		
tipoServico	Tipo de serviço	TipoServico		
descricao	Descrição do serviço	C		
{atendimentos}	Lista de Registros de atendimento realizados pelo serviço	Atendimento	X	
{unidades}	Lista de unidades do serviço	UnidadeMovel		
cor	Cor a utilizar como plano de fundo da interface/tela do serviço	N	X	[1-Azul,2-Vermelho, 3-Cinza, 4-Preto, 5-Laranja, 6-Verde]
agenda	Agenda de atendimento do Serviço	Agenda	X	
{caixa}	Lista de registros de caixa do serviço	Caixa	X	
formCadastro	Formulário de Cadastro de Cliente para solicitar na primeira vez que um certo	FormularioCadastro	X	

	cliente for utilizar o serviço			
formAtendimento	Formulário de Atendimento para ser solicitado ao cliente toda vez que for utilizar o serviço	FormularioAtendimento	X	

<b>Atendimento</b>				
<b>Descrição da Coleção</b>	Ocorrências de atendimentos de serviço. No momento que uma pessoa pede para ser atendida por alguém, para usufruir um serviço qualquer, e o responsável pelo serviço/unidade aceitar atender, um registro de Atendimento é Criado, identificado as duas pessoas (prestador, cliente), a unidade do serviço onde será feito o atendimento, etc. Se o serviço será do tipo agendado, este atendimento fica com este status, esperando para ser realizado numa data futura			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
data	Data e Hora do Serviço atendido	DT		
unidade	Unidade a qual o serviço foi prestado	Unidade		
pessoaAtendida	Pessoa que foi atendida pela prestação do serviço	Pessoa		
valor	valor cobrado pelo serviço	N(11,2)		
valorDesconto	total de descontos	N(11,2)	X	
horarioAgendado	Horário Agendado	HorarioAgendado	X	
{valoresCampos}	Valores dos Campos de formulários do Serviço que foram configurados para serem pedidos no atendimento	ValoresCampos	X	

Unidade				
<b>Descrição da Coleção</b>	Registro de unidade de atendimento do serviço. Na unidade são armazenados dados do local, como a localização(quando fixa), o nome da unidade, horários de atendimento.			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
nome	nome da unidade de atendimento	C		
responsavel	Pessoa cadastrada responsável pelo atendimento do serviço na Unidade.	Pessoa		
latitude	Informação de localização	N(2,11)		
longitude	Informação de localização	N(2,11)		
{horarioAtendimento}	Registros de horário de atendimento	HorarioAtendimento		deve ter a partir de 1 registro
agenda	Registro de controle por horário agendado	Agenda	X	

TipoServico				
<b>Descrição da Coleção</b>	Tipo de Serviço, o qual cada serviço deve ter(como técnico de informática, limpeza de central de ar, costureira, babá, etc)			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
nome	nome do tipo de serviço	C		

contrata	Se o serviço deste tipo pode ser contratado pelo aplicativo	B		
Agenda	Se o serviço deste tipo pode ser agendado um atendimento pela plataforma	B		
Coletivo	Se o tipo de serviço é um atendimento somente coletivo, ou seja, pode ser acionado mas não fica ocupado nem há fluxo de atendimento	B		
Movel	Se o serviço, poderá possuir localização móvel ou não.	B		

Area				
<b>Descrição da Coleção</b>	Área do Tipo de Serviço(como Saúde, Informática, automotiva, etc)			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
nome	nome da área	C		

HorarioAtendimento				
<b>Descrição da Coleção</b>	Serve para informar um intervalo de horas do dia para atendimento em uma unidade. Além do horário de atendimento, cruza esta informação com um possível intervalo de hora de pausa(interrupção da disponibilidade do atendimento). Estes horários podem estar associados de um, até todos os dias da semana.			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente

{atendimento}2	intervalo de horas que determina o horário de atendimento	T		
{pausa}2	intervalo de horas que determina uma pausa no horário de atendimento	T		
{diasSemana}7	dias da semana do horário de atendimento	C		

Agenda				
Descrição da Coleção	Controle de Agenda de horários agendados de unidade.			
Nome	Descrição	Tipo	Opcional	Informações Complementares
@id	Identificador do registro na coleção	C		gerado automaticamente
ativa	se o controle de atendimento por agenda está ativado para esta unidade	B		

HorárioAgendado				
Descrição da Coleção	Horários agendados para futuros atendimentos em uma determinada unidade.			
Nome	Descrição	Tipo	Opcional	Informações Complementares
@id	Identificador do registro na coleção	C		gerado automaticamente
peessoa	peessoa solicitante para utilizar o serviço	Pessoa		
unidade	Unidade para onde está sendo agendado	Unidade		
dataHora	data e hora do agendamento	DT		

status	status atual do horário agendado.	N		[Aberto, Fechado, Cancelado]
--------	-----------------------------------	---	--	------------------------------

<b>AcessoLocal</b>				
<b>Descrição da Coleção</b>	Identificação de pessoa para acesso à aplicação por meio de login e senha.			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
login	nome de usuário único para acesso ao sistema	C		
senha	palavra chave para acessar o sistema junto ao login	C		

<b>AcessoFacebook</b>				
<b>Descrição da Coleção</b>	Para acesso a aplicação por meio da rede social facebook.			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
token	valor fornecido pela rede social facebook	C		
identificador	valor fornecido pela rede social facebook	C		

<b>Interesse</b>				
<b>Descrição da Coleção</b>	Palavras chaves que categorizam interesses de pessoas na aplicação			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>

@id	Identificador do registro na coleção	C		gerado automaticamente
nome	título do interesse	C		

<b>InteressesPessoa</b>				
<b>Descrição da Coleção</b>	Identifica interesses que cada pessoa possa ter			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
pessoa	Pessoa que possui o interesse	Pessoa		
interesse	Interesse associado a pessoa	Interesse		

<b>Caixa</b>				
<b>Descrição da Coleção</b>	Registra um fluxo de caixa das movimentações financeiras de atendimentos, com controle de data de início e término do caixa,			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
{atendimentos}	lista de atendimentos os quais o caixa controla	Atendimento		
dataAbertura	data e hora de abertura do caixa	DT		
dataFechamento	data e hora de fechamento do caixa	DT		
valorAbertura	valor de abertura, valor inicial	N(11,2)		
valorFechamento	valor de fechamento, total do caixa	N(11,2)		

<b>FormularioCadastro</b>				
<b>Descrição da Coleção</b>	Formulário que é solicitado pela pessoa somente na primeira vez que irá utilizar um certo serviço. O prestador do serviço configura um conjunto de campos que devem ser solicitados para que o cliente o preencha.			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
1{campos}	lista de campos que este formulário possui configurado	Campo		
servico	serviço para o qual o formulário é utilizado	Servico		

<b>FormularioAtendimento</b>				
<b>Descrição da Coleção</b>	Formulário que é solicitado ao cliente toda vez que for utilizar/solicitar um serviço			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente
1{campos}	lista de campos que este formulário possui configurado	Campo		
servico	serviço para o qual o formulário é utilizado	Servico		

<b>Campo</b>				
<b>Descrição da Coleção</b>	Campo que é configurado para um formulário de cadastro ou de atendimento.			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Opcional</b>	<b>Informações Complementares</b>
@id	Identificador do registro na coleção	C		gerado automaticamente

formulario	Formulário ao qual o campo pertence	Formulario		
rotulo	rótulo exibido ao usuário que irá preencher o campo	C		
tamanho	tamanho máximo do valor que deverá ser preenchido pelo usuário	N	X	
tipo	tipo de campo	C		[Texto, Número, Valor, Discreto]
opcional	configura para que o campo seja de preenchimento opcional pelo usuário no ato do preenchimento do formulário	B	X	
preco	Para formulários que sejam carrinhos de compras, o campo pode representar um produto em cardápio. Logo, este campos serve para configurar o preço do campo/produto.	N		

Opcao				
Descrição da Coleção	Lista de opções para campos que são do tipo discreto			
Nome	Descrição	Tipo	Opcional	Informações Complementares
@id	Identificador do registro na coleção	C		gerado automaticamente
identificador	valor que identifica a opção	C		
rotulo	valor exibido ao usuário como opção do campo discreto	C		
campo	Campo ao qual a	Campo		

	opção pertence			
--	----------------	--	--	--

ValoresCampos				
Descrição da Coleção	Valores preenchidos de campos de formulários configurados para serem solicitados em atendimentos e no primeiro uso de um serviço por uma pessoa			
Nome	Descrição	Tipo	Opcional	Informações Complementares
@id	Identificador do registro na coleção	C		gerado automaticamente
valor	valor do campo	C		
atendimento	registro de atendimento para quando o valor é de um campo de um formulário do tipo de atendimento	Atendimento	X	
cadastro	registro de cadastro de pessoa para quando o valor é de um campo de formulário do tipo cadastro	Cadastro	X	

Cadastro				
Descrição da Coleção	Ocorrências de preenchimento de formulários cadastro			
Nome	Descrição	Tipo	Opcional	Informações Complementares
@id	Identificador do registro na coleção	C		gerado automaticamente
pessoa	pessoa que preencheu o formulário de cadastro	Pessoa		
1{valoresCampos}	lista de valores de campos que foram preenchidos no formulário, pela pessoa	ValoresCampos		

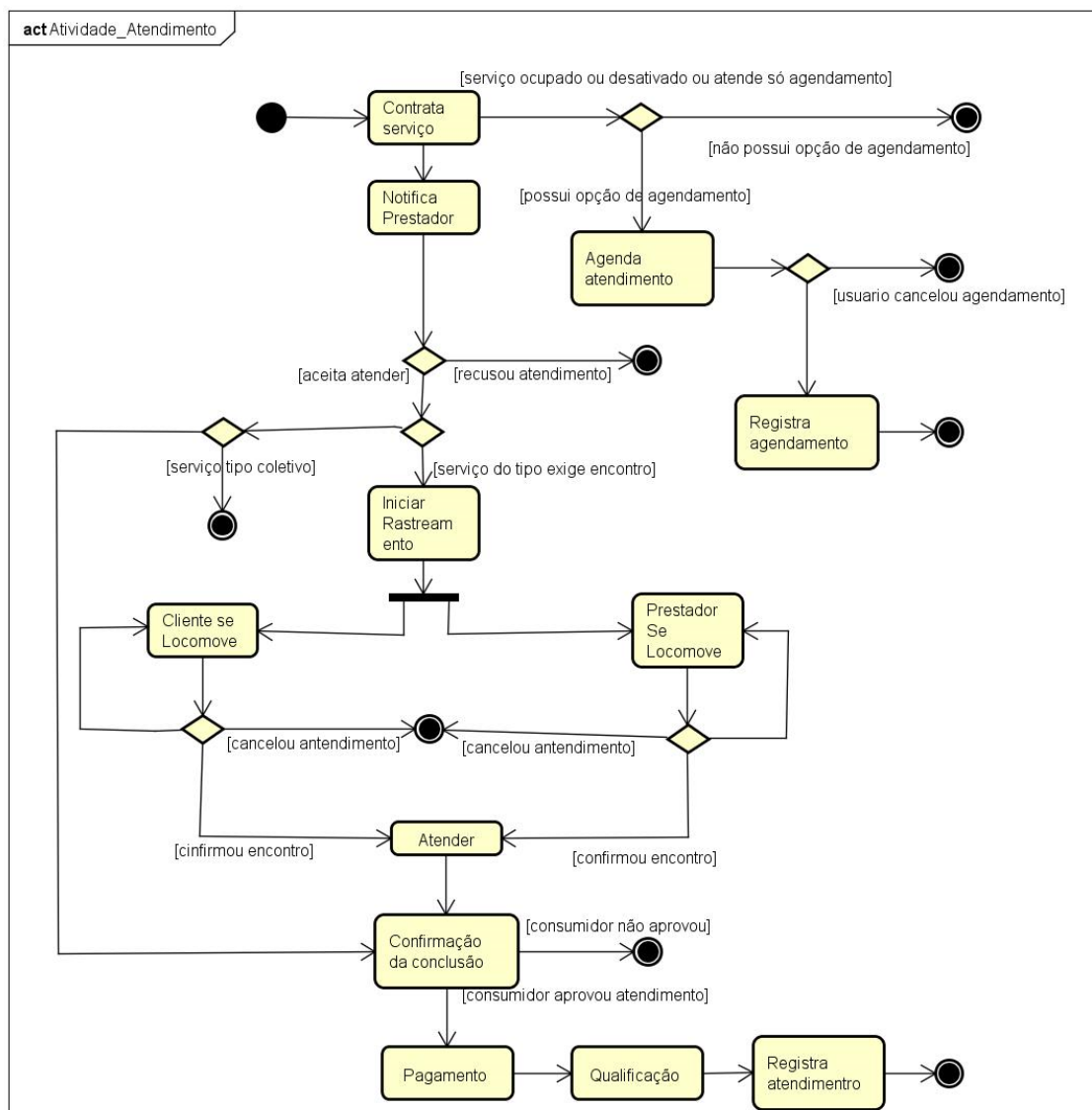
## 4. Modelo dinâmico

Para demonstrar o comportamento dos objetos e descrever ações do sistema, serão utilizados os diagramas de Sequência(para descrever ações e comunicações entre objetos nas ações), diagramas de Estados(mostrando as mudanças de estados de objetos que possuem estados diferentes durante seus ciclos de vida) e diagrama de Atividades para uma das principais atividades do sistema, que é o atendimento com encontro entre as partes.

### 4.1 Diagrama de atividades

Este diagrama é do caso de uso “Atende Cliente.

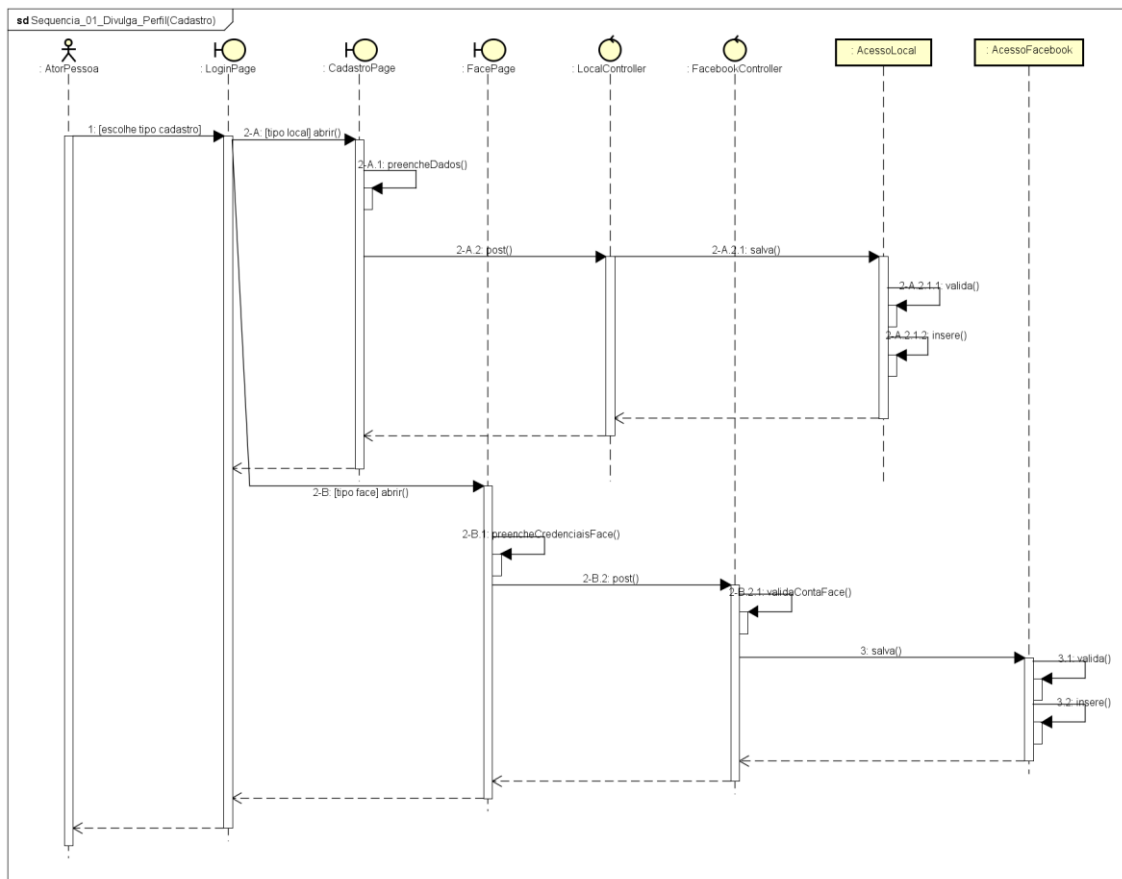
A atividade inicia quando o usuário contratante do serviço resolve acionar a opção Contratar, de um serviço qualquer disponível e divulgado por outro usuário.



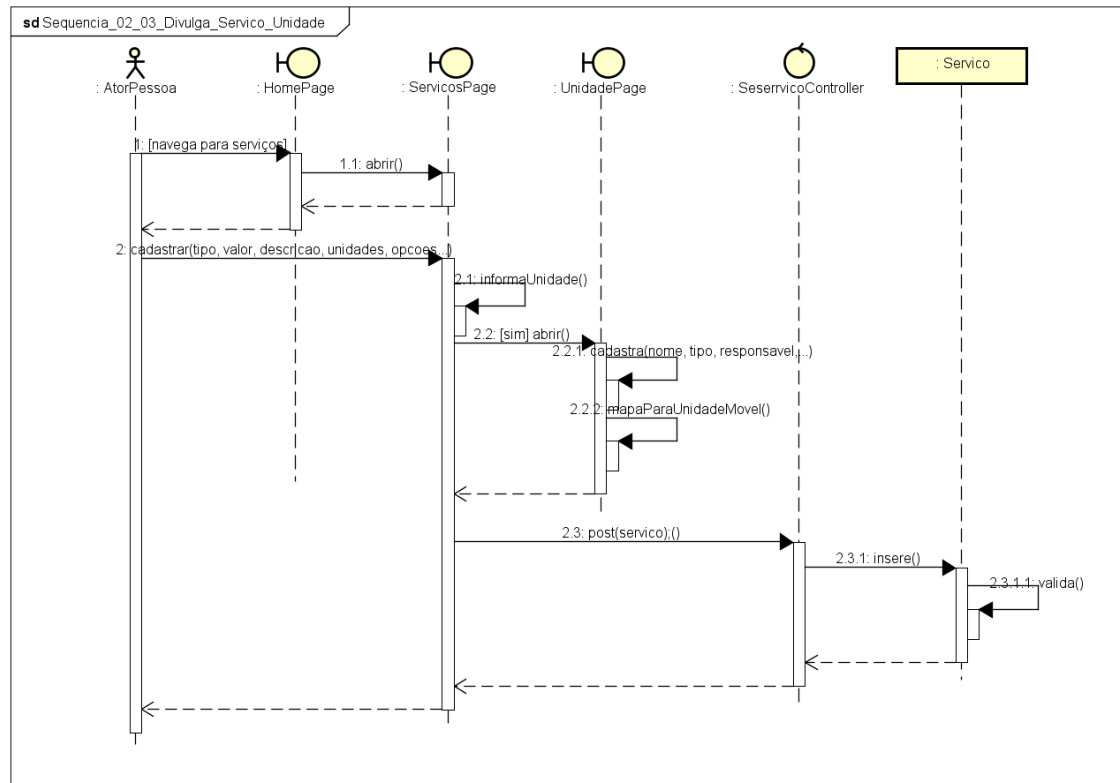
## 4.2 Diagrama de Interação

Para documentar a interação entre objetos, o diagrama de sequência foi adotado. A seguir, os diagramas de sequência para cada caso de uso, alguns englobando mais de um caso de uso.

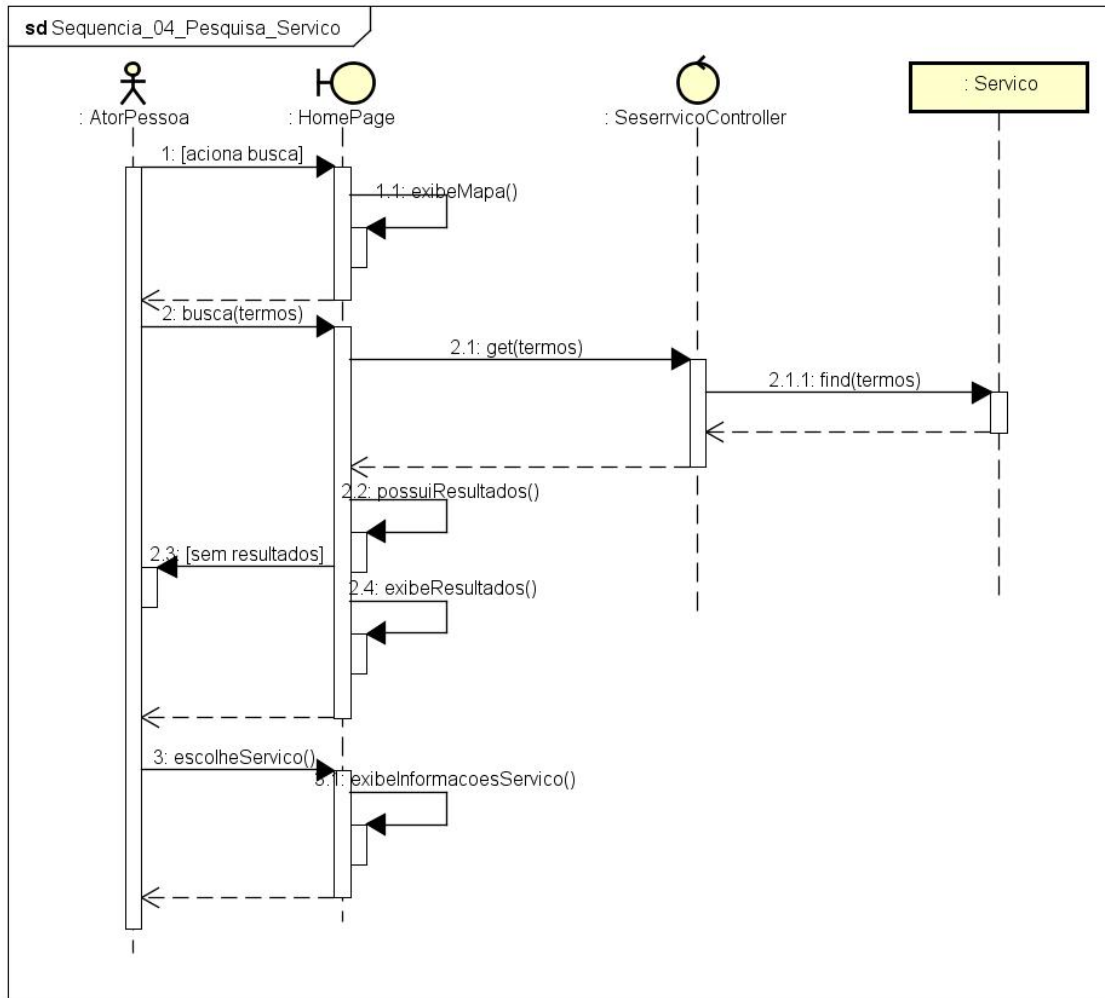
### 4.2.1 Divulga Perfil / Cadastro



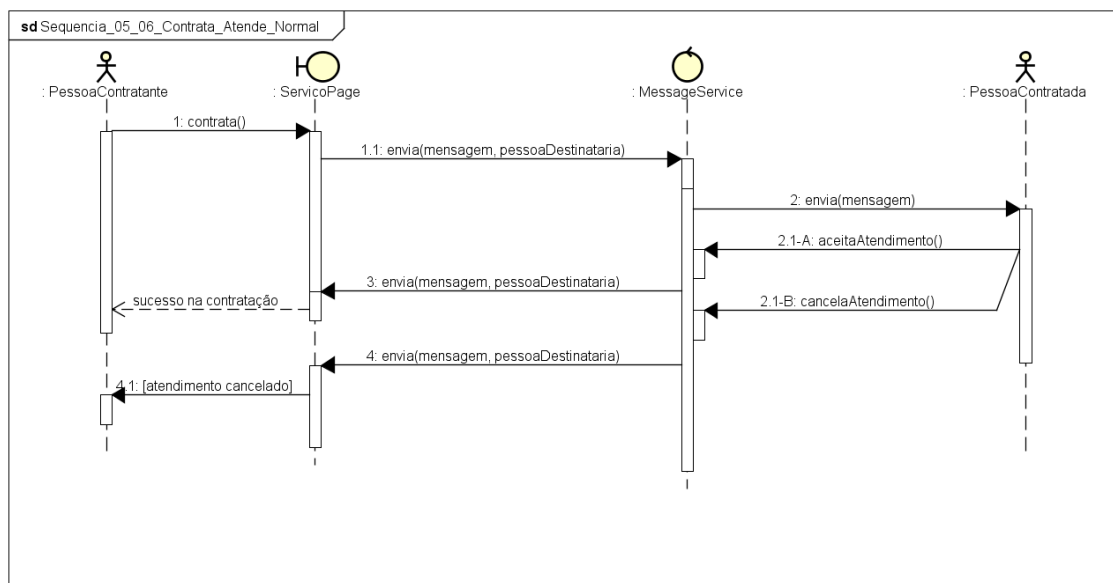
### 4.2.2 Divulga Serviço / Unidade



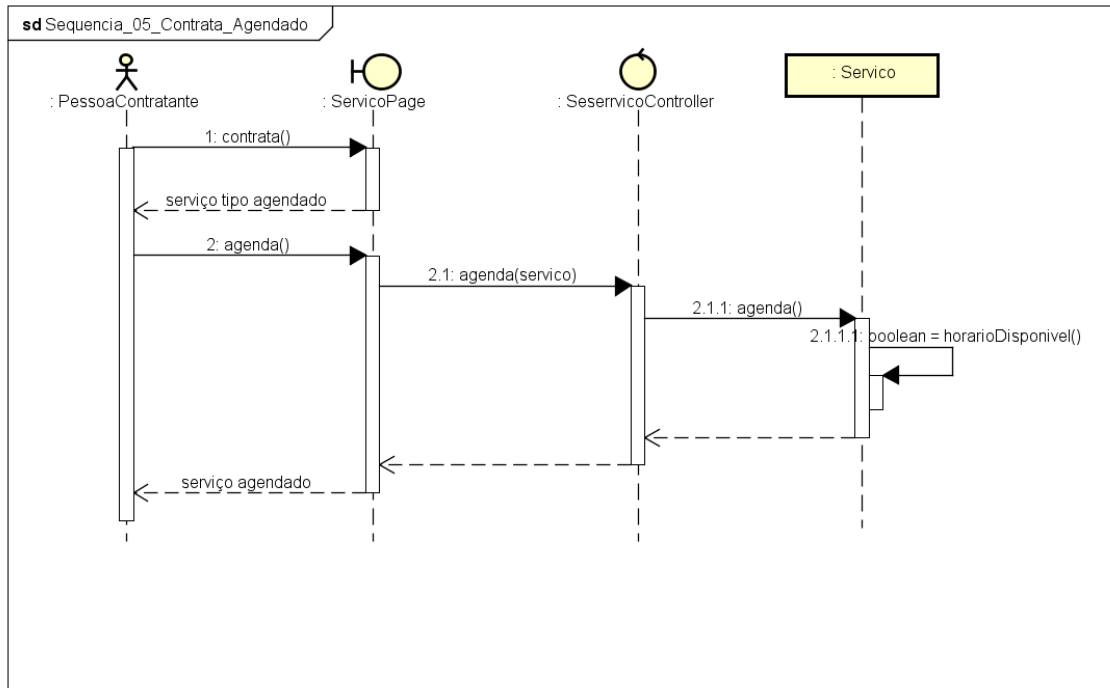
### 4.2.3 Pesquisa Serviço



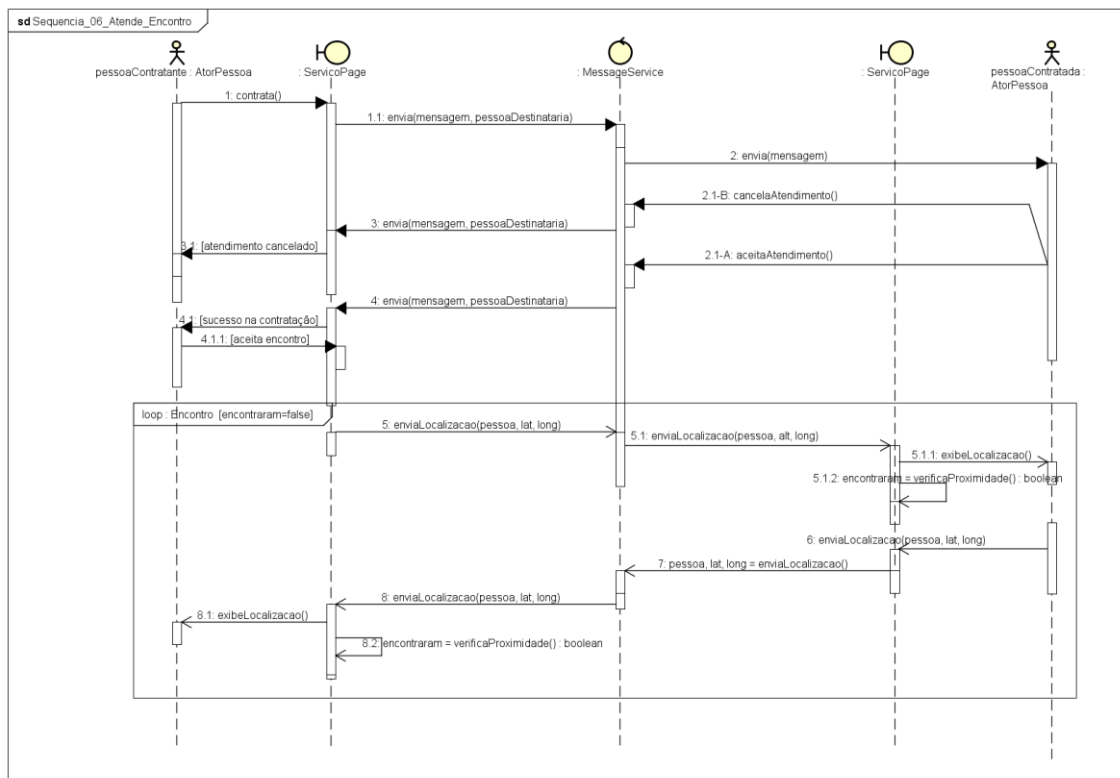
#### 4.2.4 Contrata / Atende - Padrão



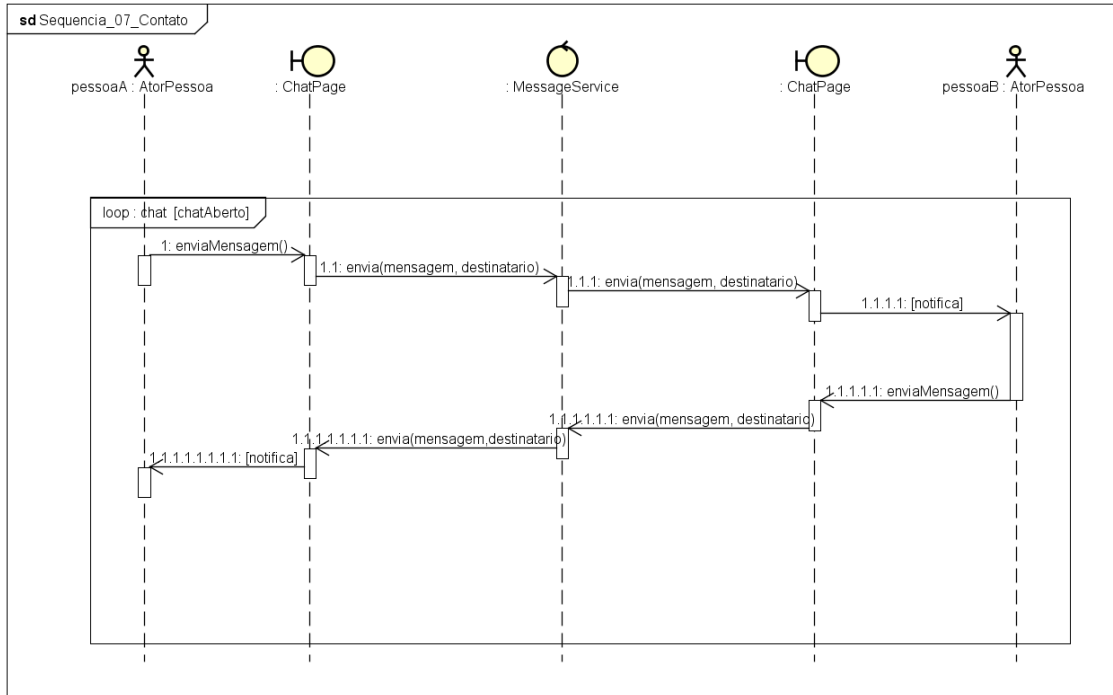
#### 4.2.5 Contrata / Atende – Agendado



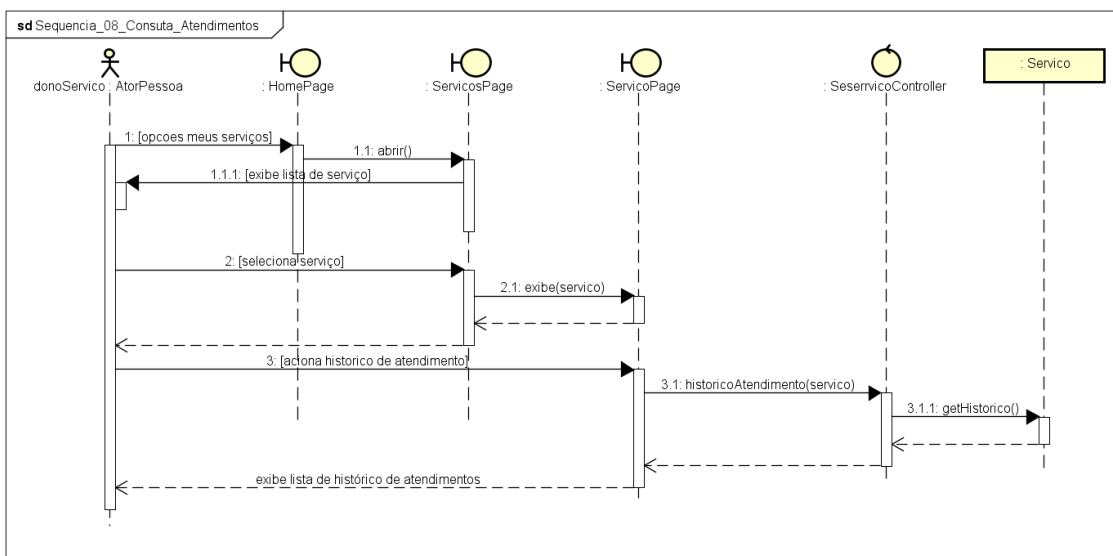
#### 4.2.6 Contrata / Atende - Com Encontro



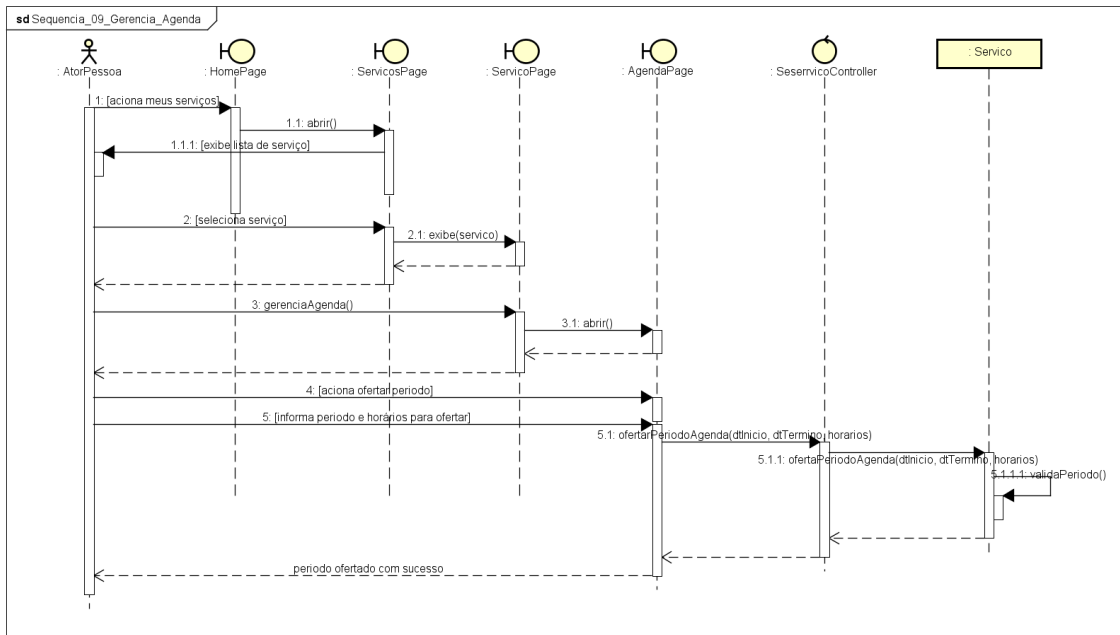
#### 4.2.7 Entrar em Contato



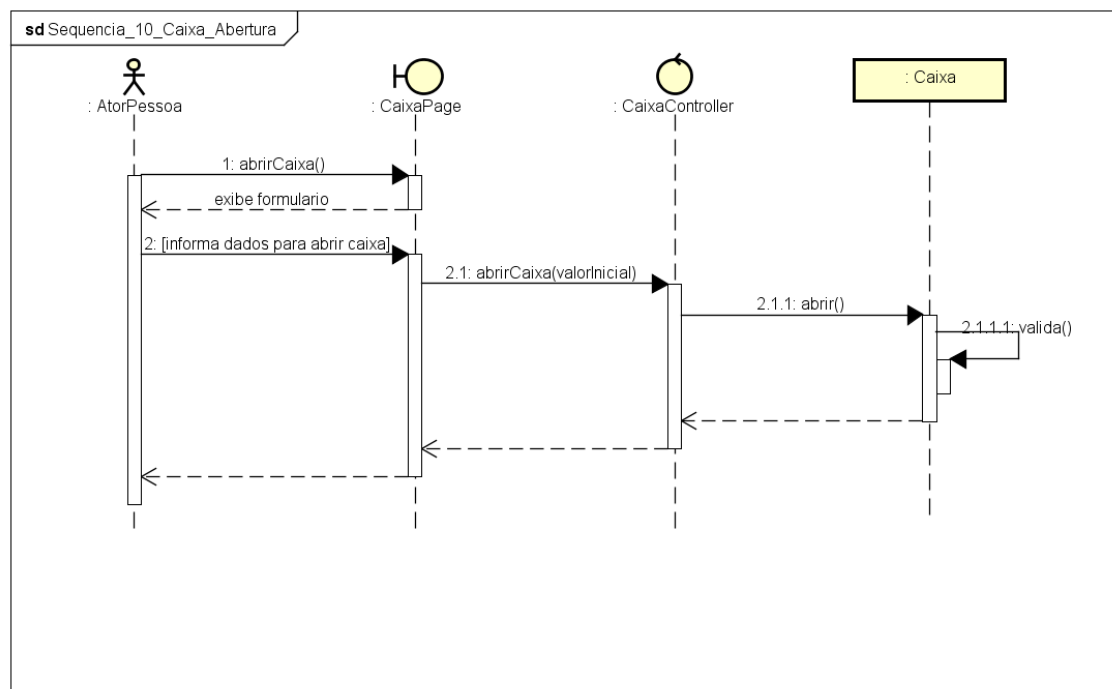
### 4.2.8 Consulta Histórico de Atendimentos



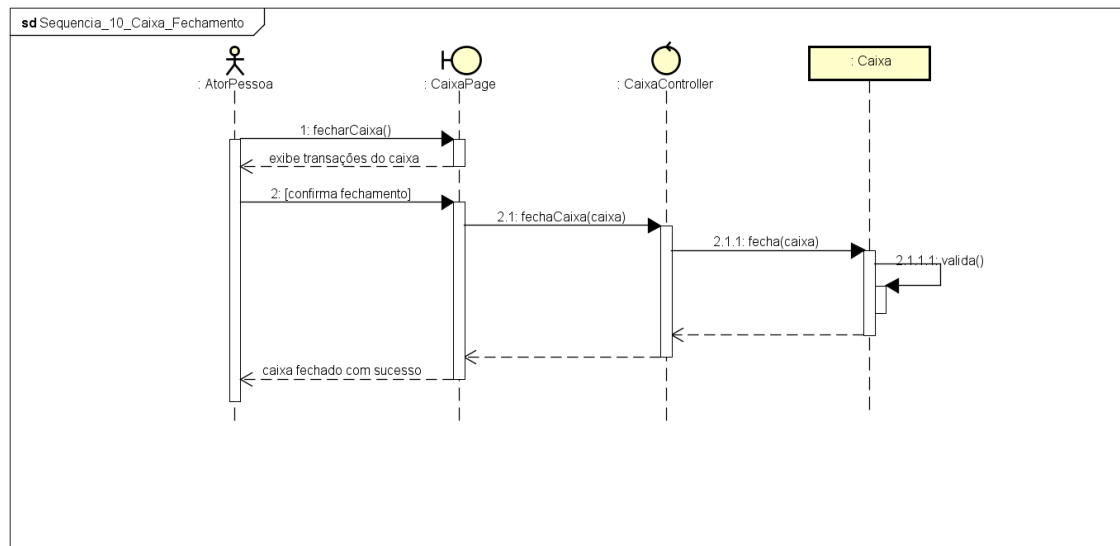
### 4.2.9 Gerencia Agenda



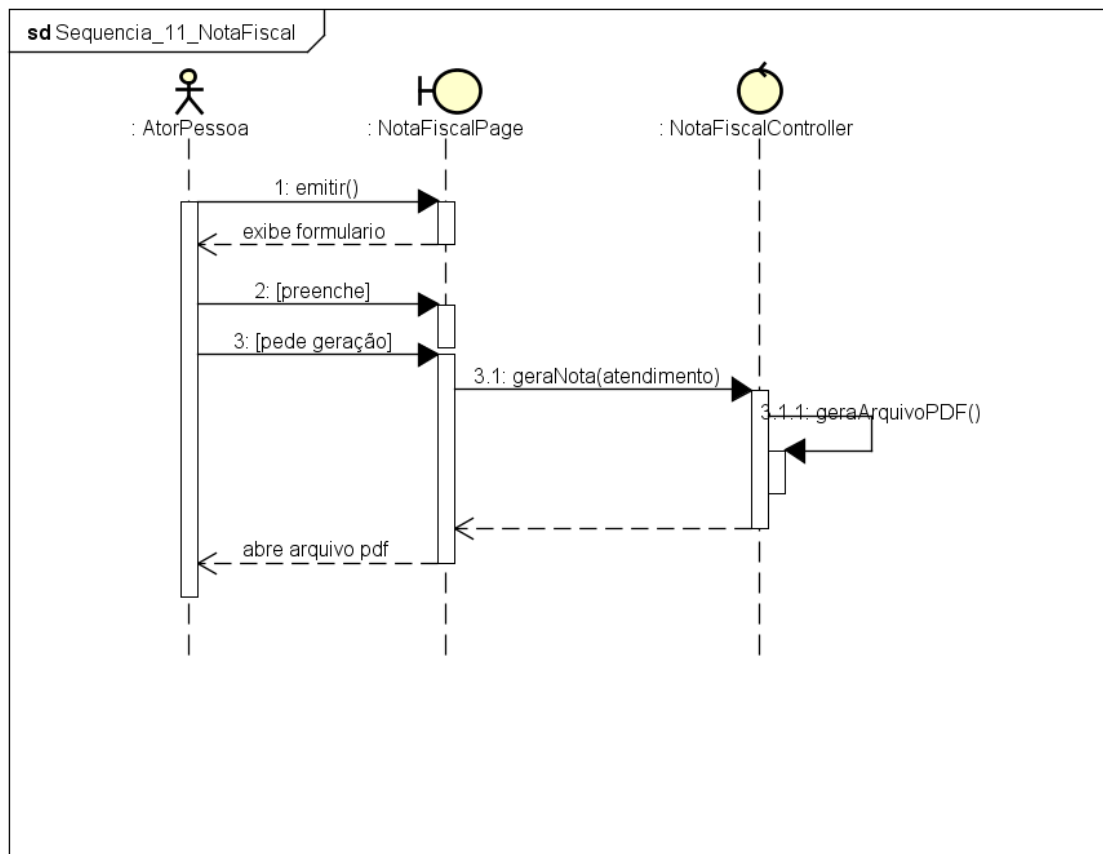
**4.2.10. Gerencia Caixa – Abertura**



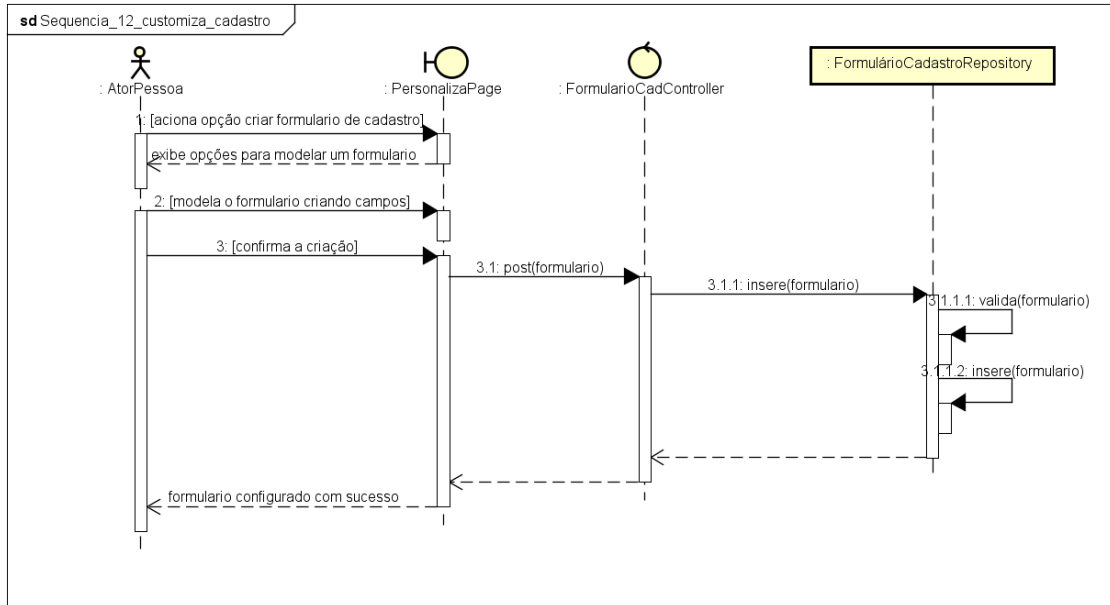
**4.2.11. Gerencia Caixa – Fechamento**



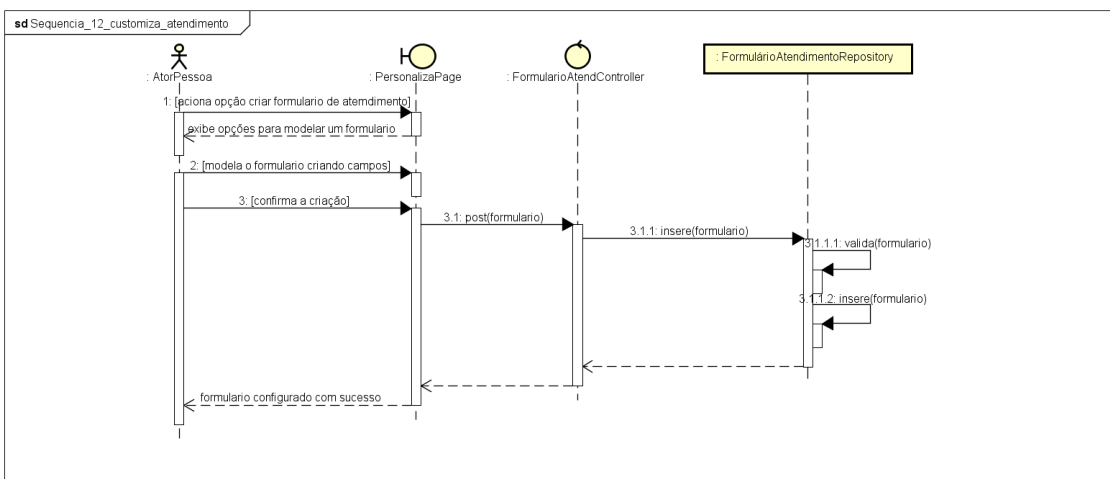
#### 4.2.12. Emite Nota Fiscal



#### 4.2.13. Personaliza - Formulário de Cadastro



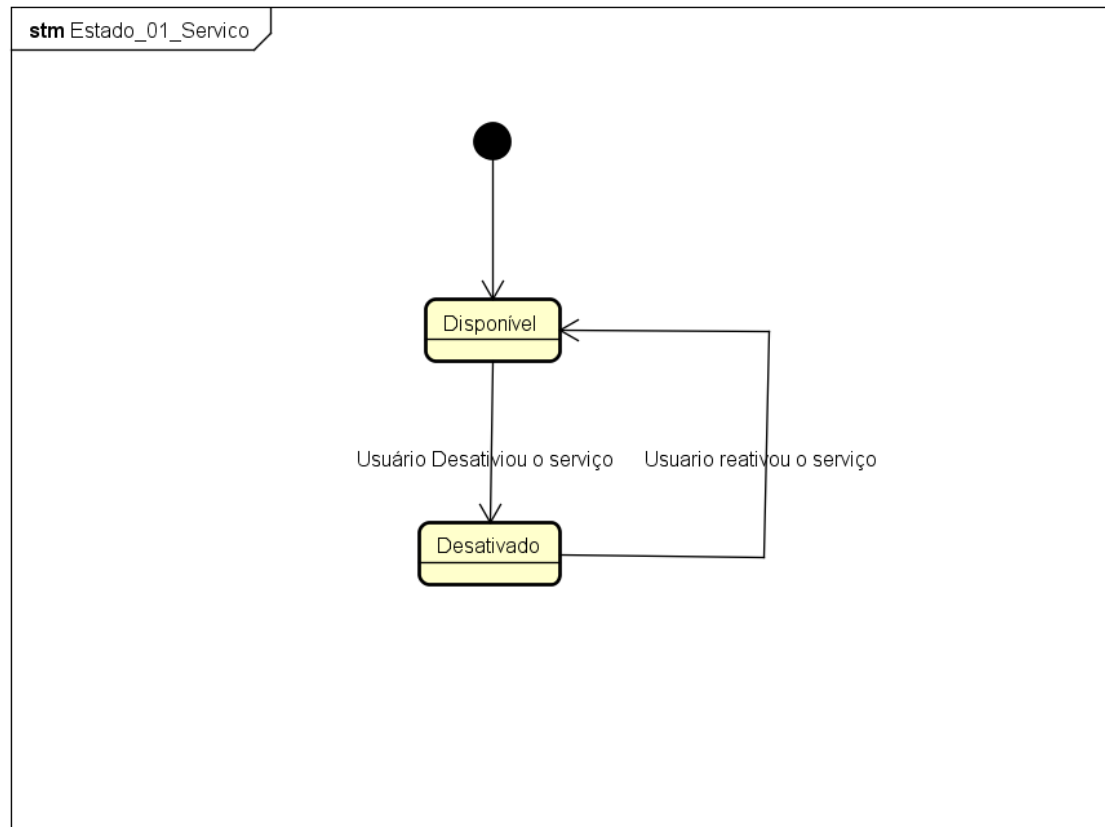
### 4.2.14. Personaliza - Formulário de Atendimento



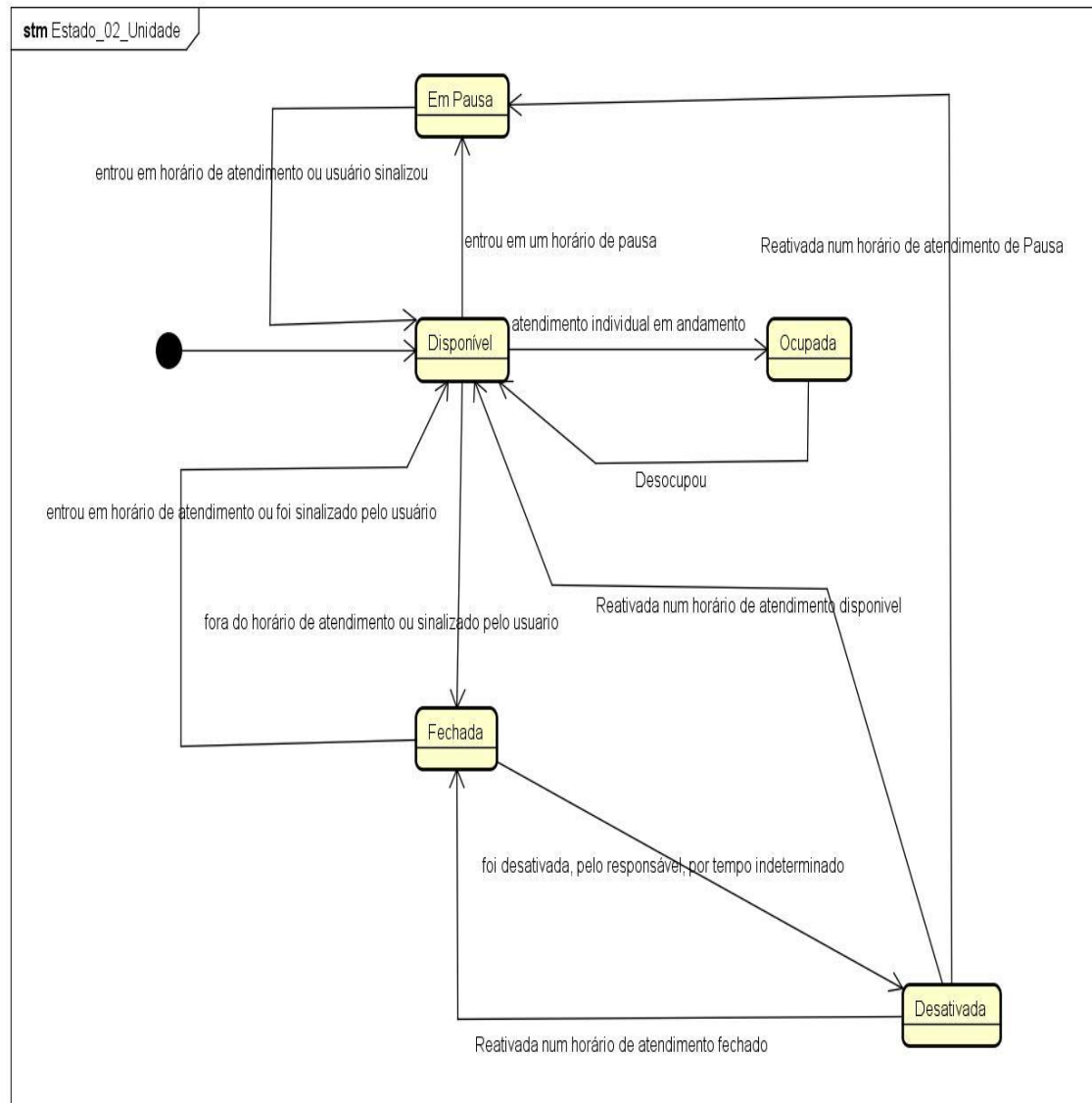
## 4.3. Diagrama de Máquina de Estados

Alguns Objetos possuem estados diferentes durante seu ciclo de vida. Para demonstrar as mudanças de estados para estes objetos específicos, foram definidos diagramas de estados.

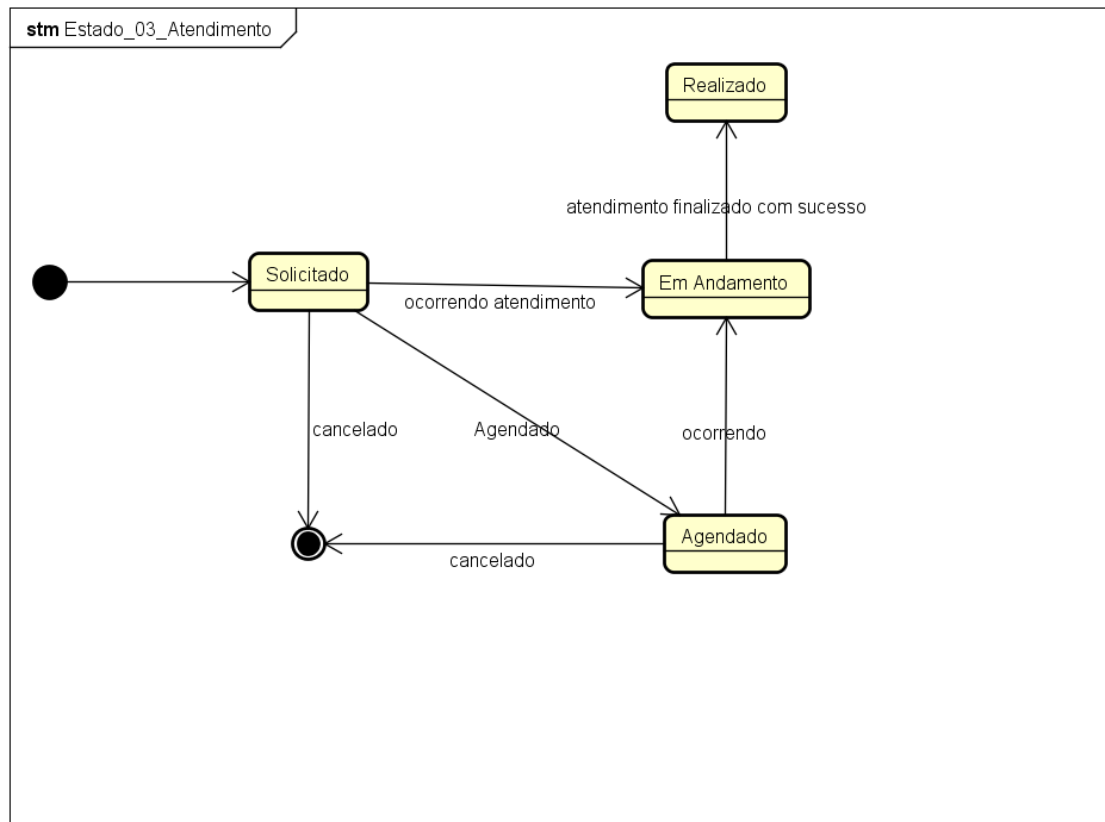
### 4.3.1. Serviço



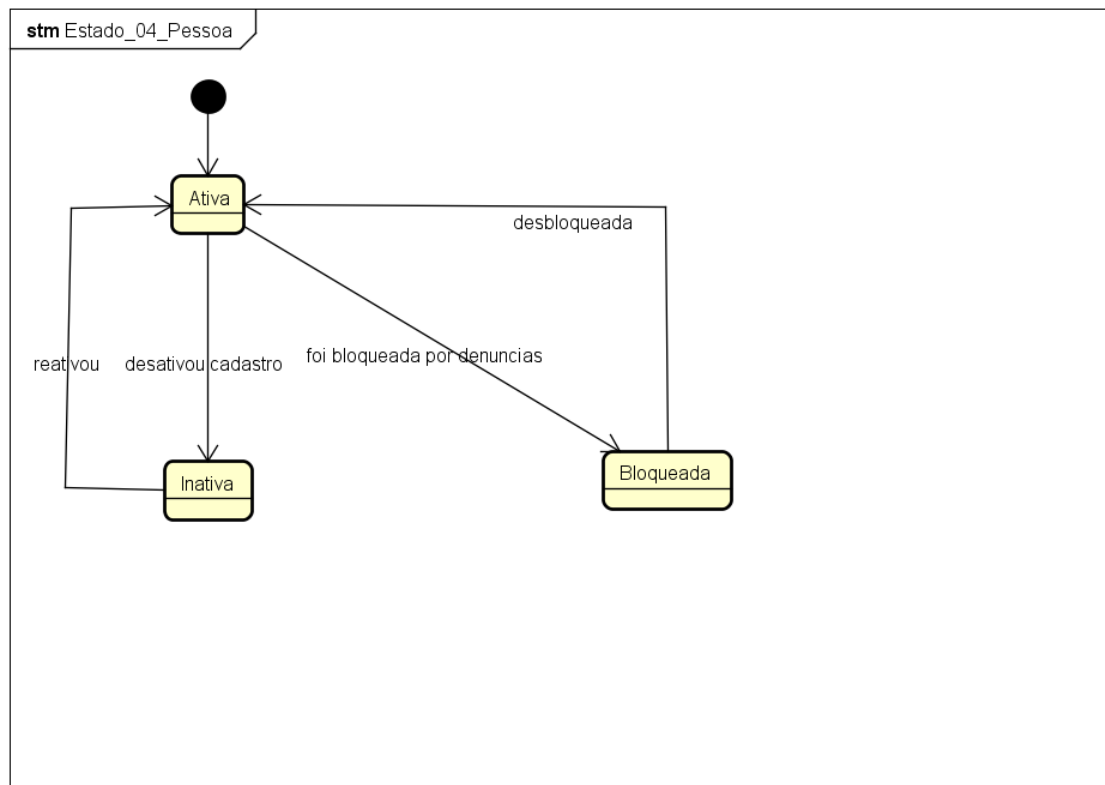
#### 4.3.2. Unidade



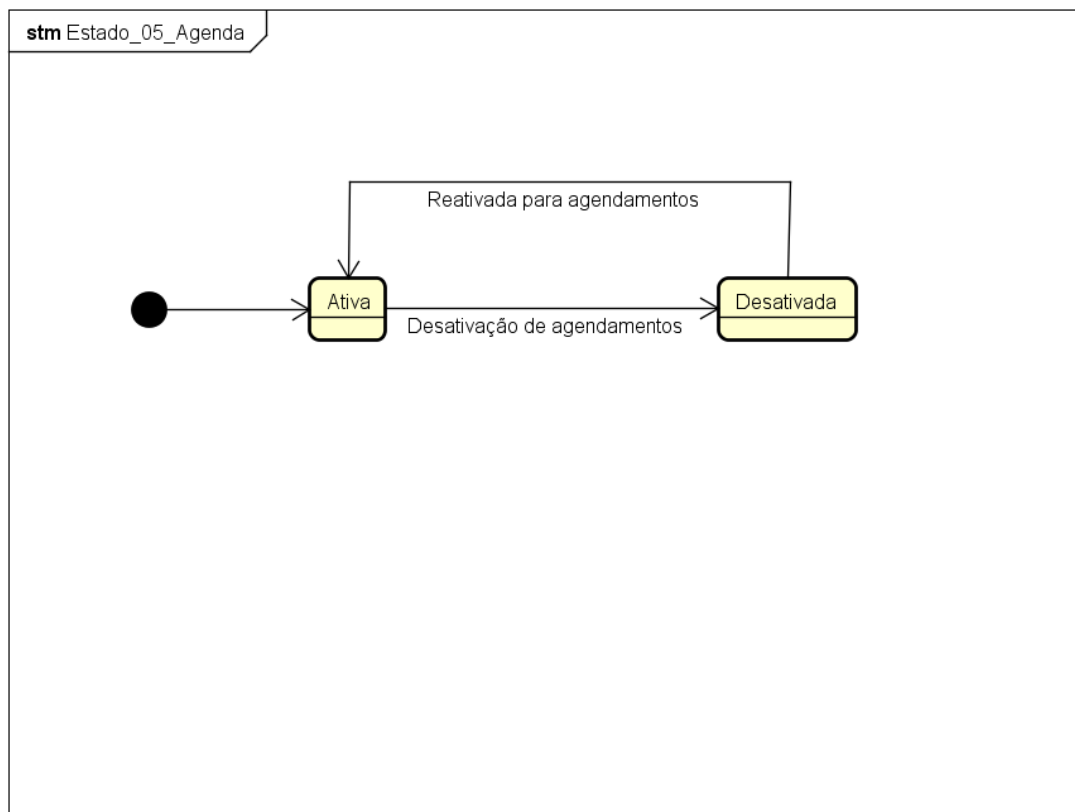
### 4.3.3. Atendimento



#### 4.3.4. Pessoa



### 4.3.5. Agenda



**APÊNDICE C – DOCUMENTO DE MODELO DE PROJETO****DOCUMENTO DE MODELO DE PROJETO**

Histórico de Modificações		
Data	Autor	Descrição
05/07/2017	Ewertom Moraes	Criação do documento.
28/07/2017	Ewertom Moraes	Definição do estilo e arquitetura, subsistemas e componentes.
15/08/2017	Ewertom Moraes	Definição do projeto de interfaces, pseudocódigo.
10/09/2017	Ewertom Moraes	Revisão e complemento do documento

**1. Projeto de Arquitetura**

A arquitetura a ser utilizada será em camadas, mais especificamente MVC. O Sistema será dividido em duas aplicações distintas: Servidor (API) e App. O servidor deverá estar disponível na web, já que a comunicação entre os dois sistemas será por este meio, permitindo que onde o usuário do app esteja, conectado com a internet, consiga utilizar o app. Para os dois subsistemas, servidor e app, componentes foram identificados, bem como suas interfaces de integração e colaboração com demais componentes da aplicação.

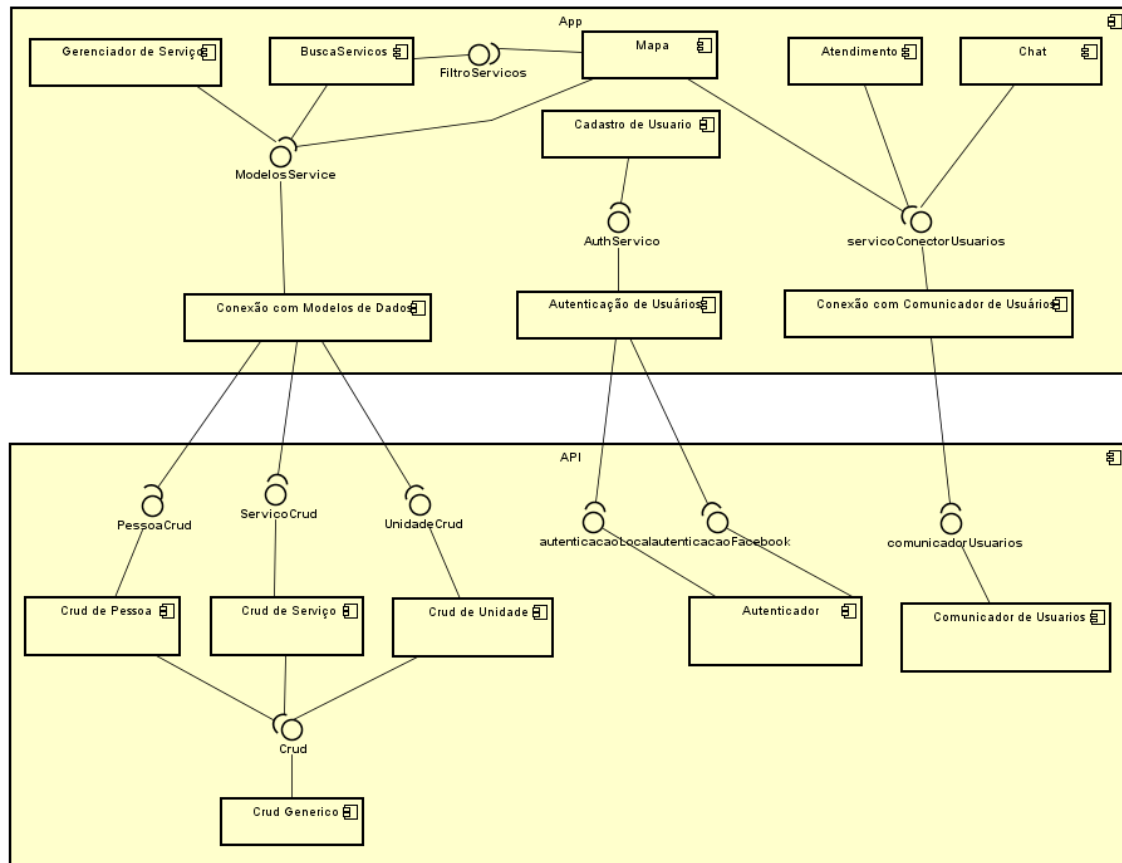


Figura 1. Diagrama de Componentes Geral – App e API.

### Instalação

A instalação poderá ocorrer da seguinte forma:

- Banco de dados e Servidor na mesma máquina OU em máquinas separadas, inclusive precisando se comunicar pela web.
- Aplicativo instalado em cada dispositivo dos usuários.

O aplicativo se comunicará através de web com a API.

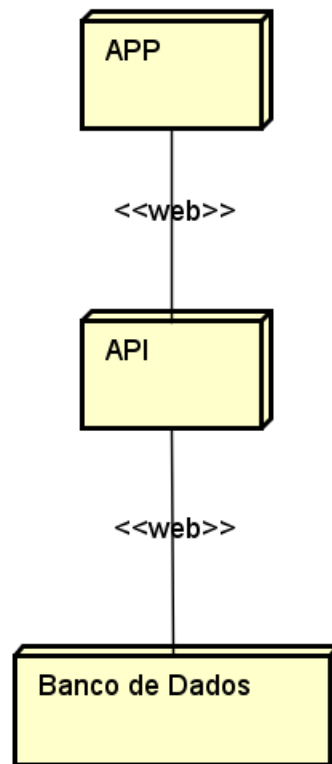


Figura 2. Diagrama de Implantação.

## 2. Classes de Projeto

Foram desenvolvidos diagramas de classe contendo novas classes de projeto para cada componente do sistema. Nestes diagramas, as classes de entidades, as especificadas na etapa análise de sistema, foram omitidas, para melhor clareza das novas classes, e melhor visualização dos diagramas. É necessário, portanto, em casos de dúvida, recorrer ao diagrama de classes de análise, para saber como estas classes de projeto e as classes de análise se relacionam. Entretanto, esta é uma análise trivial, já que na maioria dos casos, a relação é explícita. Exemplo: PessoaController instancia sua dependência Crud passando como tipo genérico T o tipo Pessoa. Além disso, a assinatura dos métodos neste diagrama deixa claro a ocorrência de referenciar uma classe de análise. Na seção 3 deste documento, quando documentado todos os métodos de cada classe de projeto, estas relações ficam mais claras e bem documentadas.

### 2.1 Classe de Projeto - API

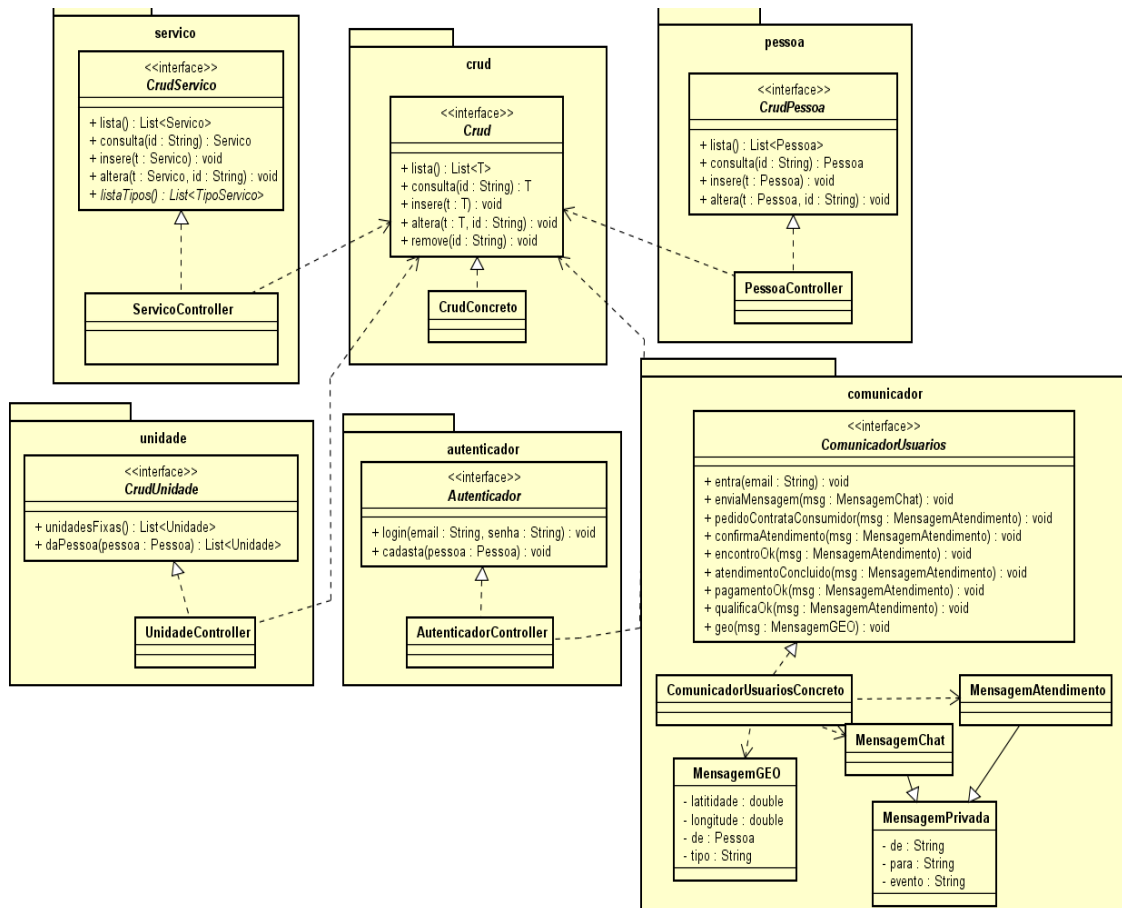


Figura 3. Diagrama de classes de projeto - API.

## 2.2 Classes de Projeto – App

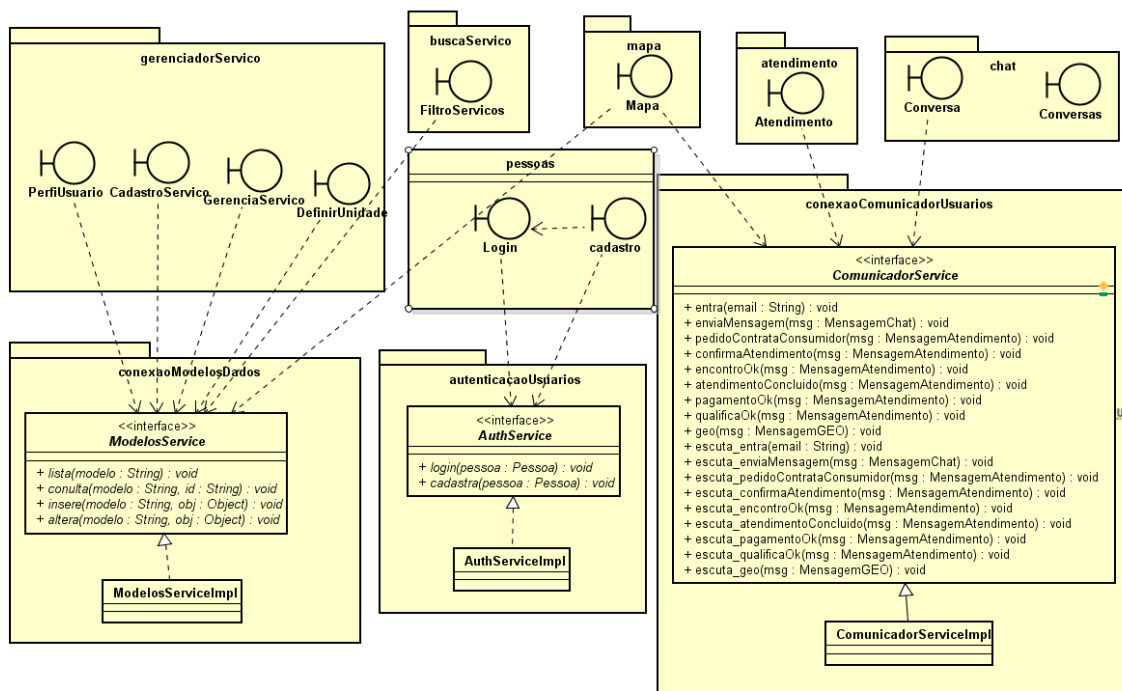


Figura 4. Diagrama de classes de projeto - App.

## 2. Projetos de Componentes

A seguir a descrição dos componentes da aplicação iservice que foram levantados analisando blocos de funcionalidades e necessidades encontradas até a etapa da análise de sistema.

### 2.1 API

#### 2.1.1. Crud Genérico

##### Função

Prover uma interface de comunicação entre um banco de dados qualquer e a camada de negócio da aplicação, de tal forma que gere reaproveitamento de Código, com disponibilização de métodos na interface pública para que qualquer tipo de entidade do Sistema possa ser armazenado e consumido do SGBD que será utilizado.

Classe Interface	Crud<T>
Método	Especificação
lista():lista<T>	Retorna uma lista de objetos do tipo T armazenados no repositório de dados
consulta(id):T	Retorna um registro do tipo T armazenados no repositório de dados
insere(t:T)	Insere um novo registro t do tipo T no repositório de dados

altera(t:T, id)	Altera um registro t do tipo T, que tenha o id especificado do repositório de dados.
remove(id)	Remove um registro do tipo T que tenha o id especificado do repositório de dados

## Rotinas do Componente

Classe: CrudConcreto<T>	
Método	Implementação
lista():lista<T>	pesquisa no banco todos os registros do tipo T; converte resultado do banco em uma lista de objetos T retorna a lista T;
consulta(id):T	pesquisa no banco registro com id == ao id informado; converte o resultado em um objeto do tipo T; retorna o objeto T;
insere(t:T)	registroParaBanco = converte T em registro compatível para o banco; insere no banco de dados o registroParaBanco;
altera(t:T, id)	registroParaBanco = converte T em registro compatível para o banco; altera do banco de dados o registroParaBanco substituindo pelo atual do banco com id == ao id informado;
remove(id)	atual = procura no banco de dados registro tipo T com id == id especificado se encontrou deleta atual

### 2.1.2 Crud de Pessoa

#### Função

Componente responsável pela persistência e consumo de dados de pessoas salvos no banco de dados.

Interface (WEB)	PessoaCrud
-----------------	------------

Método	URI	Corpo da Requisição	Especificação
lista():lista<Pessoa>	/pessoa		Retorna uma lista de pessoas cadastradas no sistema
consulta(id)	/pessoa/{id}		Retorna um registro de pessoa cadastrada no Sistema que tenha o id especificado na URI.
insere(pessoa:Pessoa)	/pessoa	pessoa:Pessoa	Insere uma nova pessoa especificada no corpo da mensagem
altera(pessoa:Pessoa, id)	/pessoa/{id}	pessoa:Pessoa	Altera a pessoa informada no corpo da requisição, e que tenha o id informado na URI

### Rotinas do Componente

Classe: Pessoa	
Método	Implementação
salvaSenha()	<code>this.senha =criptografa(this.senha);</code>

Classe: PessoaController	
Método	Implementação
lista():lista<Pessoa>	<code>retorna this.crud.lista();</code>
consulta(id)	<code>retorna this.crud.consulta(id);</code>
insere(pessoa:Pessoa)	<code>this.crud.insere(pessoa);</code>
altera (pessoa:Pessoa, id)	<code>this.crud.altera(pessoa, id);</code>
lista():lista<Pessoa>	<code>retorna this.crud.lista();</code>

### 2.1.3 Crud de Serviço

## Função

Componente responsável pela persistência e consumo de dados de serviços de pessoas salvos no banco de dados.

Interface WEB ServicoCrud			
Método	URI	Corpo da Requisição	Especificação
lista():lista<Pessoa>	/pessoa		Retorna uma lista de pessoas cadastradas no sistema
consulta(id)	/pessoa/{id}		Retorna um registro de pessoa cadastrada no Sistema que tenha o id especificado na URI.
insere(pessoa:Pessoa)	/pessoa	pessoa:Pessoa	Insere uma nova pessoa especificada no corpo da mensagem
altera(pessoa:Pessoa, id)	/pessoa/{id}	pessoa:Pessoa	Altera a pessoa informada no corpo da requisição, e que tenha o id informado na URI
servicoDaUnidade(unidade:Unidade)	/servico/ daUnidade/{id}		Procura e retorna o serviço da unidade especificada.

## Rotinas do componente

Classe: ServicoController	
Método	Implementação
lista():lista<Servico>	retorna this.crud.lista();
consulta(id)	retorna this.crud.consulta(id);
insere(servico: Servico)	this.crud.insere(pessoa);
altera (servico: Servico, id)	this.crud.altera(pessoa, id);
servicoDaUnidade(unidade: Unidade)	retorna this.crud.consulta(unidade.servico.id)

Classe: TipoServicoController	
----------------------------------	--

Método	Implementação
lista():lista<Servico>	retorna this.crud.lista();

## 2.1.4 Crud de Unidade

### Função

Componente responsável pela persistência e consumo de dados de serviços de pessoas salvos no banco de dados.

Interface WEB UnidadeCrud		
Método	Requisição	Especificação
unidadesFixas() :lista<Unidade>	GET /unidade/fixa	Retorna a lista de unidades fixas existentes.
daPessoa(pessoa:Pessoa) :lista<Unidade>	GET /unidade/pessoa/{id}	Retorna uma lista de unidades de todos os serviços da pessoa.

### Rotinas do componente

Classe: ServicoController	
Método	Implementação
unidadesFixas() :lista<Unidade>	var filtros = preparaFiltros(onde registro.localizacao == [0,0]) return this.crud.lista(filtros);
daPessoa(pessoa:Pessoa) :lista<Unidade>	var filtros = preparaFiltros(onde registro.servico.pessoa == pessoa) return this.crud.lista(filtros);

## 2.1.5 Autenticador

### Função

Provê interface web para cadastrar e autenticar usuários do Sistema.

Interface WEB Autenticador		
Método	Requisição	Especificação
login(email:string, senha:string)	POST /auth/login	Procura uma Pessoa com dados de autenticação <i>local</i> que batam com os fornecidos. A senha é criptografada antes

		para comparação. Havendo sucesso, responde <i>status</i> HTTP 201, senão 401.
cadastra(pessoa:Pessoa)	POST /auth/cadastra	Realiza o cadastro de uma pessoa(Usuário).

### Rotinas do componente

Classe: AutenticadorConcreto	
Método	Implementação
login(email:string, senha:string)	<pre>var filtros = preparaFiltros(onde pessoa.local.email == email E pessoa.local.senha == criptogra(senha) ) se(this.crudPessoa.consulta(filtros) != nulo){     responde(200); }senao{     responde(401); }</pre>
cadastra(pessoa:Pessoa)	<pre>se(dados da pessoa estão corretos){     this.crudPessoa.insere(pessoa);     responde(201) }senao{     responde(400) }</pre>

### 2.1.6 Comunicador de Usuários

#### Função

Realiza a comunicação entre usuários, disparando mensagens em tempo real para contemplar o funcionamento de chats de conversas entre usuários e troca de outras informações no esquema de comunicação ponto a ponto.

Interface WEB ComunicadorUsuarios		
Método	Requisição	Especificação
entra(email:string)	Protocolo WS evento “subscribe”	Escuta um evento de um cliente ao iniciar sua comunicação com o servidor. Cria uma caixa de mensagens para o cliente.

enviaMensagem( msg:MensagemChat)		Escuta um evento de um cliente para encaminhar mensagens de chat de um cliente ao outro. Encaminha a mensagem do tipo chat para a caixa de mensagens do destinatário.
pedidoContrataConsumidor( msg:MensagemAtendimento)	Protocolo WS Evento “PEDIDO CONSUMIDOR”	Escuta um evento de cliente e encaminha para a caixa de mensagens do destinatário especificado uma mensagem do tipo “PEDIDO CONSUMIDOR?”
ConfirmaAtendimento( msg:MensagemAtendimento)	Protocolo WS Evento “PRESTADOR ACEITOU”	Escuta um evento de cliente e encaminha para a caixa de mensagens do destinatário especificado uma mensagem do tipo “PRESTADOR ACEITOU”
encontroOk( msg:MensagemAtendimento)	Protocolo WS Evento “ENCONTRO OK”	Escuta um evento de cliente e encaminha para a caixa de mensagens do destinatário especificado uma mensagem do tipo “ENCONTRO OK”
atendimentoConcluido( msg:MensagemAtendimento)	Protocolo WS Evento “ATENDIMENTO CONCLUIDO”	Escuta um evento de cliente e encaminha para a caixa de mensagens do destinatário especificado uma mensagem do tipo “ATENDIMENTO CONCLUIDO”
pagamentoOk( msg:MensagemAtendimento)	Protocolo WS Evento “PAGAMENTO OK”	Escuta um evento de cliente e encaminha para a caixa de mensagens do destinatário especificado uma mensagem do tipo “PAGAMENTO OK”
qualificaOk( msg:MensagemAtendimento)	Protocolo WS Evento “QUALIFICA”	Escuta um evento de cliente e encaminha para a caixa de mensagens do destinatário especificado uma mensagem do tipo “QUALIFICA”
geo( msg:MensagemGEO)	Protocolo WS Evento “geolocation user”	Escuta um evento de cliente e encaminha para todos os outros clientes conectados ao servidor uma mensagem do tipo “geolocation user”

Classe: ComunicadorConcreto	
Método	Implementação
entra(email:string)	quando ouvir 'subscribe' faça criaCaixaMensagemDo(email);
enviaMensagem( msg:MensagemChat)	quando ouvir 'CHAT MESSAGE' faça emitePara(msg.para, 'CHAT MESSAGE', msg)
pedidoContrataConsumidor( msg:MensagemAtendimento)	quando ouvir 'PEDIDO CONSUMIDOR' faça emitePara(msg.para, 'PEDIDO CONSUMIDOR', msg)
ConfirmaAtendimento( msg:MensagemAtendimento)	quando ouvir 'PRESTADOR ACEITOU' faça emitePara(msg.para, 'PRESTADOR ACEITOU', msg)
encontroOk( msg:MensagemAtendimento)	quando ouvir 'ENCONTRO OK' faça emitePara(msg.para, 'ENCONTRO OK', msg)
atendimentoConcluido( msg:MensagemAtendimento)	quando ouvir 'ATENDIMENTO CONCLUIDO' faça emitePara(msg.para, 'ATENDIMENTO CONCLUIDO', msg)
pagamentoOk( msg:MensagemAtendimento)	quando ouvir 'PAGAMENTO OK' faça emitePara(msg.para, 'PAGAMENTO OK', msg)
qualificaOk( msg:MensagemAtendimento)	quando ouvir 'QUALIFICA' faça emitePara(msg.para, 'QUALIFICA', msg)
geo( msg:MensagemGEO)	quando ouvir 'geolocation user' faça emiteParaTodos('geolocation user', msg)

## 2.2 APP

### 2.2.1 Conexão com Modelos de Dados

## Função

Prover uma interface de comunicação entre as classes de fronteiro com o usuário e o a API servidora. Esta interface abstrai para as classes de telas a complexidade de requisição HTTP.

Classe Interface	ModelosService
Método	Especificação
lista(modelo : String)	Retorna uma lista de objetos cadastrados do modelo de dados especificado.
consulta(modelo: String, id : String)	Retorna um objeto do modelo de dados especificado com o id igual ao id especificado
insere(modelo: String, obj : Object)	Inserir o objeto obj que deve ser do tipo do modelo especificado.
altera(modelo : String, obj : Object)	Altera um objeto obj que deve ser do tipo do modelo especificado, onde o objeto original no repositório de dados será procurado com o atributo id do obj

## Rotinas do Componente

Classe: ModelosServiceImpl	
Método	Implementação
lista(modelo : String)	<pre> resposta = realiza requisição assincrona http .para(nomeModelo) .metodo(GET) se(resposta.status == 200)     retorna resposta.dados; senao     lancaErro(); </pre>
consulta(modelo: String, id : String)	<pre> resposta = realiza requisição assincrona http .para(nomeModelo+'/'+id) .metodo(GET) se(resposta.status == 200)     retorna resposta.dados; senao     lancaErro(); </pre>
insere(modelo: String, obj : Object)	<pre> resposta = realiza requisição assincrona http .para(nomeModelo) .metodo(POST) .corpo(obj) se(resposta.status == 201) </pre>

	<pre>         retorna resposta.dados;         senao             lancaErro();     </pre>
altera(modelo : String, obj : Object)	<pre>         resposta = realiza requisição assincrona         http             .para(nomeModelo+'/'+obj.id)             .metodo(PUT)             .corpo(obj)         se(resposta.status == 200)             retorna resposta.dados;         senao             lancaErro();     </pre>

## 2.2.2 Autenticação de usuários

### Função

Prover uma interface de comunicação entre as classes de fronteira com o usuário e o a API servidora para prover funcionalidades autenticação e cadastro. Esta interface abstrai para as classes de telas a complexidade de requisição HTTP.

Classe Interface	AuthService
Método	Especificação
login(pessoa : Pessoa)	Verifica se a pessoa instanciada para login está com dados corretos devidamente cadastrados de sua autenticação – e-mail e senha.
cadastra(pessoa: Pessoa)	Realiza o cadastro de uma pessoa

### Rotinas do Componente

Classe: AuthServiceImpl	
Método	Implementação
login(pessoa : Pessoa)	<pre>         resposta = realiza requisição assincrona         http             .para(auth/login)             .metodo(POST)             .corpo(email e senha);          caso(resposta.status == 200)             retorna resposta.dados;         caso(resposta.status == 401)     </pre>

	<pre>         retorna sinaliza de credenciais         incorretas         qualquer outro caso         retorna lancaErro(); </pre>
<pre> cadastra(pessoa: Pessoa) </pre>	<pre> resposta = realiza requisição assíncrona http .para(auth/cadastra) .metodo(POST) .corpo(pessoa);  se(resposta.status == 200)     retorna resposta.dados; senao     retorna lancaErro(); </pre>

### 2.2.3 Conexão com Comunicador de Usuários

#### Função

Realiza a comunicação entre usuários da aplicação. Para isso, emite eventos para o servidor, este que interliga os usuários por meio de uma comunicação bidirecional. Também é responsável pela configuração de escuta destes eventos quando for o caso do usuário receber estas mensagens. Ou seja, aqui temos a configuração de envio e escuta, quando o usuário precisar enviar mensagens à outros usuários e quando outros enviam mensagens a ele.

Método	Especificação
<pre> escuta_entra():Promessa&lt;string&gt; </pre>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando outro usuário começa a usar a aplicação.
<pre> escuta_enviaMensagem():Promessa&lt; MensagemChat&gt; </pre>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando o usuário recebe uma mensagem chat de outro usuário.
<pre> escuta_pedidoContrataConsumidor():Promessa&lt; MensagemAtendimento &gt; </pre>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando o usuário é prestador de serviço e alguém pediu para ser atendido.
<pre> escuta_ConfirmaAtendimento():Promessa&lt; MensagemAtendimento&gt; </pre>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando o usuário é consumidor, pediu

	atendimento de um serviço, e obteve a resposta de aceite do prestador para atender.
escuta_encontroOk():Promessa< MensagemAtendimento>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando o usuário é consumidor, pediu atendimento, e o prestador confirmou o encontro.
escuta_atendimentoConcluido():Promessa< MensagemAtendimento>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando o usuário é prestador, e recebeu a confirmação do usuário consumidor sobre a conclusão do atendimento.
escuta_pagamentoOk():Promessa< MensagemAtendimento>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando o usuário é consumidor e o prestador de serviço no termino do atendimento confirmou o pagamento.
escuta_qualificaOk():Promessa< MensagemAtendimento>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando o usuário é prestador de serviço e consumidor, no término do atendimento, qualificou o atendimento.
escuta_geo():Promessa<MensagemGEO>	Escuta um evento de envio de mensagem vinda do servidor, que é de outro usuário. Ocorre quando qualquer usuário que é prestador de serviço, possui unidades móveis publicadas, está com a aplicação funcionando em seu dispositivo e se locomoveu. Todos os outros usuários recebem esta notificação.

Classe: ComunicadorConcreto	
Método	Implementação
escuta_entra():Promessa<string>	promessa = new Promessa(); quando ouvir 'subscribe' faça

	<pre> promessa.conclui(msg); retorna promessa; </pre>
escuta_enviaMensagem():Promessa< MensagemChat>	<pre> promessa = new Promessa(); quando ouvir 'CHAT MESSAGE' faça     promessa.conclui(msg); retorna promessa; </pre>
escuta_pedidoContrataConsumidor():Promessa< MensagemAtendimento >	<pre> promessa = new Promessa(); quando ouvir 'PEDIDO CONSUMIDOR' faça     promessa.conclui(msg); retorna promessa; </pre>
escuta_ConfirmaAtendimento():Promessa< MensagemAtendimento>	<pre> promessa = new Promessa(); quando ouvir 'PRESTADOR ACEITOU' faça     promessa.conclui(msg); retorna promessa; </pre>
escuta_encontroOk():Promessa< MensagemAtendimento>	<pre> promessa = new Promessa(); quando ouvir 'ENCONTRO OK' faça     promessa.conclui(msg); retorna promessa; </pre>
escuta_atendimentoConcluido():Promessa< MensagemAtendimento>	<pre> promessa = new Promessa(); quando ouvir 'ATENDIMENTO CONCLUIDO' faça     promessa.conclui(msg); retorna promessa; </pre>
escuta_pagamentoOk():Promessa< MensagemAtendimento>	<pre> promessa = new Promessa(); quando ouvir 'PAGAMENTO OK' faça     promessa.conclui(msg); retorna promessa; </pre>
escuta_qualificaOk():Promessa< MensagemAtendimento>	<pre> promessa = new Promessa(); quando ouvir 'QUALIFICA' faça     promessa.conclui(msg); retorna promessa; </pre>
escuta_geo():Promessa< MensagemGEO>	<pre> promessa = new Promessa(); quando ouvir 'geolocation user' faça     promessa.conclui(msg); retorna promessa; </pre>
entra():Promessa<string>	<pre> quando ouvir 'subscribe' faça     criaCaixaMensagemDo(email); </pre>

<code>enviaMensagem():Promessa&lt; MensagemChat&gt;</code>	quando ouvir 'CHAT MESSAGE' faça <code>emitePara(msg.para, 'CHAT MESSAGE', msg)</code>
<code>pedidoContrataConsumidor():Promessa&lt; MensagemAtendimento &gt;</code>	quando ouvir 'PEDIDO CONSUMIDOR' faça <code>emitePara(msg.para, 'PEDIDO CONSUMIDOR', msg)</code>
<code>ConfirmaAtendimento():Promessa&lt; MensagemAtendimento&gt;</code>	quando ouvir 'PRESTADOR ACEITOU' faça <code>emitePara(msg.para, 'PRESTADOR ACEITOU', msg)</code>
<code>encontroOk():Promessa&lt; MensagemAtendimento&gt;</code>	quando ouvir 'ENCONTRO OK' faça <code>emitePara(msg.para, 'ENCONTRO OK', msg)</code>
<code>atendimentoConcluido():Promessa&lt; MensagemAtendimento&gt;</code>	quando ouvir 'ATENDIMENTO CONCLUIDO' faça <code>emitePara(msg.para, 'ATENDIMENTO CONCLUIDO', msg)</code>
<code>pagamentoOk():Promessa&lt; MensagemAtendimento&gt;</code>	quando ouvir 'PAGAMENTO OK' faça <code>emitePara(msg.para, 'PAGAMENTO OK', msg)</code>
<code>qualificaOk():Promessa&lt; MensagemAtendimento&gt;</code>	quando ouvir 'QUALIFICA' faça <code>emitePara(msg.para, 'QUALIFICA', msg)</code>
<code>geo():Promessa&lt;Mensagem GEO&gt;</code>	quando ouvir 'geolocation user' faça <code>emiteParaTodos('geolocation user', msg)</code>

## 2.2.4 Gerenciador de Serviço

### Função

Responsável por prover a interação com usuário, pegando informações das interfaces do usuário ou disponibilizando para exibição. As telas *Cadastro de Serviço*, *Gerenciar Serviço* e *Definir Unidade* estão contidas neste componente.

### Rotinas do Componente

Classe: PerfilUsuario	
Método	Implementação
servicosUsuario(servico:Servico):Lista<Servico>	retorna pegaSessaoUsuario().pessoa.servicos;

Classe: CadastroServico	
Método	Implementação
cadasta(servico: Servico)	this.modelosService.insere('servico', servico);

Classe: GerenciaServico	
Método	Implementação
altera(servico:Servico)	this.modelosService.altera('servico', servico);

Classe: DefinirUnidade	
Método	Implementação
definirUnidade(servico:Servico, unidade:Unidade)	se(servico.unidades.contem: unidade) servico.unidades.substituir(unidade) senao servico.unidades.adiciona(unidade)

## 2.2.5 Busca de Serviços

### Função

Faz a busca por serviços mediante aplicação de filtros realizados por uma interface de usuário.

### Rotinas do Componente

Classe: FiltroServicos	
Método	Implementação

<code>filtraBusca():Lista&lt;Servico&gt;</code>	<code>filtros = coletaDadosFiltroUsuario(); this.modelosService.lista('servico', filtros)</code>
---	--

## 2.2.6 Mapa

### Função

Dispõe o mapa de serviços para o usuário. Atualiza constantemente os dados de localização das unidades de atendimentos móveis de serviços no mapa.

### Rotinas do Componente

Classe: Mapa	
Método	Implementação
<code>consultaUnidadesFixas()</code>	<code>this.modelosService.lista(unidades, onde unidade.tipo == 'FIXA')</code>
<code>configuraEscutaLocalizacoes()</code>	<code>promessa = this.comunicadorService .escuta_geolocalizacao(); promessa.quandoResponder faça:   marcadorNovo = nulo;   procuraNoMapaUsuario da resposta;   se achou marcadorMapa     marcadorNovo = marcadorMapa;   senao     marcadorNovo = instancia novo marcador;   marcadorNovo.latitude = resposta.latitude;   marcadorNovo.longitude = resposta.longitude;   atualizaMapa();</code>

## 2.2.7 Atendimento

### Função

Componente responsável por interagir com o usuário durante um atendimento, seja consumidor ou prestador. Faz uso do *comunicador service* para troca de informações em tempo real durante o atendimento.

### Rotinas do Componente

Classe: Atendimento	
---------------------	--

Método	Implementação
pedidoContrataConsumidor()	mensagem.de = pega email do usuario logado; mensagem.para = pega email do outro usuario; this.comunicador.pedidoContrataConsumidor(mensagem);
ConfirmaAtendimento()	mensagem.de = pega email do usuario logado; mensagem.para = pega email do outro usuario; this.comunicador.ConfirmaAtendimento (mensagem);
encontroOk()	mensagem.de = pega email do usuario logado; mensagem.para = pega email do outro usuario; this.comunicador.encontroOk (mensagem);
atendimentoConcluido()	mensagem.de = pega email do usuario logado; mensagem.para = pega email do outro usuario; this.comunicador.atendimentoConcluido(mensagem);
pagamentoOk()	mensagem.de = pega email do usuario logado; mensagem.para = pega email do outro usuario; this.comunicador.pagamentoOk(mensagem);
qualificaOk()	mensagem.de = pega email do usuario logado; mensagem.para = pega email do outro usuario; this.comunicador.qualificaOk(mensagem); Emite mensagem de sucesso ao usuário; redireciona usuário para tela <i>Perfil Usuario</i> ;

## 2.2.8 Chat

### Função

Componente para compor o funcionamento e interação do chat de mensagens privadas ao usuário – mensagens de texto trocadas e visualizadas entre pares de usuários.

### Rotinas do Componente

Classe: Conversas	
Método	Implementação
listaConversas()	conversas = pega lista de chats gravados no dispositivo; retorna conversas;

Classe: Conversa	
Método	Implementação
enviaMensagem()	<pre> mensagem.de = pega email do usuario logado; mensagem.para = pega email do outro usuário da conversa; this.comunicador.enviaMensagem(mensagem); exibe mensagem do próprio usuário que mandou na tela, como ultima mensagem </pre>
configuraEscutaMensagensDaConversa()	<pre> promessa = this.comunicadorService .ecuta_enviaMensagem(); promessa.quandoResponder faça:     this.mensagensConversa.adiciona(resposta) </pre>

### 3. Modelos de Interfaces

Para a aplicação IService foram desenhados protótipos das interfaces do usuário. Aqui as interfaces também são chamadas de *tela*. Alguns protótipos de tela exibem tudo que nela será disponibilizado. Na maioria, o protótipo pode ser considerado como uma projeção completa do que deve conter na tela. Para caso em que falte algo no desenho, esclarecimentos são acrescentados.

#### 3.1 Tela Login

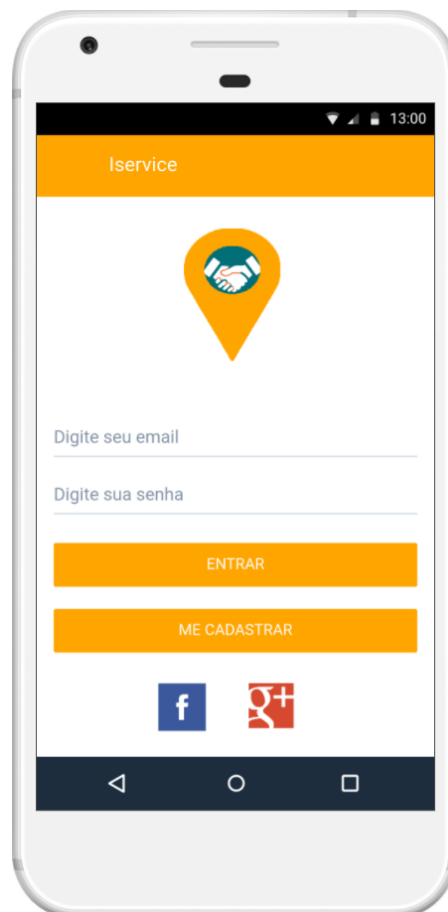


Figura 5. Protótipo da interface de usuário Login

Tela Login	
Descrição	Tela inicial do app. Serve para autenticar quaisquer usuários que irá utilizar a aplicação. Conta com a opção de cadastro de usuários. Possui as opções de login com redes sociais. Após ser autenticado ou cadastrado com sucesso, o usuário é redirecionado para a tela de perfil de usuário.
Acionamentos	Ao abrir o app.
Links	<ul style="list-style-type: none"> <li>• Cadastro de Pessoa</li> </ul>
Observações	

### 3.2 Tela Cadastro de pessoa (usuário)



Figura 6. Protótipo da interface de usuário Cadastro de pessoa

Tela Cadastro de pessoa (Usuário)	
Descrição	É onde os novos usuários irão se cadastrar na plataforma por meio de cadastro padrão. Contempla cadastro de pessoas físicas e jurídicas, onde os campos que são exclusivos de cada tipo aparecem mediante o tipo de pessoa selecionado. Após ser cadastrado com sucesso, o usuário é redirecionado para a tela de perfil de usuário.
Acionamentos	<ul style="list-style-type: none"> <li>• Login</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Não há links</li> </ul>
Observações	Campos adicionais que não aparecem no protótipo são: telefone, interesses (lista de termos textuais que o usuário tem interesse).

### 3.3 Tela Perfil de pessoa (usuário)



Figura 7. Protótipo da interface de Perfil de pessoa

Tela Perfil de pessoa	
Descrição	Tela que contém o resumo do cadastro do usuário, além de uma lista de serviços que ele publicou na plataforma. É Considerada a “tela inicial” do app.
Acionamentos	<ul style="list-style-type: none"> <li>• Tela Login (Redirecionamento)</li> <li>• Botão “Perfil” no menu <i>tabs</i> inferior</li> <li>• Tela Atendimento (Redirecionamento)</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Mapa – botão “mapa” no menu <i>tabs</i> inferior.</li> <li>• Tela Conversas– botão “chat” no menu <i>tabs</i> inferior.</li> <li>• Tela cadastrar serviço - Botão adicionar serviço.</li> <li>• Tela gerenciar serviço – Botão gerenciar serviço</li> </ul>
Observações	

### 3.4 Tela Mapa de serviços

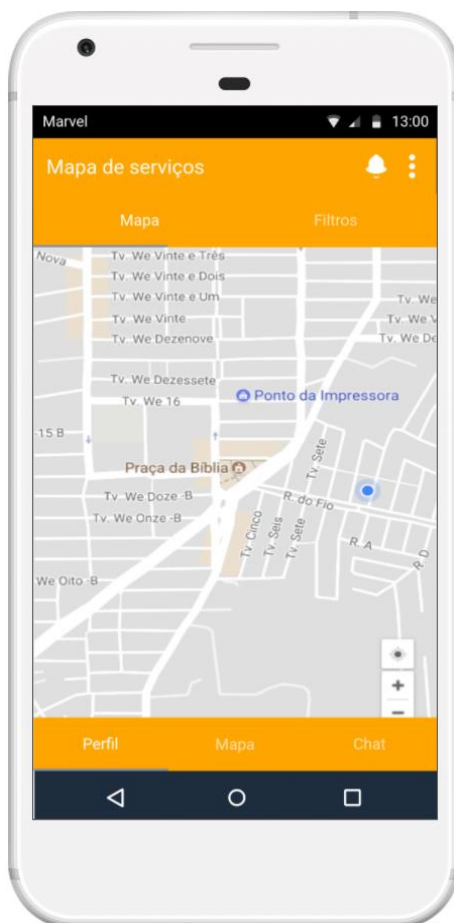


Figura 8. Protótipo da interface de usuário Mapa de serviços

Tela Mapa de serviços	
Descrição	Contem principalmente um mapa no qual são exibidas as unidades de atendimento fixas e móveis. Ou seja, os prestadores de serviço representados por suas unidades de seus serviços cadastrados.
Acionamentos	<ul style="list-style-type: none"> <li>• Botão “Mapa” no menu <i>tabs</i> inferior</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Perfil – botão “mapa” no menu <i>tabs</i> inferior.</li> <li>• Tela Conversas – botão “chat” no menu <i>tabs</i> inferior.</li> <li>• Tela Perfil de serviço – ao tocar em um marcador de unidade no mapa.</li> <li>• Tela Filtros de serviço – Botão Filtros no menu <i>tabs</i> superior</li> </ul>
Observações	

### Tela Filtros de serviços



Figura 9. Protótipo da interface de usuário Filtros de serviços

Tela Filtros de serviços	
Descrição	Contem filtros para serem utilizados em uma busca refinada por serviços. Atualmente é só acionada na tela mapas, mas futuramente poderá ser utilizada em outros mecanismos de buscas.
Acionamentos	<ul style="list-style-type: none"> <li>• Botão “Filtros” na tela Mapa de serviços</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Perfil – botão “perfil” no menu <i>tabs</i> inferior.</li> <li>• Tela Conversas– botão “chat” no menu <i>tabs</i> inferior.</li> <li>• Tela Mapa– botão “mapa” no menu <i>tabs</i> inferior.</li> <li>• Tela Mapa – Botão Aplicar Filtros.</li> </ul>
Observações	Outros campos de filtros podem ser adicionados nesta tela. Neste caso, a ideia de interação se mantem, com campos um abaixo do outro.

### 3.5 Tela Perfil do serviço



Figura 10. Protótipo da interface de usuário Perfil do serviço

Tela Perfil do serviço	
Descrição	Exibe as principais informações sobre um serviço, como o tipo, descrição, preço e reputação atual. Objetivo principal desta tela é acionar a contratação do serviço, além de poder entrar em contato com o prestador do serviço.
Acionamentos	<ul style="list-style-type: none"> <li>• Ao tocar em um marcador de unidade de serviço no mapa, na tela Mapa de serviços</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Perfil – botão “perfil” no menu <i>tabs</i> inferior.</li> <li>• Tela Conversas– botão “chat” no menu <i>tabs</i> inferior.</li> <li>• Tela Mapa– botão “mapa” no menu <i>tabs</i> inferior.</li> <li>• Tela Conversa - botão “entrar em contato”.</li> <li>• Tela Atendimento – botão “contratar”.</li> </ul>
Observações	Quando o serviço é exclusivo para agendamentos, no lugar do botão “Contratar” fica o botão “Agendar”. Neste caso, a navegação para Tela Atendimento deixa de existir, e passa a ter acionamento para uma caixa de diálogo para informar dados do agendamento. Após

	agendamento bem-sucedido, o usuário é redirecionado para sua tela de Perfil de usuário.
--	---

### 3.6 Tela Atendimento



Figura 11. Protótipo da interface de usuário Atendimento

Tela Atendimento	
Descrição	Tela onde o processo de atendimento ocorre e é acompanhado passo a passo. A métrica que as etapas do atendimento vão ocorrendo, os ícones ao lado esquerdo das etapas passam da cor acinzentado para a cor secundária ativa do tema (verde), além do horário em formato <i>HH:mm</i> aparecer ao lado. Os passos e ícones, portanto, sempre serão exibidos, por mais que, por exemplo, a última etapa concluída ainda seja a de <i>encontro confirmado</i> .
Acionamentos	<ul style="list-style-type: none"> <li>• Botão “Contratar” na tela perfil de serviço.</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Menu Detalhes no menu superior <i>tabs</i>. Leva para uma tela que combina: perfil do prestador (dados básicos), perfil do serviço (sem botões agendar,</li> </ul>

	contratar, entrar em contato) e perfil do usuário consumidor(dados básicos).
Observações	O botão na parte inferior “Confirmação da etapa xxx” reflete um estado dinâmico por etapa, onde o usuário aciona o botão, quando lhe solicitado, para confirmar a conclusão de uma etapa. A medida que as etapas vão sendo confirmadas, o botão vai mudando entre ativo/inativo e seu rótulo.

### 3.7 Tela Cadastro de serviço



Figura 12. Protótipo da interface de usuário Cadastro de serviço

Tela Cadastro de serviço	
Descrição	Tela para cadastrar um novo serviço com os principais dados de um serviço. Outras informações ficam para a gerência de serviços. O usuário seleciona, por meio de mecanismo de auto completar, um tipo de serviço.

	Depois informar a faixa de preço para o seu serviço, além de uma descrição de texto livre. Por fim aciona a tela de definir unidade para definir uma inicial para o serviço. Voltando da tela de unidade, poderá acionar botão de efetivação do cadastro “Publicar Serviço”.
Acionamentos	<ul style="list-style-type: none"> <li>• Botão “Adicionar serviço” (sinal +) na tela perfil de usuário.</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Definir Unidade – ao tocar no quadro “definir unidade” ou, se já foi definida uma, ainda poderá ser acionado com o uso do botão de editar (lápiz) ao lado da unidade no mesmo quadro.</li> </ul>
Observações	A tela ainda possui o botão para efetivar o cadastro, (“Publicar serviço”), no final, após a definição de unidade.

### 3.8 Tela Definir unidade de atendimento



Figura 13. Protótipo da interface de usuário Definir unidade de atendimento

Tela Unidade de atendimento

Descrição	Tela para definir uma unidade de um serviço. O usuário informa o tipo de unidade por meio de <i>radio button group</i> . Somente quando é selecionado <i>fixa</i> , o campo “localização fixa” é exibido (Localização fixa aciona diálogo para informar com auto completar parte do endereço fixo a ser configurado para a unidade fixa). Em Horários de atendimento, opção dias da semana, o usuário aciona o botão ao lado para exibir um diálogo com a lista de dias da semana. Neste diálogo, o usuário informa um ou vários dias e confirma, voltando para a tela de definir unidade para concluir o cadastro. A pergunta <i>quem vai atender</i> , quando escolhido a opção <i>outra pessoa</i> é exibido um campo/componente parecido com <i>localização fixa</i> , onde, também com auto completar, pesquisa por um outro usuário cadastrado para ser o responsável pelo atendimento da unidade.
Acionamentos	<ul style="list-style-type: none"> <li>• Quadro Unidade de atendimento na tela Cadastro de serviço.</li> <li>• Botão editar unidade (lápiz) ao lado de cada unidade de atendimento na tela gerenciar serviço.</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Definir Unidade – ao tocar no quadro “definir unidade” ou, se já foi definida uma, ainda poderá ser acionado com o uso do botão de editar (lápiz) ao lado da unidade no mesmo quadro.</li> </ul>
Observações	

### 3.9 Tela Gerenciar serviço



Figura 14. Protótipo da interface de usuário Gerenciar serviço

Tela Gerenciar serviço	
Descrição	<p>Tela para gerenciar um serviço já cadastrado pelo usuário. Serve para modificar algumas informações do serviço e para complementar o cadastro de serviço. Conta com opções:</p> <ul style="list-style-type: none"> <li>• Acionar que está atendendo ou não (aberto ou fechado).</li> <li>• Consulta e gerenciamento do caixa do serviço.</li> <li>• Agenda (Para tipos de serviços que podem ser agendados, esta opção aparece para acionar uma tela de gerenciamento da agenda).</li> <li>• Manutenção nos campos de preço e descrição.</li> <li>• Adicionar, editar, desativar unidades de atendimento.</li> <li>• Modo Atendimento. (Para tipos de serviços que podem atender pelo app, esta função pode ser ligada/desligada para o serviço)</li> </ul>

	<ul style="list-style-type: none"> <li>• Formulário de atendimento – Leva à definição de formulário de atendimento – tela definir formulário.</li> <li>• Formulário de cadastro – Leva à definição de formulário de cadastro – tela definir formulário.</li> <li>• Tema visual – abre um diálogo simplório para escolher um esquema de cores para colorir tela do perfil do serviço quando for acionado por consumidores.</li> </ul>
Acionamentos	<ul style="list-style-type: none"> <li>• Botão gerenciar serviço (lápiz) ao lado de cada serviço na tela perfil de usuário.</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela Definir Unidade – ao tocar no quadro “definir unidade” nos botões de adicionar (sinal +) ou editar uma unidade existente (lápiz).</li> <li>• Tela Gerenciar Caixa – ao acionar a opção de gerenciar caixa (lápiz).</li> <li>• Definir Formulário – ao acionar os textos link “Formulário de atendimento” ou “Formulário de cadastro”.</li> </ul>
Observações	

### 3.10 Tela Conversas



Figura 15. Protótipo da interface de usuário Conversas

Tela Conversas	
Descrição	Lista as conversas já iniciadas entre usuário e outros usuários.
Acionamentos	<ul style="list-style-type: none"> <li>• Botão <i>chat</i> no menu <i>tabs</i> inferior.</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela <i>Conversa</i> ao tocar no botão de conversa ao lado de cada conversa (ícone triângulo) .</li> <li>• Tela <i>Perfil</i> – botão “perfil” no <i>menu tabs inferior</i>.</li> <li>• Tela <i>Mapa de serviços</i>– botão “mapa” no <i>menu tabs inferior</i>.</li> </ul>
Observações	

### 3.11 Tela Conversa

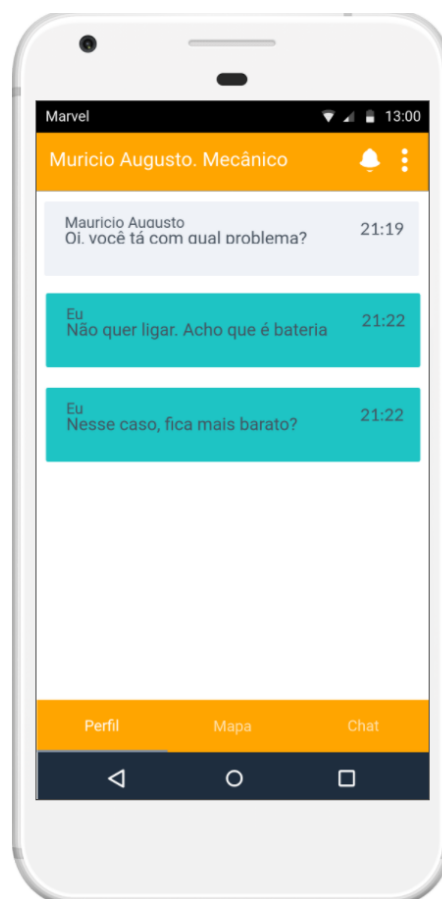


Figura 16. Protótipo da interface de usuário Conversa

Tela Conversa

Descrição	Exibe lista de mensagens de texto (conversa) trocadas entre o usuário e outro usuário da aplicação.
Acionamentos	<ul style="list-style-type: none"> <li>• Botão de conversa ao lado de cada conversa listada, na tela <i>Conversas</i>.</li> <li>• No <i>menu notificações</i>, quando a notificação é sobre uma mensagem de conversa recebida.</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Tela <i>Perfil</i> – botão “perfil” no <i>menu tabs inferior</i>.</li> <li>• Tela <i>Mapa de serviços</i>– botão “mapa” no <i>menu tabs inferior</i>.</li> <li>• Tela <i>Conversas</i> – botão “chat” no <i>menu tabs inferior</i>.</li> </ul>
Observações	

## APÊNDICE D – PESQUISA DE CAMPO PARA APERFEIÇOAMENTO DE REQUISITOS COM RESULTADOS

### Pesquisa sobre Prestação de Serviços por meio de aplicativos

Este formulário é uma Pesquisa para um Trabalho de Conclusão de Curso (TCC), na área da Computação. O objetivo é saber, de pessoas de todos os gêneros e classes, quanto a prestar e consumir serviço, quais são as dificuldades existentes, possíveis melhorias e novas ideias para auxiliar prestadores de serviço e consumidores finais.

E, com o debate sobre o assunto, o questionário trabalha a ideia de um app para divulgar e contratar qualquer tipo de serviço. Desde um vendedor ambulante, um carrinho de lanches, passando por pessoas físicas que fornecem serviços com atendimento em um endereço ou à domicílio, até mesmo grandes estabelecimentos. Inclusive você, mesmo hoje não prestando serviços atualmente, poderia passar a divulgar o que sabe fazer, para até quem sabe lhe gerar uma renda extra. Todas as áreas seriam contempladas - vendas, transporte, alimentação, serviços técnicos, domésticos e qualquer outro que você possa imaginar.

Vamos lá?

Desde já, muito obrigado por sua participação.

\*Obrigatório

### Legenda

*	Campo Obrigatório
○	Onde somente uma alternativa deve ser marcada
□	Várias alternativas podem ser marcadas

**Endereço de e-mail \***

Respostas omitidas

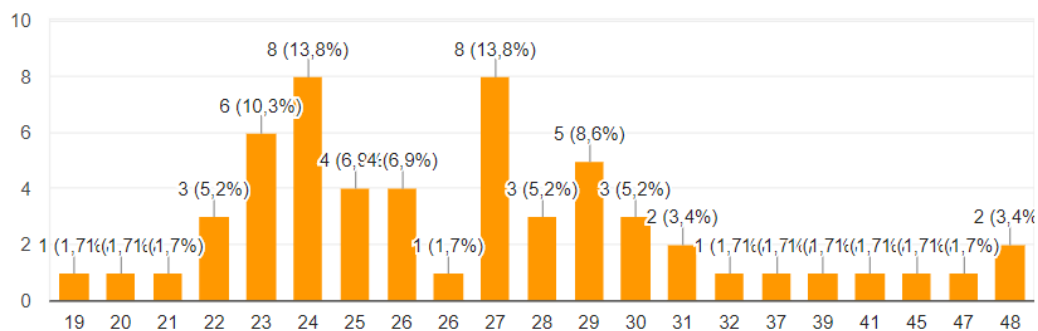
---

**Seu Nome \***

---

**Idade**

58 respostas



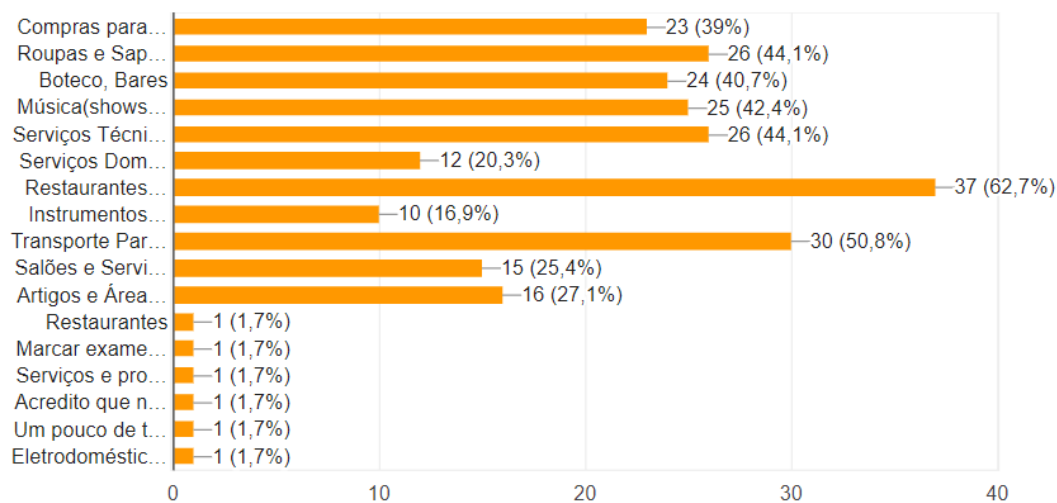
## Você como consumidor

Perguntas direcionadas a você como consumidor de serviços, de todos os gêneros, debatendo seus interesses, dificuldades e sugestões de como a busca e consumo de serviços por meio digital poderia funcionar.

**Quais dos Itens que, para você, seria relevante na hora de buscar um serviço ou produto? \***

- Compras para o Lar
- Roupas e Sapatos
- Boteco, Bares
- Música(shows musicais, artigos musicais, discos)
- Serviços Técnicos(Eletrônico, Refrigeração, Encanador, Pintor, etc)
- Serviços Domésticos(Diarista, Babá)
- Restaurantes e Lanchonetes
- Instrumentos Musicais
- Transporte Particular (MotoTaxi, Taxi, Carona)
- Salões e Serviços de Beleza
- Artigos e Áreas de Esportes
- Outro:

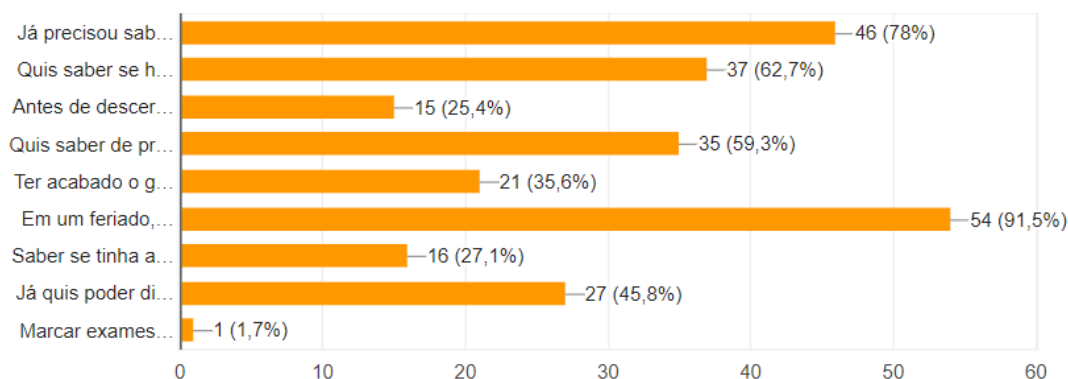
59 respostas



### Você já passou por uma das seguintes situações? quais? \*

- Já precisou saber se um estabelecimento estava aberto, mas só teve como saber se deslocando até o local.
- Quis saber se havia um tipo de técnico ou profissional de qualquer gênero perto de você, que atendesse a domicílio(exemplo: técnico de refrigeração, eletricista, encanador, técnico de informática).
- Antes de descer do ônibus, agendar um moto-taxi para lhe buscar na parada que vai descer.
- Quis saber de promoções de um determinado tipo de serviço próximo de onde você estava
- Ter acabado o gás de cozinha da sua casa e você precisasse saber se tem algum vendedor ambulante de gás próximo de sua rua ou endereço
- Em um feriado, quando normalmente ocorre de estabelecimentos terem seu horário de atendimento fora do normal, querer saber em tempo real se o mesmo encontra-se realmente funcionando
- Saber se tinha alguém perto de você que precisasse de um serviço ou produto que você poderia realizar
- Já quis poder divulgar qualquer tipo de serviço que você pudesse realizar, mesmo não sendo sua área profissional, mas que lhe pudesse render um bom dinheiro
- Outro:

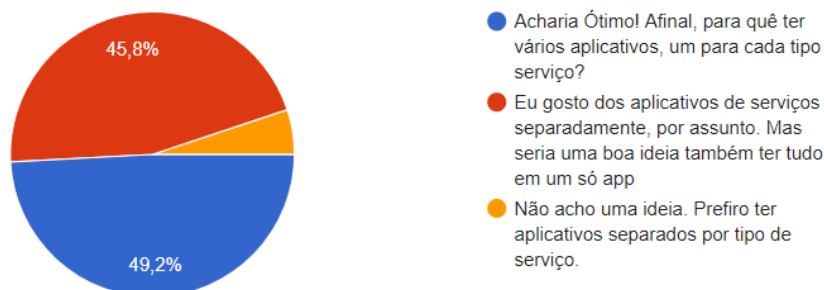
59 respostas



**Atualmente existem muitos aplicativos de serviços( para transporte, alimentação, etc.) Se existisse um aplicativo único que pudesse disponibilizar qualquer tipo de serviço, você... \***

- Acharia Ótimo! Afinal, para quê ter vários aplicativos, um para cada tipo serviço?
- Eu gosto dos aplicativos de serviços separadamente, por assunto. Mas seria uma boa ideia também ter tudo em um só app
- Não acho uma ideia. Prefiro ter aplicativos separados por tipo de serviço.

59 respostas

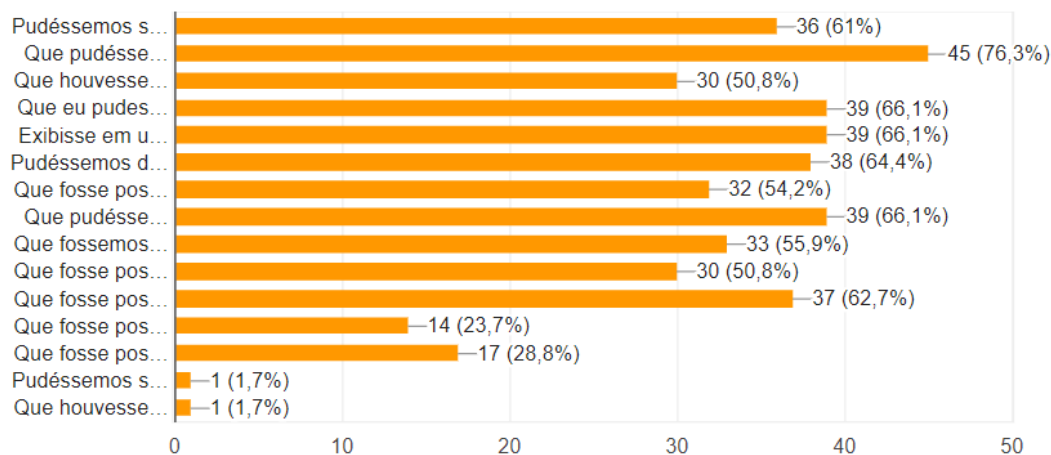


**Seria interessante se em um aplicativo para divulgação de serviços... \***

- Pudéssemos saber o horário fixo de atendimento de serviços e estabelecimentos
- Que pudéssemos saber, em tempo real, que estabelecimentos e prestadores estão "abertos" no exato momento da consulta
- Que houvesse sempre uma sinalização se o local ou a pessoa que presta serviço está realmente atendendo no momento que consultássemos

- Que eu pudesse entrar em contato com um prestador de serviço, para tirar dúvidas, diretamente pelo aplicativo.
- Exhibisse em um mural notícias sobre serviços de meu interesse mediante critérios como localização, preços e condições
- Pudéssemos dar uma nota e 'curtir' estabelecimentos e prestadores de serviço
- Que fosse possível, pra alguns serviços, poder agendar um atendimento futuro
- Que pudéssemos realizar pagamento por um serviço diretamente no aplicativo
- Que fôssemos avisados de promoções sobre área de serviços ou estabelecimentos específicos , próximo de nós.
- Que fosse possível, eu, em meu nome, divulgar minhas habilidades e serviços que eu possa realizar, pra ter uma renda extra.
- Que fosse possível dar uma nota depois de ser atendido por uma pessoa ou estabelecimento
- Que fosse possível ver a localização móvel de prestadores e até pessoas comuns que costumam atender em Domicílio

59 respostas



**Na última pergunta, além das funcionalidades listadas, existe algo a mais que você esperaria de um aplicativo com este propósito? Descreva com suas palavras.**

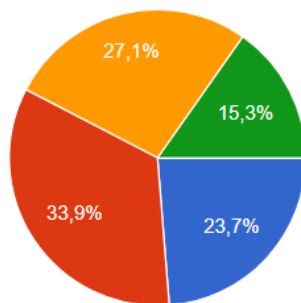
**RESPOSTAS:**

- *Na verdade, o Google já faz o serviço que não marquei na pergunta acima, porém, nem sempre é eficaz... Como sabe se o estabelecimento está de fato aberto. Um mural com os preços tabelados e disponibilidade de chat direto com o estabelecimento é de fato uma boa ideia!*
  - *Não*
  - *O contingente de prestadores de serviços e os custos do serviço listados.*
  - *Classificação em níveis para o consumidor saber a reputação de cada prestador de serviço a partir da notas e curtidas. Também comentários/avaliações de quem foi cliente.*
  - *Notificações de possíveis eventos a serem realizados em tais estabelecimentos*
  - *Atualmente sistema de recompensa para fidelização é uma tendência em qualquer aplicativo.*
  - *Imagens reais dos produtos e serviços oferecidos.*
  - *Um aplicativo que tivesse tudo em um único local, por exemplo: de reserva de restaurante, já me daria: horário, serviço, cardápio, valores, formas de pagamento. Informações que as vezes não temos!*
  - *Que tivesse um design fácil de mexer, visto que teria várias opções de escolha para o consumidor. De preferência que não houvessem anúncios no rodapé do aplicativo.*
  - *Que ele fosse de fácil acesso e constantemente atualizado.*
  - *Localização dos ônibus*
  - *Visualizar e comparar preços entre serviços*
  - *Total segurança sobre os processos*
  - *Pagamento com cartão de credito*
  - *Que fosse possível um atendimento 100% digital, do início ao fim, sem que em algum momento seja necessário utilizar ligação.*
- 

**Você usa algum aplicativo que divulga serviços e estabelecimentos? \***

- Sim, eu uso aplicativos de serviços, e eles atendem às minhas expectativas
- Sim, eu uso aplicativos de serviços, porém eles não atendem à todas as minhas expectativas
- Não, conheço mas não utilizo
- Não, e não conheço nenhum do gênero
- Outro:

59 respostas

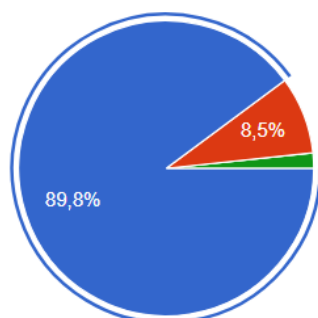


- Sim, eu uso aplicativos de serviços, e eles atendem às minhas expectativas
- Sim, eu uso aplicativos de serviços, porém eles não atendem à todas as minhas expectativas
- Não, conheço mas não utilizo
- Não, e não conheço nenhum do gênero

**Imagine que existisse um aplicativo o qual você pudesse ter a visibilidade em um mapa de todos os tipos de serviço perto de você, de ambulantes, camelôs, barracas de lanche, pessoas comuns que saibam fazer algum serviço, até grandes estabelecimentos. Com um clique você poderia saber sobre as informações de qualquer um desses estabelecimentos ou pessoas, poderia se comunicar com elas por mensagens, contratar pelo aplicativo com cartão de crédito, avaliar os serviços. O que acha da experiência de um aplicativo com esta função na sua vida? \***

- Parece muito boa! Eu usaria! Poderia facilmente descobrir o que tem de útil por perto, pra qualquer necessidade que eu tenha de serviços
- Parece boa! Porém eu não acho que usar um aplicativo com esta função faria diferença em meu cotidiano.
- Achei uma ideia ruim! Não acho que seria útil

59 respostas



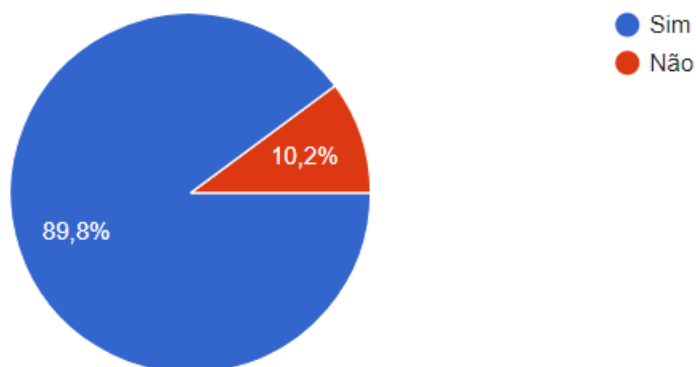
- Parece muito boa! Eu usaria! Poderia facilmente descobrir o que tem de útil por perto, pra qualquer necessidade...
- Parece boa! Porém eu não acho que usar um aplicativo com esta função faria diferença em meu cotidiano.
- Achei uma ideia ruim! Não acho que seria útil
- Parece muito boa! Eu poderia facilmente descobrir o que tem de ú...

**Além do seu emprego e função que exerce, acha que possui alguma outra forma de você ganhar dinheiro? \***

- Sim

Não

59 respostas

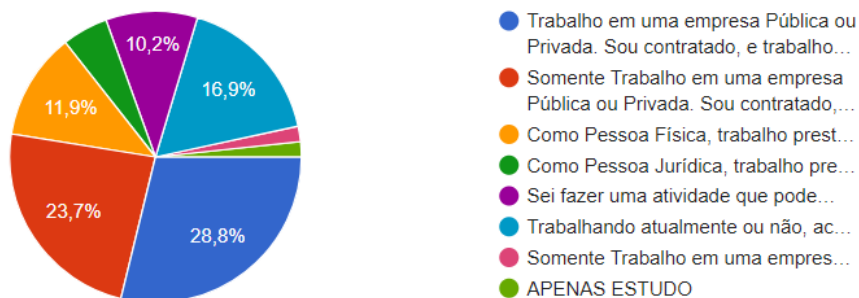


---

**Você se enquadra em qual cenário profissional? \***

- Trabalho em uma empresa Pública ou Privada. Sou contratado, e trabalho com outra atividade além do meu emprego.
- Somente Trabalho em uma empresa Pública ou Privada. Sou contratado, e acho que eu poderia exercer uma atividade extra
- Como Pessoa Física, trabalho prestando serviço e já divulgo meu trabalho em sites e redes sociais
- Como Pessoa Jurídica, trabalho prestando serviço e já divulgo meu trabalho em sites e redes sociais
- Sei fazer uma atividade que pode me render um bom dinheiro, mas não divulgo. Acho que seria bom se as pessoas pudessem ver quem sou, que sei fazer tais atividades, e onde estou para me chamarem
- Trabalhando atualmente ou não, acredito que eu seja apenas um consumidor de serviços, não me identifico prestando nenhum tipo de serviço.
- Outro:

59 respostas



## Você como prestador de serviço

A seguir, algumas perguntas para você que teria o perfil de prestar serviços, de qualquer gênero.

### Que Habilidade ou Serviço você poderia exercer? \*

Por exemplo: técnico de informática, designer, electricista, professor de aulas de reforço, manicure, consultor de marketing, advogado, barraca de lanches, pipoqueiro, xerox, lan house, etc.

#### RESPOSTAS:

*Técnico de informática, desenvolver*

*Corretor de imóveis*

*Na área alimentícia, fornecendo almoços, lanches, doces etc.*

*Designer, ilustrador*

*Empresaria*

*Hortas, serviços de jardinagem, fabricação de compostagem, atividades com plantas em geral.*

*Marketing Digital, Arte digital/manipulação de imagens e projetos web*

*Consulta médica*

*Aulas de reforço*

*Designer*

*consultor de marketing*

*tecnico de informatica*

*Fotografia*

*Técnico de áudio,Produtor Comercial*

*Consultorias contábeis*

*Aulas de Historia e humanidades*

*Fotografia de Eventos*

*Desenvolvedor de sistemas*

*Técnico de Informática, Desenvolvedor de Softwares*

*Consultor de marketing e divulgador de serviços através de redes sociais*

*Tec de informatica, professor, consultor.*

*Estatística*

*redator*

*consultoria e desenvolvimento de de soluções em TI*

*micro empreendimento*

*Consultas médicas a domicilio*

*Designer grafico*

*Culinarista, vendedora, artesã*

*Maquiadora*

*Manutenção de aparelhos eletrônicos*

*Professora de alho de reforço*

*Venda ou troca de roupas e acessórios não mais úteis para mim. Atividade popularmente conhecida como: "brechó"*

*Professor de aulas de reforço.*

*Fotógrafo, advogado, professor*

*Instrutor de curso na area tec de segurança. Montando projetos necessarios para a regulamentacao de empresas nessa area .*

*Organização de festas, cafes da manhã, brindes*

*Tradutor de artigos científicos*

*AULA DE VIOLÃO*

*Técnico de informática.*

*Desenvolvedor de Software*

*Desenvolvimento de softwares/sites*

*Criação de sites*

*Vendas externas*

*Saas*

*ESTETICA Podologia, Depilação*

TECNICO INFORMATICA

Desenvolvimento de Sistemas

Vendedor de produtos

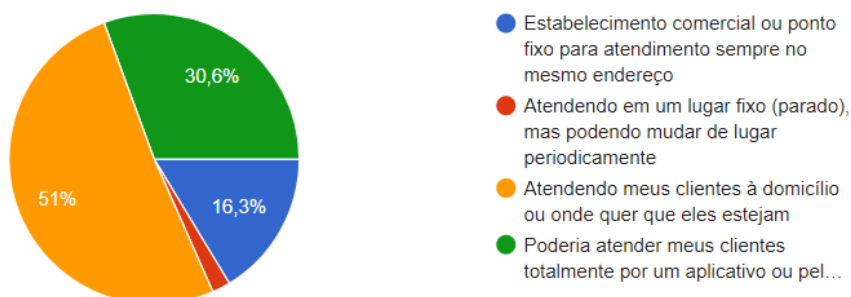
Manutenção de micros, notebooks e smartphones;

---

**Em que cenário o serviço que você exerceria se enquadra \***

- Estabelecimento comercial ou ponto fixo para atendimento sempre no mesmo endereço
- Atendendo em um lugar fixo (parado), mas podendo mudar de lugar periodicamente
- Atendendo meus clientes à domicílio ou onde quer que eles estejam
- Poderia atender meus clientes totalmente por um aplicativo ou pela internet

49 respostas

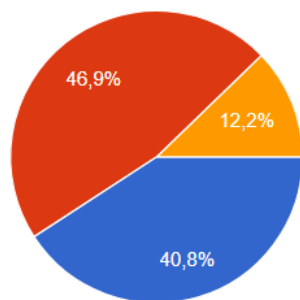


**Para este serviço, acha que seria interessante (ou que fizesse sentido), aceitar pagamento com cartão de crédito? \***

Pagamento com cartão de crédito é sempre atraente aos clientes. O pagamento com cartão de crédito seria do cliente para você direto pelo aplicativo. Mesmo você não possuindo conta para recebimento de pagamento com cartões de créditos, você poderia receber mediante transferência bancária.

- Sim, seria ótimo! Mas seria melhor que fosse sempre com crédito em 1x
- Sim, seria ótimo! Seria bom também parcelar em algumas vezes para o cliente
- Não, acredito que não se enquadra para meu tipo de serviço ou não é de meu interesse

49 respostas



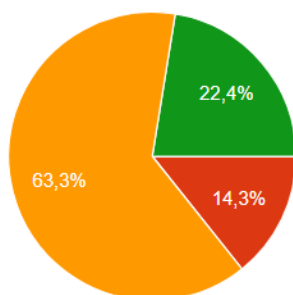
- Sim, seria ótimo! Mas seria melhor que fosse sempre com crédito em 1x
- Sim, seria ótimo! Seria bom também parcelar em algumas vezes para o cliente
- Não, acredito que não se enquadra para meu tipo de serviço ou não é de meu interesse

### Qual a faixa de valores que seus serviços custariam ao cliente? \*

Se o seu serviço engloba vários produtos/atividades com diferentes preços, considere aquele de maior valor na sua resposta.

- Até R\$ 10,00
- Acima de R\$ 10,00 e até R\$ 100,00
- Acima de R\$ 100,00
- Qualquer valor

49 respostas

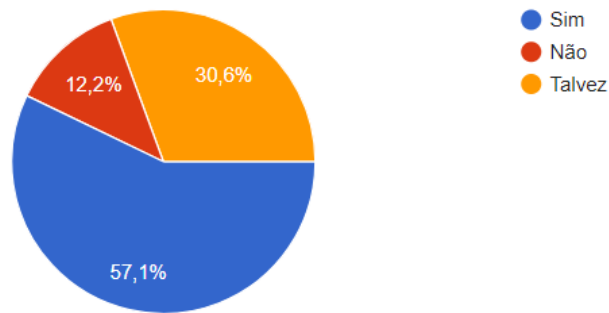


- Até R\$ 10,00
- Acima de R\$ 10,00 e até R\$ 100,00
- Acima de R\$ 100,00
- Qualquer valor

### Seria interessante poder compartilhar sua localização para que outras pudessem saber que está por perto e lhe contatar? \*

- Sim
- Não
- Talvez

49 respostas

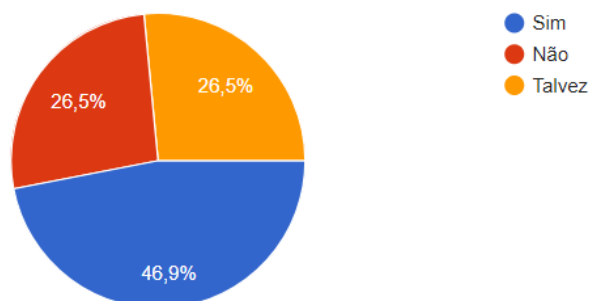


---

**Seria interessante informar uma localização Fixa para que outras pudessem tomar conhecimento que no seu endereço informado pudessem lhe procurar pra serem atendidos? \***

- Sim
- Não
- Talvez

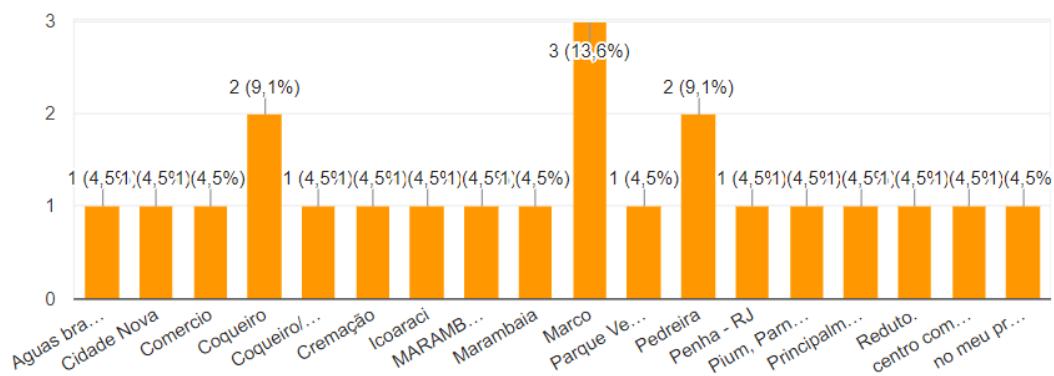
49 respostas



---

**Se na pergunta anterior respondeu sim, quanto ao local de atendimento fixo, em qual bairro seria?**

22 respostas

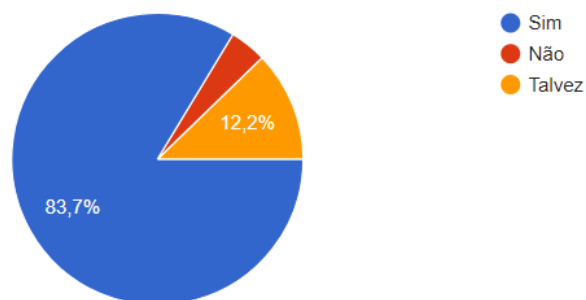


**Seria bom poder, você mesmo ou os interessados, agendarem um horário de atendimento para que você os atenda em uma data e hora futura? \***

É interessante para consultórios, serviços à domicílio, aluguel de salas, de equipamentos, etc..

- Sim
- Não
- Talvez

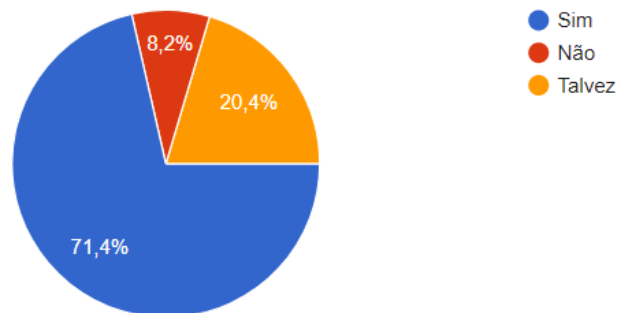
49 respostas



**Seria bom para seu negócio poder realizar um fluxo de caixa, controlando valores diariamente ou por turnos? \***

- Sim
- Não
- Talvez

49 respostas



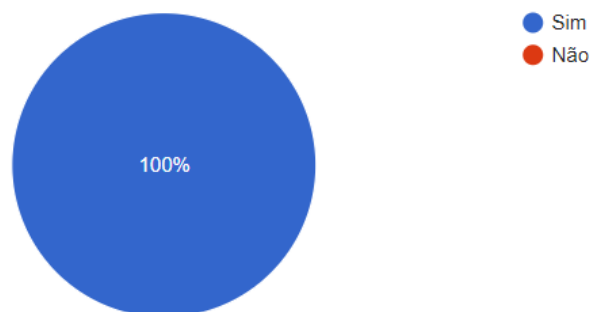
---

**Então, se existisse um aplicativo que possuísse essas opções para você divulgar seus serviços, você usaria? \***

Sim

Não

49 respostas



---

## Contratação de seus serviços pelo app

Algumas perguntas que definiriam melhor suas condições para atender pessoas as quais se interessassem pelo seu serviço ou produto no aplicativo

**Quais dias da semana você poderia atender seus clientes? \***

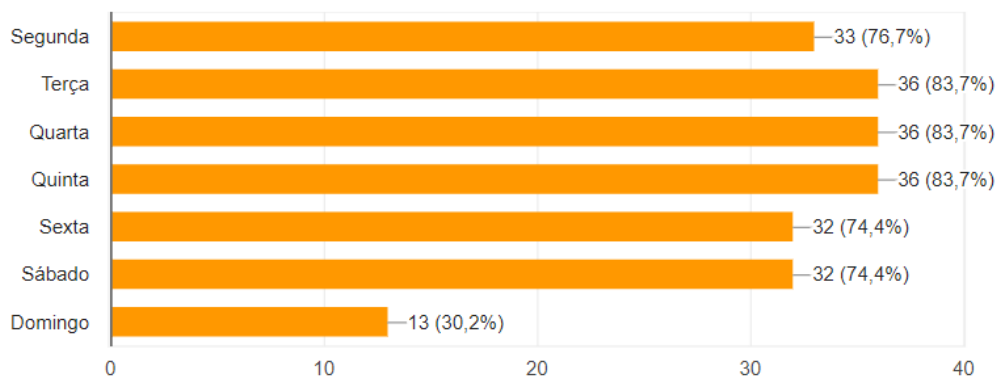
Segunda

Terça

Quarta

- Quinta
- Sexta
- Sábado
- Domingo

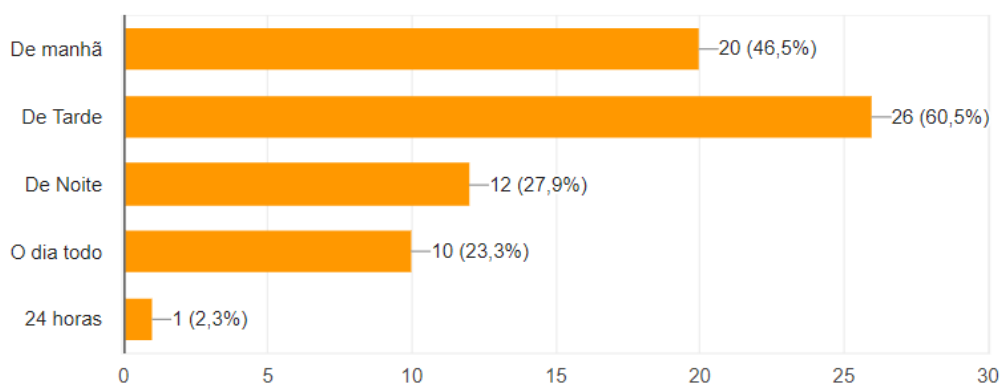
43 respostas



#### E os horários, nestes dias, para atender os interessados pelo serviço? \*

- De manhã
- De Tarde
- De Noite
- O dia todo
- 24 horas

43 respostas

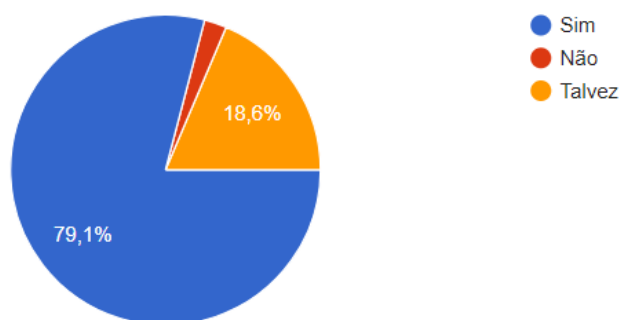


**E se houvesse uma forma de seus clientes e interessados se cadastrarem(opcionalmente) para contratar seus serviços, onde o cadastro e as informações solicitadas fosse você que definisse. Seria útil ao seu negócio? \***

Por exemplo, você poderia mostrar um formulário de perguntas ao seu cliente antes de efetivar o atendimento ou agendamento do serviço com você. Isso seria, útil, por exemplo, se você acha interessante gerar relatórios de clientes, categorizando por tipo de cliente, e outras informações que você mesmo definiria. Exemplo: qualquer pessoa que quisesse te contratar, informaria nome, endereço, telefone, preferência por tipo do seu serviço, etc.

- Sim
- Não
- Talvez

43 respostas

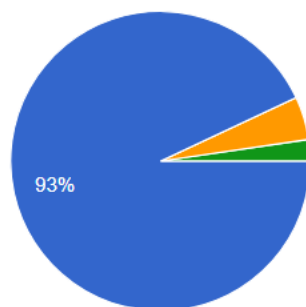


**O que acha da ideia de divulgar seu serviço em um aplicativo, e fazendo um bom número de atendimento direto na aplicação, você e seu negócio ganhassem um destaque para todos os demais usuários? \***

Os prestadores de cada tipo de serviço que mais atendessem pelo app ganhariam destaques nas pesquisas feitas pelos consumidores na aplicação

- Muito bom! Seria ótimo poder conseguir atingir essa marca e aumentar minha visibilidade.
- Acredito que seria difícil eu conseguir isto.
- Não faria diferença para meu negócio ganhar destaque em um aplicativo

43 respostas



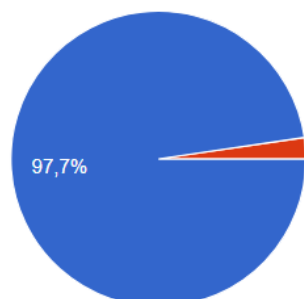
- Muito bom! Seria ótimo poder conseguir atingir essa marca e aumentar minha visibilidade.
- Acredito que seria difícil eu conseguir isto.
- Não faria diferença para meu negócio ganhar destaque em um aplicativo
- Muito bom! Seria ótimo poder conseguir atingir essa a marca e aumentar minha visibilidade.

**Para utilizar um aplicativo como este, que poderia lhe ajudar a aumentar a divulgação de seu serviço, estaria disposto a pagar uma comissão sobre seus serviços com um percentual menor que 5% sobre os atendimentos que você realizasse pela aplicação? \***

Você só pagaria esta taxa para atendimentos realizados pela plataforma, e quando realizasse. Se você ganhasse em um atendimento R\$ 100,00 , pagaria uma taxa de até R\$ 5,00 para o aplicativo.

- Acho justo! Afinal, se o aplicativo me ajudasse a ganhar mais clientes, seria ótimo para meu negócio
- Acho injusto! Eu não estaria disposto a pagar este percentual ao fornecer meus serviços.

43 respostas



- Acho justo! Afinal, se o aplicativo me ajudasse a ganhar mais clientes, seria ótimo para meu negócio
- Acho injusto! Eu não estaria disposto a pagar este percentual ao fornecer meus serviços.

**Se o aplicativo lhe ajudasse a conseguir novos clientes (uma quantidade boa para seu negócio), você estaria disposto a dar um desconto para estes clientes novos? \***

- Sim! Seria ótimo conseguir expandir minha rede de clientes. Eu daria o desconto a eles.

Não! Não acho que seja interessante para meu negócio.

43 respostas



## Contato Futuro

**Gostaria de expor algum comentário geral sobre o questionário ou a ideia proposta?**

### RESPOSTAS:

*Caso estruturado detalhadamente com o auxílio de profissionais das diversas áreas correlatas ao aplicativo, será muito bem feito.*

*Espero que fabriquem essa idéia! ela é custo beneficio*

*Gostaria que esse aplicativo realmente pudesse vigorar ,acho que seria importante tanto a comercialização tanto para consumidores.*

*É uma proposta bem interessante. Que se colocada em prática pode ajudar muita gente a expandir pequenos negócios.*

*Que esse aplicativo seja de fato criado.*

*As idéias dos aplicativos são interessantes devido o período informatizado que estamos vivenciando.*

*O google maps faz boa parte do que a proposta desse aplicativo tenta atender, porém, em alguns momentos o GM é muito deficiente e falho, principalmente com relação ao comércio informal e pequenos estabelecimentos, tampouco há no GM um mecanismo que me permita pagar pelos serviços contratados ou receber por serviços realizados.*

*Estamos realmente precisando de um aplicativo que nos ajude a entrar no mercado.*

*Bem pensado*

*Achei bem inteligente e amplo*

*Bom trabalho!*

*INTERESSANTE*

*Expõe uma excelente idéia*

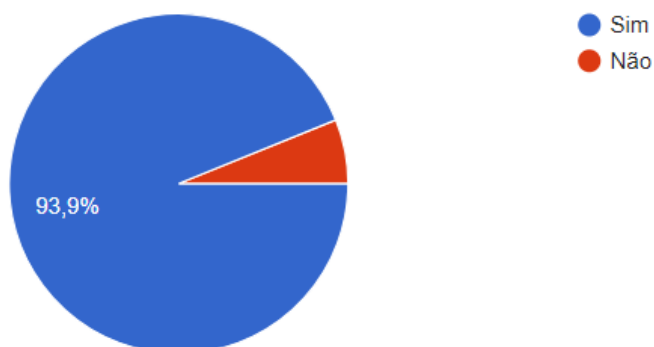
---

**Você aceitaria receber um e-mail, futuramente, quando este aplicativo for lançado, para se cadastrar e experimentar? \***

Sim

Não

49 respostas



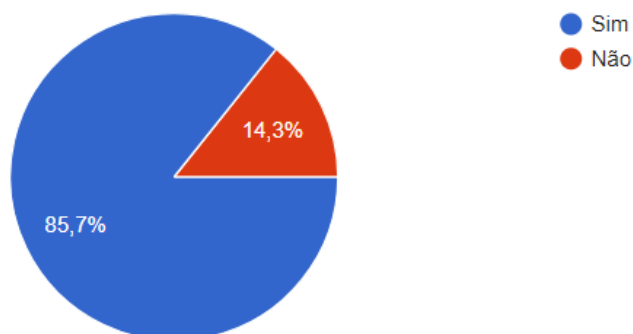
---

**Você aceita fazer ser voluntário na etapa de testes do aplicativo, para experimentá-lo antes de seu lançamento e contribuir com suas opiniões para melhorar ferramenta? \***

Sim

Não

49 respostas



---

**Se você aceita receber e-mail de contato, por favor informe abaixo**

Respostas omitidas

**APÊNDICE E – QUESTIONÁRIO PARA AVALIAÇÃO DE USABILIDADE**

## Formulário de Usabilidade IService

\*Obrigatório

### Rapidez de aprendizado \*

Em uma escala de 1 a 5 , onde 1 é pouco e 5 é muito.

	1	2	3	4	5
Nota	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

### Facilidade de uso, depois de aprendido \*

Em uma escala de 1 a 5 , onde 1 é pouco e 5 é muito.

	1	2	3	4	5
Nota	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

### O quão você acha que o uso pode ser fácil de ser memorizado / lembrado ? \*

Em uma escala de 1 a 5 , onde 1 é pouco e 5 é muito.

	1	2	3	4	5
Nota	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

### O aplicativo é agradável, bom de ser utilizado \*

Em uma escala de 1 a 5, onde 1 é pouco e 5 é muito.

	1	2	3	4	5
Nota	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

### O aplicativo é rápido para executar as tarefas \*

Em uma escala de 1 a 5, onde 1 é pouco e 5 é muito.

	1	2	3	4	5
Nota	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

---

Frequência de vezes que não encontrou uma opção que esperava encontrar, como um link, uma mensagem de ajuda, um botão \*

- Nenhuma
  - Baixa
  - Alta
- 

Frequência de vezes que não soube o que fazer. Algo como um pensamento : "E agora?" \*

- Nenhuma
  - Baixa
  - Alta
-

Frequência de vezes que você sentiu ter ficado sem resposta do sistema sobre algo que aconteceu? \*

Quando o sistema não emite uma mensagem ou alerta a você

- Nenhuma
- Baixa
- Alta

---

Gostaria de acrescentar comentários, elogios, críticas ?

Sua resposta

---