



**Everton de A. Costa RA 920202060**

**Felipe Domingos de Andrade RA 918200978**

**Gabriel Rabassi Mussulini Carnelosso RA 3019105474**

**Hericles Gustavo Araújo de Melo RA 2217202399**

**Lucas Vidal de Sousa RA 915202089**

**Luiz Henrique Ramos RA 918111335**

**Marcus Vinicius dos Santos Borges RA 918203359**

**Marcelo Santos de Jesus RA 920202061**

**Philip Korb RA 918101434**

**Rafael Galdino Andrade RA 918205778**

**PROJETO EM COMPUTAÇÃO APLICADA**

**Chatbot**

SÃO PAULO - SP  
2021

**Everton de A. Costa RA 920202060**  
**Felipe Domingos de Andrade RA 918200978**  
**Gabriel Rabassi Mussulini Carnelosso RA 3019105474**  
**Hericles Gustavo Araújo de Melo RA 2217202399**  
**Lucas Vidal de Sousa RA 915202089**  
**Luiz Henrique Ramos RA 918111335**  
**Marcus Vinicius dos Santos Borges RA 918203359**  
**Marcelo Santos de Jesus RA 920202061**  
**Philip Korb RA 918101434**  
**Rafael Galdino Andrade RA 918205778**

## **PROJETO EM COMPUTAÇÃO APLICADA**

### **Chatbot**

Trabalho apresentado à Universidade Nove de Julho, UNINOVE, em cumprimento parcial às exigências da disciplina de Projeto em Computação Aplicada, sob orientação do Prof. **Edson Melo de Souza**

# **IMÓVEL**

## **INTEGRANTES**

**Everton de Albuquerque Costa (920202060) - everton.albuquerque@uni9.edu.br**

**Felipe Domingos de Andrade (918200978) - felipedomingos1@uni9.edu.br**

**Gabriel Rabassi Mussulini Carnellosso (3019105474) - Gabriel.rabassi@uni9.edu.br**

**Hericles Gustavo Araújo de Melo (2217202399) - hericlesaraujo@uni9.edu.br**

**Lucas Vidal de Sousa (915202089) - lucassousa832@uni9.edu.br**

**Luiz Henrique Ramos (918111335) - luizhrb@uni9.edu.br**

**Marcelo Santos de Jesus (920202061) - marcelosj2004@uni9.edu.br**

**Marcus Vinicius dos Santos Borges (918203359) - mvinicius\_borges1@uni9.edu.br**

**Philip Korb (918101434) - pkorb96@uni9.edu.br**

**Rafael Galdino Andrade (918205778) - rafael.galdino@uni9.edu.br**

## 1. Introdução

Aos lembrarmos de locação de imóveis automaticamente lembramos de corretores, com agendas cheias, horários de visitas sempre escassos, estacionamentos que não eram gratuitos.

Para melhorar a eficácia de empreendimento dos imóveis para locação, desenvolvemos uma ferramenta onde podemos filtrar os locatários de forma mais simples, automática, do conforto de suas casas ou de qualquer local de disponibilize acesso à internet, fazendo com que o tempo dedicado seja apenas para candidatos que se enquadram no perfil do imóvel, onde utilizamos a ferramenta Selenium para automatizar o Chatbot, no qual não utiliza APIs para o seu funcionamento.

Os candidatos serão selecionados por meio do WhatsApp, respondendo um questionário em questões. Tais perguntas ajudam na hora de escolher um perfil, verificando como positivo ou negativo para locação, entre elas, para quantas pessoas seriam a locação, e quantas delas têm emprego fixo, aposentadas ou pensionistas. Utilizando esses dados, o sistema elabora os perfis dos locatários e faz uma comparação com os requisitos do locador x respostas do locatário.

A partir do momento que o sistema considera um resultado positivo em relação ao locatário, o Chatbot envia ao locador o contato do mesmo, juntamente com as informações respondidas na plataforma, reduzindo o volume de atendimentos de clientes não compatíveis com os imóveis, e reduzindo o volume de questões a serem esclarecidas. Assim aumentando a margem de compatibilidade e reduzindo o tempo de trabalho.

**Resumo.** Foi desenvolvido um Chatbot para whatsapp, onde o locador informa os parâmetros para o perfil do locatário que deseja e com base nessas informações e respostas do usuário, ele elimina as pessoas não compatíveis para alocar o imóvel, as pessoas compatíveis para a locação do imóvel ele separa e manda as informações das respostas para outro contato.

**Objetivo:** Desenvolver um Chatbot que seleciona as pessoas compatíveis para alocar um imóvel com base no formulário pré-informado pelo locador.

**Metodologia:** O desenvolvimento foi realizado com a utilização da linguagem python junto Selenium.

**Resultados:** Os métodos usados para o desenvolvimento e implementação propostos mostram que as técnicas e métodos utilizados permitem que o chatbot controle quem está apto para locar o imóvel com base no formulário.

**Conclusão:** As técnicas empregadas neste projeto permitiram mitigar tempo e custo apresentando uma solução mais eficaz de selecionar as pessoas compatíveis para a locação de imóvel.

## 2. Fundamentação Teórica

O trabalho apresentado é um desenvolvimento de uma plataforma para avaliar locatários automaticamente, assim conseguiremos avaliar sua eficácia e automatizar processos, promovendo ganho de tempo e redução de custos.

Portanto é importante ressaltar que se trata também de uma nova abordagem, um novo plano de ação para lidar com nossos consumidores.

A abordagem inicial do projeto foi desenvolvida a partir de uma necessidade real, de um dos integrantes:

"Criar um chatbot para auxiliar os proprietários de imóveis locarem suas casas.

Por exemplo, meu tio tem um apartamento que ele gostaria de alugar para uma pessoa, mas ele tem um perfil de pessoa em mente. Além disso ele tem que ficar respondendo muitos contatos e nem sempre com o perfil que ele procura.

Fora que tem de combinar horário de visita etc..." (Rafael Galdino Andrade, 2021).

### 2.1 Selenium

Selenium é um projeto que abrange uma variedade de ferramentas e bibliotecas que permitem e suportam a automação de navegadores da web.

Ele fornece extensões para emular a interação do usuário com os navegadores, um servidor de distribuição para escalonar a alocação do navegador, e a infraestrutura para implementações da Especificação W3C Web Driver que permite escrever código intercambiável para todos os principais navegadores da web.

O Selenium fornece uma ferramenta de reprodução para a criação de testes funcionais sem a necessidade de aprender uma linguagem de script de teste (Selenium IDE). Ele também fornece uma linguagem de domínio específico de teste, para escrever testes em várias linguagens de programação populares, incluindo **C#, Groovy, Java, Perl, PHP, Python, Ruby e Scala**. Os testes podem ser executados na maioria dos navegadores web modernos. Selenium é executado em Windows, Linux e macOS. É um software de código aberto lançado sob

a licença Apache 2.0.( *Apache License* em inglês é uma licença de software livre permissiva de autoria da Apache Software Foundation (ASF).

Todo *software* produzido pela ASF ou qualquer um dos seus projetos e subprojetos é licenciado de acordo com os termos da licença Apache. Alguns projetos não pertencentes à ASF também utilizam esta licença. A licença Apache (versões 1.0, 1.1 e 2.0) exigem a inclusão do aviso de *copyright* e um *disclaimer*, mas não é uma licença *copyleft* - ela permite o uso e distribuição do código fonte tanto no *software open source* como no software proprietário.)

### 3. Metodologia

Importação das bibliotecas.

```
ChatBot.py > ...
1  from time import sleep
2  from datetime import datetime
3  from selenium import webdriver
4  from selenium.webdriver.common.keys import Keys
5  from selenium.webdriver.common.by import By
6  from selenium.webdriver.support.ui import WebDriverWait
7  from selenium.webdriver.support import expected_conditions as EC
8  import selenium.common.exceptions
9
```

Inicialização do objeto webdriver com o drive do Google Chrome (Versão 96.0.4664.45) e inicialização do whatsapp web.

```
10  chrome = webdriver.Chrome(executable_path=r'./chromedriver.exe')
11  chrome.get('https://web.whatsapp.com/')

```

Fica no laço até encontrar o círculo com a img do perfil. Isso evita que o programa de erro na etapa de autenticação pelo QRCode.

```
25  while True:
26      try:
27          WebDriverWait(chrome, 10).until(EC.presence_of_element_located((By.XPATH, '//*[@id="side"]/header/div[1]/div/img')))
28          break
29      except selenium.common.exceptions.TimeoutException:
30          sleep(5)
31  print(datetime.today().strftime('%Y-%m-%d %H:%M:%S'), 'Whatsapp identificado.')
32  chrome.find_element(By.XPATH, '//*[@id="side"]/header/div[1]/div/img').click()
33  sleep(5)

```

Logo após encontrar o círculo com a imagem do perfil, captura o nome do usuário tanto no whatsapp web quanto no business.

```
35  try:
36      # Extrai o nome do usuário no whatsapp business
37      user = chrome.find_element(By.XPATH, '//*[@id="app"]/div[1]/div[1]/div[2]/div[1]/span/div[1]/span/div[1]/div/div[2]/div/div[1]').text
38  except:
39      # Extrai o nome do usuário no whatsapp
40      user = chrome.find_element(By.CLASS_NAME, '_13Nkt').text
41  finally:
42
43      chrome.find_element(By.XPATH, '//*[@id="app"]/div[1]/div[1]/div[2]/div[1]/span/div[1]/span/div[1]/header/div/div[1]/button').click()
44      print(datetime.today().strftime('%Y-%m-%d %H:%M:%S'), 'Nome do usuário capturado.')
45      print(datetime.today().strftime('%Y-%m-%d %H:%M:%S'), 'Procurando conversa não lida.')
46      telefone = ''

```

Entra no laço infinito principal do programa.

```
48  while True:

```

Entra em outro laço infinito logo em seguida e permanece nele até encontrar a div onde é armazenado a lista de conversas. Isso evita que o programa de erro caso não tenha nenhuma conversa.



```

52     while True:
53         try:
54             WebDriverWait(chrome, 10).until(EC.presence_of_element_located((By.XPATH, '//*[@id="pane-side"]/div[1]/div/div')))
55             break
56         except selenium.common.exceptions.TimeoutException:
57             sleep(5)
58

```

Logo após sair do laço anterior, armazena dentro da variável “lc” as conversas que são subelementos da div que armazena a lista de conversas. Depois cria um array, e insere as conversas dentro dele e inverte a ordem dos elementos.

```

59     lc = chrome.find_element(By.XPATH, '//*[@id="pane-side"]/div[1]/div/div').find_elements(By.CLASS_NAME, '_3m_Xw')
60
61     # Lista de conversas organizada
62     lco = []
63
64     # Organiza lista de conversas
65     for c in lc:
66         lco.append(c)
67
68     lco.reverse()

```

Seleciona uma conversa por vez da lista de conversas até encontrar uma conversa que tenha a Class “\_1pJ9J”. Essa Class no HTML só está presente quando existe uma mensagem na conversa que não foi lida pelo usuário.

```

70     # Seleciona uma conversa por vez da lista de conversas organizada
71     for c in lco:
72
73         # Verifica se é uma mensagem que não foi lida
74         # Só responde uma mensagem específica
75         if is_exist(c, '_1pJ9J'):
76             span = c.find_element(By.CLASS_NAME, '_ccCW')
77             telefone = span.get_attribute('title')
78             print(datetime.today().strftime('%Y-%m-%d %H:%M:%S'), 'Encontrou uma mensagem não lida do', span.get_attribute('title'))
79
80             c.click()

```

Após encontrar uma conversa que tenha uma mensagem que não foi lida, o programa entra em outro laço infinito e permanece nele interagindo com o cliente até encontrar uma mensagem não lida de outro cliente na lista de conversas.

```

82     # Fica no laço até encontrar uma mensagem não lida de outro usuário na lista de conversas.
83     while True:

```

Armazena na variável “lm” as mensagens da conversa que está sendo analisada.

```

84     # Pega as mensagens de uma conversa
85     lm = chrome.find_element(By.CLASS_NAME, 'y8WcF').find_elements(By.CLASS_NAME, '_2wUmf')

```

Inicializa as variáveis que vão armazenar as respostas do cliente.

```

92     TemAnimalDomestico = ''
93     QtdePessoasLocacao = 0
94     QtdeCrianças = 0
95     QtdePessoasAutonomas = 0
96     QtdePessoasAssalariadas = 0
97     BarulhoExcessivo = ''
98     ClientName = ''

```

Analisa uma mensagem por vez da conversa.

```
100  ▾  for m in lm:
```

Captura as respostas do cliente e a última mensagem enviada pelo Chatbot.

```
105 ▾ if user in emissor:
106     UltimaMsgRobo = texto
107     UltimaMsgCliente = ''
108     x = True
109 ▾ else:
110     ▾ if x:
111         UltimaMsgCliente = texto
112
113         # Armazena as respostas do usuário nas variáveis
114     ▾ if UltimaMsgRobo == 'Você possui animal doméstico? Digite somente SIM ou NÃO.':
115         TemAnimalDomestico = UltimaMsgCliente
116     ▾ elif UltimaMsgRobo == 'A locação seria para quantas pessoas? Digite somente o número.':
117         QtdePessoasLocacao = UltimaMsgCliente
118     ▾ elif UltimaMsgRobo == 'Quantas são crianças? Digite somente o número e caso não tenha, digite o número zero.':
119         QtdeCrianças = UltimaMsgCliente
120     ▾ elif UltimaMsgRobo == 'Quantas pessoas tem emprego fixo ou são aposentadas ou são pensionistas? Digite somente o número e caso não tenha, digite o número e caso não tenha, digite o número zero.':
121         QtdePessoasAssalariadas = UltimaMsgCliente
122     ▾ elif UltimaMsgRobo == 'Barulho excessivo no imóvel, a qualquer horário do dia, é proibido. Você está de acordo? Digite somente SIM ou NÃO.':
123         BarulhoExcessivo = UltimaMsgCliente
124     ▾ elif UltimaMsgRobo == 'Qual é seu nome?':
125         ClientName = UltimaMsgCliente
126
127 ▾ if UltimaMsgCliente != '':
128     x = False
```

Analisa a última mensagem do Chatbot e do cliente para identificar qual etapa da conversa o cliente está e processar a resposta do cliente com base nas preferências do locatário.

```
130 > if UltimaMsgRobo == 'Você possui animal doméstico? Digite somente SIM ou NÃO.' and UltimaMsgCliente != '':...
143
144 > elif UltimaMsgRobo == 'A locação seria para quantas pessoas? Digite somente o número.' and UltimaMsgCliente != '':...
161
162 > elif UltimaMsgRobo == 'Quantas são crianças? Digite somente o número e caso não tenha, digite o número zero.' and UltimaMsgCliente != '': ...
178
179 > elif UltimaMsgRobo == 'Quantas pessoas tem emprego fixo ou são aposentadas ou são pensionistas? Digite somente o número e caso não tenha, digite o número e caso não tenha, digite o número zero.' and UltimaMsgCliente != '': ...
195
196 > elif UltimaMsgRobo == 'Barulho excessivo no imóvel, a qualquer horário do dia, é proibido. Você está de acordo? Digite somente SIM ou NÃO.' and UltimaMsgCliente != '': ...
210
211 > elif UltimaMsgRobo == 'Qual é seu nome?' and UltimaMsgCliente != '':...
214
215 > elif UltimaMsgRobo == 'Retornaremos seu contato em breve para agendar uma visita e esclarecer qualquer dúvida que tenha sobre o imóvel e a locação.' and UltimaMsgCliente != '':...
236
237 > elif (UltimaMsgRobo == '') or UltimaMsgRobo == 'Lamento, seu perfil não é compatível para fazer a locação do imóvel. Agradecemos seu contato.'
244
```

Verifica se existe uma mensagem não lida de outro cliente na lista de conversas. Se existir, sai do laço da linha 83, retorna para o laço principal da linha 48 e interage com o novo cliente. Caso contrário permanece interagindo com o cliente atual.

```
259 # Seleciona uma conversa por vez da lista de conversas organizada
260 ▾ for c2 in lco2:
261     # Verifica se é uma mensagem que não foi lida
262     ▾ if is_exist(c2, '_1pJ9J'):
263         exit = True
264         break
265
266 ▾ if exit:
267     break
```

Existe um vídeo do Chatbot em execução na pasta do projeto.

[https://drive.google.com/file/d/1dVFUkedhsOnSvwP\\_5XIFRIRAcwqLoJ7A/view?usp=sharing](https://drive.google.com/file/d/1dVFUkedhsOnSvwP_5XIFRIRAcwqLoJ7A/view?usp=sharing)

## 4. Como Executar o Chatbot

### Para executar o Chatbot:

0 - Download do Chrome, caso não tenha (Chatbot somente funciona com o Google Chrome)

1 - Download python 3.9.7

2 - Download vscode

3 - Instalar a extensão do Python no vscode

4 - Alterar a variável "home" no arquivo "\venv\pyvenv.cfg" com o diretório onde fica o executável do python na sua máquina

Exemplo

home = D:\Python\Python39 (Antes)

home = C:\XXX\ZZZ\RRR\Python39 (Depois)

5 - Selecionar o CMD como Terminal padrão para o vscode

1 - Aperte a tecla Ctrl + "

2 - Selecione V

3 - Select Default Profile

4 - Click a opção CMD

6 - Selecionar o interpretador python da venv para rodar com o ambiente virtual

1 - Aperte: Ctrl + Shift + P

2 - Escreva: python:Select interpreter

3 - Selecione o interpretador que tem "venv"

7 - Fim

\* Use o comando abaixo no powershell como administrador para liberar a execução, caso necessário

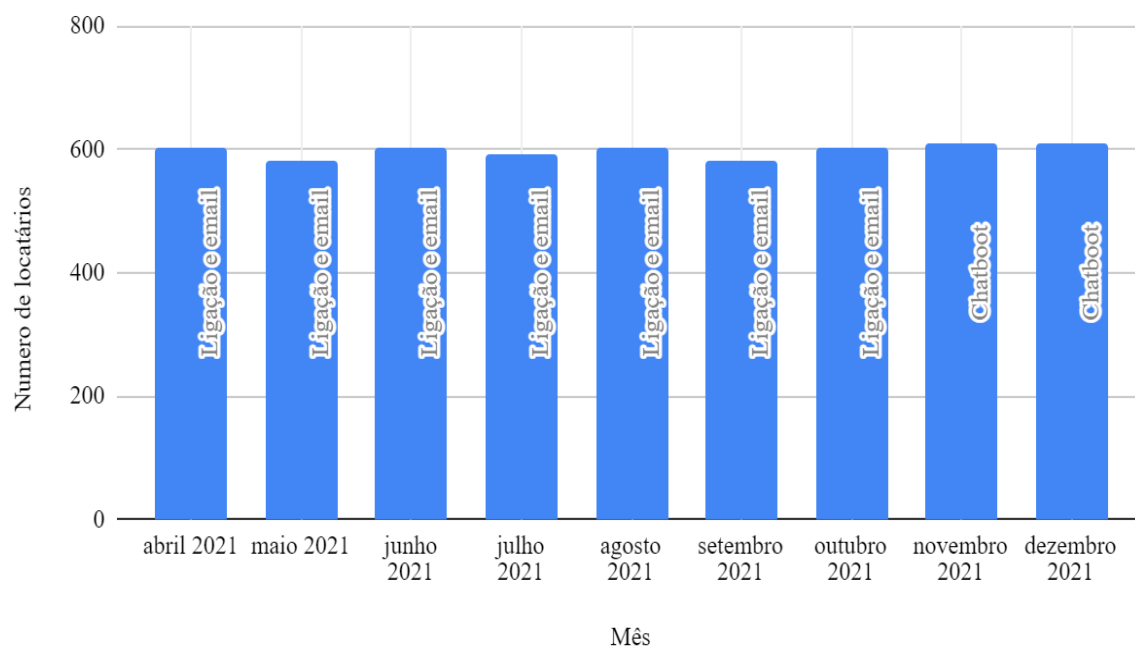
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

## 5. Análise dos Resultados

**Durante o período de teste 01-11-2021 - 21/12/2021 identificamos que:**

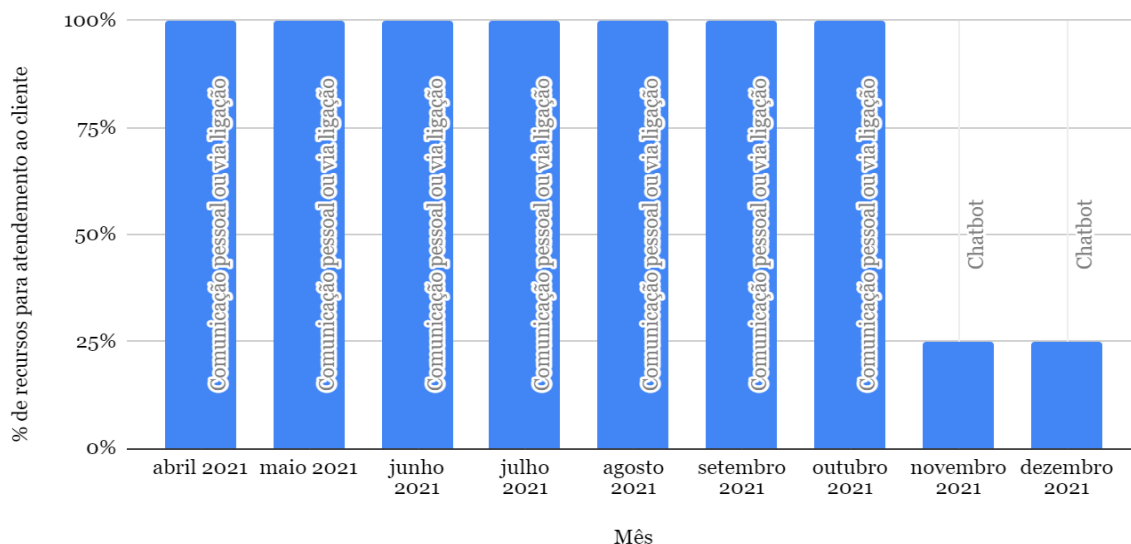
**Em relação ao número de locatários à procura:** não houve redução por conta da nova abordagem de atendimento podendo considerar positivo, e o número de atendimentos foi diminuído sem ter redução na conversão de locações comprovando que o aproveitamento está melhor.

Locatários antes e depois do chatboot mantendo a mesma proporção de atendimento



**Em relação à rentabilidade:** Houve uma diminuição dos gastos com recursos para atendimento, faltas de agendamentos e tratativas sem conversão de locação.

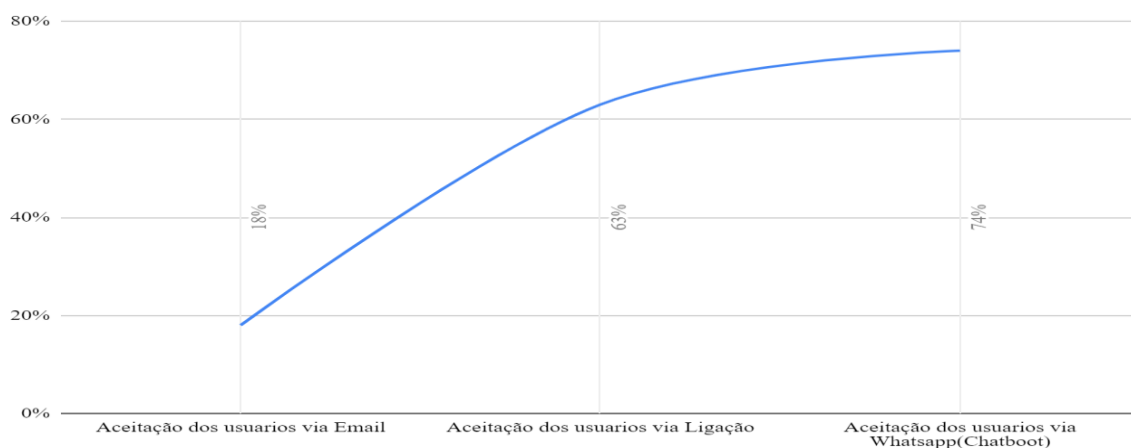
## Queda da utilização de recursos para comunicação apos implantação do chatboot



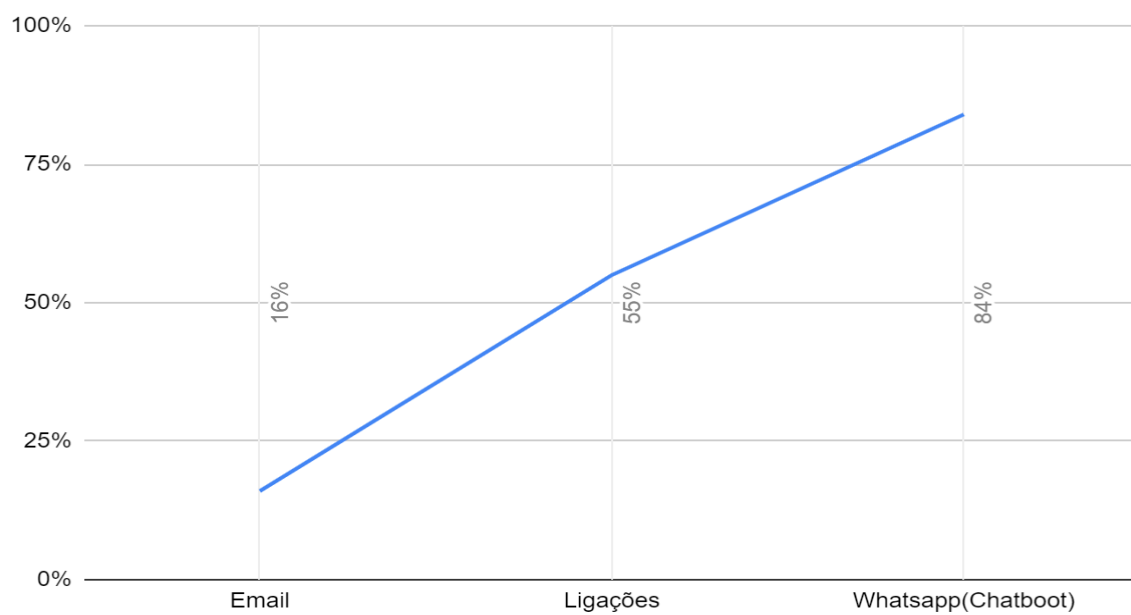
**Em relação à Lucratividade:** Somente houve aumento em relação a diminuição de gastos.

**Em relação aos usuários:** Houve melhor aptidão na pesquisa de acordo com a implantação do chatbot onde inicia-se com uma melhor abordagem, o designer tem aptidão nos resultados trazendo especificamente os melhores resultados para o usuário. Além disso, são mais acessíveis pois utilizam uma plataforma onde já estão habituados, o Whatsapp.

**Gráfico referente a aceitação dos usuários a plataforma de atendimento:**



### Gráfico referente ao Feedback dos usuários:



## 6. Conclusão

Neste trabalho abordamos o *desenvolvimento de um chatbot de whatsapp para uma locadora de imóveis, onde com base nas informações e respostas do usuário, ele verifica se a pessoas está apta para alocar o imóvel, e concluímos* que a nova plataforma de whatsapp desenvolvida pelo 'iMóvel' foi um sucesso no final do projeto, pois atende os requisitos iniciais solucionando os problemas apresentados, sem impactar negativamente no processo de locação como um todo.

## 7. Referências

Frigeri, M. and Monteiro, M. S. A. (2014) Qualis Periódicos: indicador da política científica no Brasil?. *Estudos de Sociologia*, v. 19, n. 37, p. 229-315.

[https://pt.wikipedia.org/wiki/Aplica%C3%A7%C3%A3o\\_web](https://pt.wikipedia.org/wiki/Aplica%C3%A7%C3%A3o_web)

[https://pt.wikipedia.org/wiki/Selenium\\_\(software\)](https://pt.wikipedia.org/wiki/Selenium_(software))

<https://www.selenium.dev/pt-br/documentation/>

[https://pt.wikipedia.org/wiki/Licen%C3%A7a\\_Apache](https://pt.wikipedia.org/wiki/Licen%C3%A7a_Apache)