

**UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIENCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE COMPUTAÇÃO**

**GABRIEL MIRANDA SILVA**

**DESENVOLVIMENTO DE UMA RPA APLICADA NO CONTEXTO DE UM *CALL*  
*CENTER***

**João Monlevade**

**2022**

**GABRIEL MIRANDA SILVA**

**RPA APLICADA NO CONTEXTO DE UM *CALL CENTER***

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Orientador: Marlon Paolo Lima

Coorientador: Victor Geraldo Gomes

**João Monlevade**

**2022**



## FOLHA DE APROVAÇÃO

**Gabriel Miranda Silva**

### **Desenvolvimento de uma RPA aplicada no contexto de um call center**

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel

Aprovada em 03 de novembro de 2022

#### Membros da banca

Dr. Marlon Paolo Lima - Orientador (UFOP)  
Me. Victor Geraldo Gomes - Coorientador (PontoBR Sistemas)  
Dr. Rafael Frederico Alexandre (UFOP)  
Dr. Lucinéia Souza Maia (UFOP)

O professor Marlon Paolo Lima, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 09/02/2023.



Documento assinado eletronicamente por **Marlon Paolo Lima, COORDENADOR(A) DE CURSO DE SISTEMAS DE INFORMAÇÃO**, em 09/02/2023, às 21:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0472827** e o código CRC **8BD97A65**.

## **AGRADECIMENTOS**

Inicialmente, agradeço a Deus por ter me dado condições em todos os âmbitos de concluir o curso e o presente trabalho.

Agradeço também a minha família que me apoiou durante toda jornada, em especial meus pais e minha noiva.

Sou grato a meu orientador e ao coorientador pelo acompanhamento e apoio fornecido durante a execução desse trabalho.

## RESUMO

A automação de processos é uma tecnologia relevante e cada vez mais desejada por organizações, apesar de ser um tópico relativamente novo. Nesse sentido, surgiu o conceito de RPA – Automação Robótica de Processos, que atua não só em ambientes físicos através de robôs mecânicos, mas também em processos realizados em ambientes virtuais através de *softwares*. Este trabalho apresenta a aplicação de uma RPA em ambiente real de um *software* em um *call center* por meio da interação das tecnologias C# e Selenium. Foram realizados testes comparativos em um ambiente real entre a abordagem proposta e um operador. Resultados indicam que a RPA é vantajosa em diversos aspectos quando comparado ao ser humano, possibilitando a redução de custos e aumento da produtividade.

**Palavras-chave:** RPA – Automação Robótica de Processos; Robotização de processo; Selenium; *Call center*.

## ABSTRACT

Process automation is a relevant technology and is increasingly desired by organizations, despite being a relatively new topic. In this sense, the concept of RPA (Robotic Process Automation) emerged, which operates not only in physical environments through mechanical robots but also in processes carried out in virtual environments through software. This work presents the application of an RPA in a real *software* environment in a *call center* through the interaction of C# and Selenium technologies. Comparative tests were performed in a real environment between the proposed approach and an operator. Results indicate that RPA is superior in several aspects when compared to human beings, making it possible to reduce costs and increase productivity.

**Keywords:** Robotic Process Automation; Process robotization; Selenium; *Call center*.

## LISTA DE FIGURAS

|   |           |
|---|-----------|
| <b>Figura 1 – Exemplo de código c# utilizando selenium.....</b>         | <b>23</b> |
| Figura 2 – Fluxo de interação C# e Selenium no contexto dessa RPA. .... | 24        |
| Figura 3 – Esboço do passo a passo da modelagem e desenvolvimento. .... | 28        |
| Figura 4 – Fluxo básico do robô.....                                    | 29        |
| Figura 5 - 1º passo da rotina.....                                      | 30        |
| Figura 6 - 2º passo da rotina.....                                      | 30        |
| Figura 7 - 3º passo da rotina.....                                      | 31        |
| Figura 8 - 4º passo da rotina.....                                      | 31        |
| Figura 9 – 5º passo da rotina.....                                      | 32        |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 - Comparação de métricas da RPA com um operador ..... | 40 |
|--|----|



## LISTA DE PSEUDOCÓDIGOS

|  |           |
|--|-----------|
| Pseudocódigo 1 – referente a rotina da figura 7.....         | 33        |
| <b>Pseudocódigo 2 – referente a rotina da figura 11.....</b> | <b>34</b> |

## LISTA DE ABREVIATURAS

|      |   |                                   |
|------|---|-----------------------------------|
| API  | – | Application Programming Interface |
| HTML | – | HyperText Markup Language         |
| KPI  | – | Key Performance Indicators        |
| RPA  | – | Robotic Process Automation        |
| TI   | – | Tecnologia da Informação          |
| URL  | – | Uniform Resource Locator          |

## SUMÁRIO

|   |           |
|---|-----------|
| <b>1 INTRODUÇÃO</b> .....                               | <b>14</b> |
| 1.2 OBJETIVOS .....                                     | 16        |
| 1.3 JUSTIFICATIVA .....                                 | 16        |
| 1.4 ESTRUTURA DO TRABALHO .....                         | 16        |
| <b>2 CONCEITOS GERAIS E REVISÃO DA LITERATURA</b> ..... | <b>18</b> |
| 2.1 TRABALHOS RELACIONADOS .....                        | 18        |
| <b>3 TECNOLOGIAS</b> .....                              | <b>21</b> |
| 3.1 SELENIUM .....                                      | 22        |
| 3.2 C SHARP .....                                       | 23        |
| 3.3 KATALON RECORDER .....                              | 25        |
| 4.1 CONTEXTO .....                                      | 26        |
| 4.3 DEFINIÇÃO DE ROTINAS .....                          | 29        |
| 4.3.1 PSEUDOCÓDIGOS DAS ROTINAS .....                   | 32        |
| 4.4 DESENVOLVIMENTO .....                               | 34        |
| <b>6 CONCLUSÃO</b> .....                                | <b>42</b> |
| 6.1 LIMITAÇÕES .....                                    | 43        |
| 6.2 TRABALHOS FUTUROS .....                             | 43        |
| <b>REFERÊNCIAS</b> .....                                | <b>45</b> |

## 1 Introdução

Sabe-se que tarefas e processos repetitivos e padronizados estão presentes em diversos contextos organizacionais. Esse fato implica em alguns pontos positivos, como: facilidade de mensurar produção e estabelecer metas, padronização da produção, divisão de tarefas, identificação de gargalos, facilidade de otimização fracionada por setor da produção, dentre outros pontos positivos. No entanto, quando esses processos repetitivos são realizados apenas ou majoritariamente por pessoas, alguns pontos devem ser levados em consideração, como: lesões causadas por esforços repetitivos, alto índice de erros, uma parte da produção depende do empenho da outra, dependência de muitos colaboradores no processo, dentre outros aspectos. Logo, diante desses fatos, mecanismos devem ser adotados para que essas situações sejam contornadas.

Segundo Abbott e Bogenshneider (2018), a automação de processos teve seu início em meados do século XX no setor automotivo com objetivo de maior eficiência e qualidade do produto final. Na atualidade, esse conceito de automação está cada vez mais amplo, por exemplo, através da RPA – Automação Robótica de Processos, do inglês *Robotic Process Automation*. Dado os fatos expostos, soluções que visam automatizar processos repetitivos ou parte deles são muito úteis. De acordo com Mariano (2020), essa tecnologia é uma forma avançada de utilizar *softwares* ou *hardwares* que fazem de forma eficiente tarefas realizadas (assistida ou não) por humanos.

Nesse sentido, o presente trabalho busca abordar de forma prática e teórica uma RPA desenvolvida para um contexto de um *call center*. É importante ressaltar que esse trata de vários processos, porém, o trabalho atuou no procedimento de migração de planos de telefonia. Sendo assim, foi utilizada uma base de dados contendo milhares de informações de clientes. Essa tecnologia de robotização atua em uma plataforma Web (acessado através do navegador de um computador), a qual é operada através de processos repetitivos realizados por diversos operadores de *call center*. Para isso, foi mapeada toda a rotina e tratativas dos processos e desenvolvido um *software* (robô) que as realizam.

Ademais, vale citar que esse procedimento havia sido implementado quando o sistema funcionava apenas no navegador Internet Explorer e era utilizado a

tecnologia Autolt para a automação, que trabalha identificando quadros da tela. No entanto, com a descontinuidade do Internet Explorer surgiu a necessidade que o sistema funcionasse no navegador Google Chrome e conseqüentemente uma nova RPA precisou ser desenvolvida. Visto isso, foi elaborada a atual abordagem, sendo que essa não utiliza o Autolt, e sim o Selenium, que funciona identificando os elementos HTML do sistema web.

Por fim, além das abordagens do desenvolvimento da RPA, serão demonstrados os resultados do funcionamento do robô em comparação com as atividades realizadas por pessoas, levantando-se em consideração as métricas de ambos agentes.

### **1.1 Definição do problema**

Em ambientes de *call center* é comum a situação de procedimentos que envolvam um alto número de colaboradores realizando tarefas repetitivas. Esse fato levanta algumas implicações, como por exemplo, custos altos, danos à saúde do operador, alto índice de erros, morosidade nos processos, dentre outras questões.

Nesse sentido, foi identificado em um determinado *call center* um processo que trata de uma rotina padrão e repetitiva envolvendo um grande volume de dados de clientes. Essas informações possuem diversos atributos, como: RG, CPF, nome, telefone, *status* no sistema, plano de telefonia, dentre outros. Esses dados interagem com um sistema web através da inserção manual realizada por um operador de *call center* e, após as interações, é retornada uma tratativa específica para cada cliente. A tarefa principal do colaborador desse *call center* ao usar o sistema web é migrar um determinado plano de telefonia celular para outro plano. Para executar essa ação, o operador do *call center* precisa preencher vários campos em diversas telas do sistema web. Todos os preenchimentos desse sistema são realizados por um humano, o que torna o procedimento extremamente lento e suscetível a erros, além de prejudicar a saúde física e psicológica do indivíduo, visto que ele realizará atividades repetitivas rotineiramente.

Dessa forma, esse trabalho visa atuar nesse problema através da construção de um robô baseado no conceito de RPA, que além de realizar tarefas de preenchimento dos campos, consegue fazer as tratativas necessárias para cada cliente.

## 1.2 Objetivos

O presente trabalho tem como objetivo demonstrar o desenvolvimento de um robô que irá atuar em um sistema de um grande *call center*. Para isso, serão abordadas as tecnologias utilizadas no projeto e os respectivos papéis destas. Posteriormente, será apresentado como foi o desenvolvimento da RPA, abordando assuntos como, a forma de captura dos dados, a modelagem da solução, a definição e mapeamento de rotinas, o fluxo completo da RPA, dentre outros aspectos. Por fim, serão demonstrados os resultados obtidos com a implementação do robô.

## 1.3 Justificativa

Sabe-se que em um contexto de globalização a demanda por processos cada vez mais eficientes é grande. Nesse sentido, a RPA atua muito bem, visto que ela tem como objetivo automatizar processos de determinada organização. Fato esse, que se bem implementada, aumenta demasiadamente a produtividade. Dessa forma, abordando o conceito de RPA, é possível tratar, além de conceitos teóricos, sua aplicação prática. Partindo da identificação de um processo repetitivo e manual, podemos fazer o mapeamento desse processo, o desenvolvimento do robô e a coleta de resultados. Logo, esse trabalho visa abordar todas essas fases de desenvolvimento do *software* (robô) em um contexto de *call center*.

Outro fator que se destaca, é a questão da RPA ser uma tecnologia relativamente nova e não existir muitos trabalhos acadêmicos relacionados (FIGUEIREDO, 2019), principalmente trabalhos que abordam desde a parte teórica até a prática de forma detalhada. Sendo assim, esse trabalho tenta ocupar essa lacuna, visto que aborda os conceitos teóricos da RPA e traz todas etapas do desenvolvimento dessa em um ambiente real de *call center*.

## 1.4 Estrutura do trabalho

Esse trabalho foi estruturado em 6 capítulos. Inicialmente, na introdução, é iniciada a abordagem do tema e a contextualização do mesmo. Logo em seguida, no capítulo 2, foram demonstrados os conceitos gerais da RPA e foram abordados diferentes autores. Ainda nesse capítulo serão tratados também trabalhos que

possuem relação com esse trabalho. Já no capítulo 3 são expostas as tecnologias utilizadas no projeto, bem como seus papéis no contexto desse projeto. No capítulo 4 é apresentada a abordagem proposta para o projeto. No capítulo 5 foram demonstrados os resultados obtidos por essa pesquisa. Por fim, no capítulo 6 foram abordadas as conclusões.

## 2 Conceitos gerais e revisão da literatura

A automação de processos possui algumas ramificações e categorias, entre essas existe a Automação Robótica de Processos, do inglês *Robotic Process Automation* (RPA). Ela consiste em um conjunto de passos que determinam uma regra para um “robô” realizar (CASEY, 2019).

Para Marques (2002), robotização refere-se a um conjunto de atividades mecanizadas, que possibilitam uma maior produção de unidades com uma qualidade superior e redução de custos. No entanto, quando é citado o termo robô é remetida uma visão de máquinas físicas que automatizam determinados processos. Conquanto, vale ressaltar que a RPA não se refere necessariamente a um robô físico, mas também em uma aplicação de *software* programada para executar rotinas padrões de determinados processos repetitivos realizados por seres humanos, conforme Lacity et al. (2015)

É importante ressaltar que a automatização de rotinas via *software* não é um recurso novo, no entanto o interesse de organizações em deixar seus processos repetitivos automatizados via *software* tem crescido com surgimento de tecnologias como a RPA (LACITY et al., 2015). Para Mager (2019), tecnologias de RPA estão sendo, cada vez mais, adotadas para otimização de processos de forma que permita uma redução de custos, maior eficiência e redução de erros.

Kirchmer e Franz (2019) abordam a RPA como um subconjunto de recursos de *software* capaz de interagir em interfaces de usuários de outros sistemas de *software*, imitando a rotina que um ser humano adotaria. Para Santiago e Rodriguez (2017), *softwares* de RPA automatizam processos com base em regras, envolvendo tarefas de rotinas, resultados determinísticos, dados estruturados e também permite a transferência de dados de várias fontes de entrada, como *e-mail* e planilhas.

### 2.1 Trabalhos Relacionados

Em Sherman e Rodrigues (2018) foi elaborada uma pesquisa descritiva para apresentar como inovações tecnológicas podem influenciar nos processos que envolvem a emissão de notas fiscais em uma organização da área metalúrgica. Para atingir esse fim, os autores inicialmente verificaram a expectativa dos responsáveis pela RPA e como era o processo de emissão de notas antes da implementação dessa.



Logo em seguida, fizeram uma análise de negócio para implementação dessa tecnologia. E, por fim, levantaram os impactos positivos e negativos dessa aplicação.

No estudo publicado por Devarajan (2018) é apresentado os conceitos de RPA e seus respectivos benefícios. Um fator interessante do texto desse autor é a questão de reforçar a importância da aplicação de tecnologias de RPA em diferentes setores organizacionais como: bancário, auditorias, companhias de seguros, varejo e fábricas. Além de demonstrar a relevância da RPA nesses respectivos segmentos, é apresentado também em quais processos dessas organizações a RPA teria uma atuação benéfica.

A pesquisa de Mariano (2020) objetivou identificar como a governança de RPA pode ser estruturada em grandes organizações, para que a implementação de tecnologias de RPA seja eficiente. Foram apresentados os conceitos de RPA e seus benefícios. Além disto, o autor defendeu que o desenvolvimento da RPA é relativamente simples para a TI. No entanto, o desafio está na estruturação da RPA para o setor de governança. Para atender esses objetivos, o autor apresentou métricas de avaliação de uma RPA, como exemplos de mecanismos que a governança de TI pode adotar responsabilidades dentro da governança de RPA, ou seja, foram abordadas questões que são recomendadas a serem levadas em consideração pelo setor de governança de RPA.

Na monografia escrita por Girão (2021) foi apresentado o conceito de RPA baseado em algumas perspectivas e também uma visão estratégica da vantagem da adoção de RPA em organizações de diferentes setores. Todavia, sua pesquisa teve um foco maior em evidenciar as vantagens da RPA, por exemplo, como funcionam seus processos, demonstrar casos de usos e relatar de maneira mais detalhada a implementação dessa tecnologia em uma organização da área de *Customer Response* (Resposta ao Cliente).

Júnior (2021) visou avaliar a implementação de uma RPA em um processo de cadastro de usuários em determinado sistema de viagens e no processo de lançamento de notas de serviços de fretes. Nesse trabalho, foram realizados alguns passos. Primeiramente foi feito o entendimento do processo de gestão e controle do setor responsável pela RPA. Em seguida foram analisados de maneira isolada ambos os processos e, por último, foram demonstrados os resultados alcançados com a implementação da RPA. Nesses resultados ficou constatado que a RPA é bastante

benéfica, já que se tratava de processos com grande volume e com muita repetição, o que possibilitou a RPA atuar reduzindo custos e aumentando a produtividade.

Barreto (2022) elaborou em sua monografia uma RPA utilizando o Selenium como tecnologia. O contexto de atuação desse autor se deu em um procedimento que busca levantar para organizações de contabilidade as empresas que possuem pendências fiscais no Estado do Rio Grande do Norte. Por fim, como resultado de seu trabalho, foi constatado que a RPA deixou todo o processo mais célere, uma vez que possibilitou uma redução de 36% do tempo do procedimento, acarretou também em uma redução de custos anual considerando um operador humano de R\$ 1.386,00 e ainda obteve uma redução de erros no procedimento.

### 3 Tecnologias

A Automação Robótica de Processos depende da integração de tecnologias, principalmente quando essa atua em ambiente de sistemas computacionais. É relevante citar que para o desenvolvimento de RPAs sem utilizar plataformas já construídas é comum o uso de tecnologias como Python, Java, Javascript, C#, Autolt e Selenium, visto que essas tecnologias possuem algumas características, como:

- . Python: possui uma sintaxe simples e boa legibilidade, contém muitas bibliotecas, muita versatilidade, visto que não depende de uma plataforma específica.
- . Java: é uma linguagem derivada do C e C++ que roda em qualquer plataforma, uma vez que funciona utilizando uma máquina virtual que utiliza o hardware para rodar o programa. É uma linguagem utilizada em projetos maiores, visto que tem como uma de suas maiores características a programação orientada a objetos.
- . JavaScript: uma linguagem que apresenta boa performance e várias bibliotecas, ademais, é uma linguagem que tem aplicação em diversos contextos diferentes.
- . C#: linguagem utilizada em grandes projetos, sendo a principal da plataforma .NET. Suporta diversos paradigmas de programação, como a programação orientada a objetos. Ademais, é uma linguagem que conta com inúmeros recursos e apresenta boa compatibilidade com diversas ferramentas, assim, possibilitando uma boa produtividade.
- . Autolt: é uma tecnologia que visa automatizar a GUI do Windows. Ela funciona fazendo uma combinação de teclas, movimentação do mouse e clicks, se orientando através de frames da tela.
- . Selenium: tecnologia muito utilizada em testes unitários, porém, seu recurso de identificar elementos HTML em uma página *web* faz com que ela seja amplamente utilizada em automações.

Para o presente trabalho foram empregadas algumas tecnologias que trabalham em conjunto como: Selenium que atua como um framework capaz de

interagir com os elementos de páginas HTML; a linguagem de programação Microsoft C#, que contém o código principal do robô; e o Katalon entrando como uma tecnologia para acelerar determinadas etapas do desenvolvimento da RPA.

### 3.1 Selenium

O Selenium é um projeto que abrange uma variedade de ferramentas e bibliotecas que permitem e suportam a automação de navegadores da web (Selenium, 2022). O funcionamento da tecnologia é relativamente simples. Em suma, o *framework* integra com o *driver* do navegador (componente que possui um conjunto de métodos que permitem uma interação com o navegador) e fornece um conjunto de ferramentas que permitem interagir com elementos HTML de diferentes sites. Por conta desses quesitos, essa ferramenta é utilizada também para testes de aplicação *web*. Esses fatos estão exemplificados no seguinte trecho de código da [Figura 1](#), disponível na documentação do Selenium.

Outro ponto que vale ser ressaltado dessa tecnologia é o fato dela ser toda disponível gratuitamente e possuir seu código fonte aberto. Em outras palavras, além da possibilidade de melhor entendimento do framework, é possível contribuir com melhorias para comunidade. Ademais, o Selenium possui integração simples com os principais navegadores, fato esse que permite uma maior flexibilidade ao desenvolvedor.

```
using OpenQA.Selenium.Chrome;

namespace SeleniumDocs.Hello
{
    public class HelloSelenium
    {
        public static void Main()
        {
            var driver = new ChromeDriver();

            driver.Navigate().GoToUrl("https://selenium.dev");

            driver.Quit();
        }
    }
}
```

**Figura 1 – Exemplo de código c# utilizando Selenium.**

Fonte: <https://www.selenium.dev/documentation/>

O exemplo da [Figura 1](#) acima está presente na documentação do Selenium. O código acima está implementado na linguagem C# e, quando executado, irá acessar a URL referenciada no exemplo.

Dado os fatos expostos, essa tecnologia foi utilizada nesse projeto, uma vez que apresenta um grande volume de ferramentas de interação com elementos HTML, tem boa interação com a linguagem de programação C# e possui uma curva de aprendizado relativamente pequena.

### 3.2 C Sharp

Segundo a documentação da Microsoft (2022), a linguagem de programação C# é orientada a objetos e a componentes. Essa fornece construções para dar suporte diretamente a esses conceitos, tornando o C# uma linguagem natural para criação e uso de componentes de *software*. Ademais, é importante citar que a C# surgiu da conceituada linguagem de programação C. Além disso, nessa tecnologia são frequentemente adicionados recursos para dar suporte a novas demandas de trabalho. Esse fato à torna muito utilizada por desenvolvedores de diferentes seguimentos. Outro fato relevante da linguagem é que os *softwares* desenvolvidos por ela são

executados na arquitetura .NET . Todas essas questões estão dispostas na documentação da linguagem em Microsoft (2022).

A linguagem C# foi a principal utilizada nesse trabalho. Ela teve como função principal a integração com uma API (*Application Programming Interface*) que fornece dados de clientes. A API foi desenvolvida pelo cliente e ela é consumida para pegar os dados do cliente que terão o plano de telefonia migrado para outro. Além disso, a linguagem C# foi responsável por eventuais tratativas necessárias, sejam elas inerentes ao sistema ou ao próprio robô. Tratativa pode ser explicada como comportamentos que o robô precisa ter em determinadas situações. Um exemplo que podemos citar é quando um cliente tem restrição financeira (está em débito com a operadora de celular). Se ele tiver em débito, o robô irá salvar um status específico para esse cliente e, em seguida, buscará um próximo cliente para tratar. A relação do C# com o Selenium nesse projeto está exemplificada na [Figura 2](#).

Vale ressaltar que sua escolha para esse projeto se deu pelos seguintes fatos: familiaridade da equipe com a linguagem, RPA executada em sistemas operacionais da Microsoft e boa compatibilidade com o Selenium e *drivers* de navegadores.

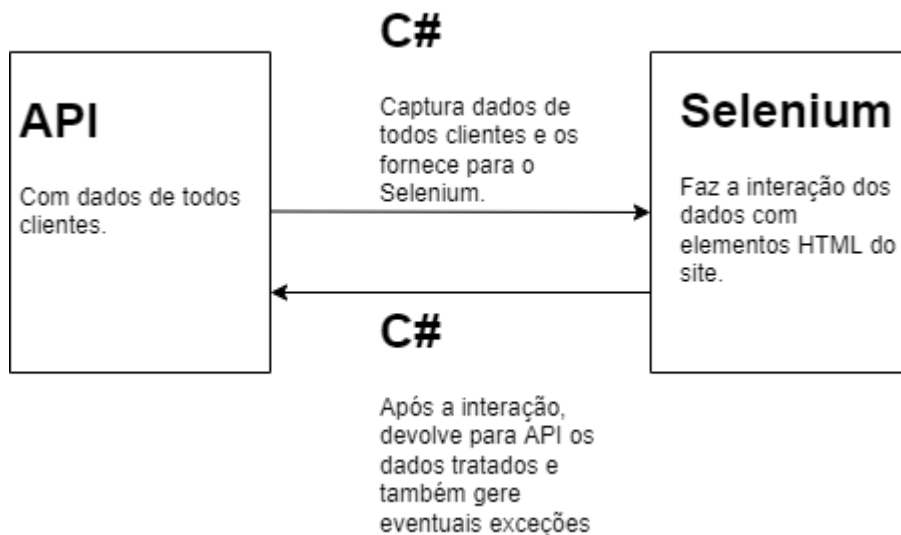


Figura 2 – Fluxo de interação C# e Selenium no contexto dessa RPA.

Fonte: Autoria própria

O funcionamento desse fluxo presente na Figura 2 é esquematizado da seguinte maneira:

1. Através da linguagem de programação C#, um método busca os dados fornecidos pela API;
2. Com esses dados, o Selenium faz suas interações com os elementos HTML da página;
3. À medida que essas interações ocorrem, o C# faz eventuais tratativas e devolve o resultado tratado para a base de dados.

### 3.3 Katalon Recorder

O Katalon Recorder é um *software* disponível como uma extensão do navegador Google Chrome. Sua função é basicamente gerar um código em Selenium com base em uma rotina adotada pelo usuário em um *site*. Em suma, seu funcionamento é bem simples, o usuário realiza uma rotina (navega) em determinado *website* e o Katalon consegue identificar quais elementos HTML o usuário interagiu e também como foi essa interação. Como exemplo podemos citar se foi algum clique em botão, inserção de dados em campo, dentre outras situações. Após a interação do usuário, o Katalon gera toda a rotina em código em Selenium.

Apesar do Katalon ser um facilitador para utilizar o Selenium, principalmente para reduzir a curva de aprendizado, quando se insere rotinas muito grandes ou complexas ele não se torna interessante. Isso acontece porque ele acaba gerando rotinas que apresentam muitos erros. Outro fator a ser destacado é quando se tenta gerar rotinas via Katalon em *sítes* que possuem mecanismos de segurança ou nome de campos HTML dinâmicos. Nessas situações o *software* não consegue cumprir seus objetivos, visto que ele captura dados dispostos na página HTML no momento da interação, ou seja, ao serem recarregados os atributos daquela página irão mudar a cada novo carregamento da página. Logo, diante dessa situação, o próprio desenvolvedor precisa programar as rotinas em Selenium manualmente contornando esses impasses.

Nesse sentido, essa tecnologia foi utilizada no início do desenvolvimento e em etapas mais simples, já que com ela é possível gerar rotinas utilizando o Selenium mais rapidamente.

## 4 Abordagem Proposta

Neste capítulo será apresentado como a RPA especificada nesse trabalho foi desenvolvida, passando pelo contexto de atuação da RPA, como esse foi modelado para a implementação, a definição das rotinas adotadas no sistema a serem automatizados e as metodologias de desenvolvimento utilizadas.

### 4.1 Contexto

Em um determinado *call center* existe um processo que consiste basicamente na migração de clientes para diferentes planos de telefonia. Para que isso aconteça é necessária a inserção de diversos dados em determinado sistema web. Os operadores de *call center* preenchem vários campos nas telas do sistema. Para cada cliente, é necessário realizar tratativas específicas. O objetivo final é conseguir migrar o plano do cliente com sucesso. Entretanto, como mencionado acima, e como, por exemplo, um cliente pode ter restrição financeira e não será possível migrar seu plano. A inserção desses dados, antes da implementação da RPA, era toda realizada por muitos operadores (funcionários do *call center*), fato esse que gerava muitos movimentos repetitivos, deixava todo processo extremamente suscetível a erros e com muita morosidade.

Diante desse cenário, foi mapeado todo o fluxo do operador no sistema, inclusive as tratativas. Em seguida foi desenvolvida uma RPA que busca os dados de cada cliente via API e navega pelo sistema automaticamente realizando as inserções de dados e tratativas necessárias. O objetivo principal da RPA é realizar a migração do plano do cliente sem a intervenção do operador de *call center*, ou seja, o robô irá substituir a mão de obra humana.

Vale ressaltar que todas as etapas abordadas nessa seção foram desenvolvidas por uma equipe de uma determinada fábrica de *software*, as quais o autor desta monografia atuou ativamente em todas. Igualmente, é importante citar que



a referida atuação se deu principalmente no desenvolvimento dos códigos fontes das rotinas apresentadas em “4.3 Definição de rotinas”.

## 4.2 Modelagem da solução

Para Booch et al. (2005) existem quatro objetivos majoritários para modelagem de *software*, sendo eles: permitir uma visualização do sistema, como ele é ou como desejamos que ele seja; especificar a estrutura ou comportamento do sistema; oferecer um guia para construção do sistema; e facilitar a visualização do histórico das decisões tomadas no projeto. Posto isto, para desenvolver a RPA, foram seguidos os seguintes passos:

**1º passo:** Identificar a forma que os dados dos clientes estavam dispostos para serem consumidos. Nesse contexto, existe uma API que fornece dados de todos os clientes.

**2º passo:** Mapear toda a rotina de um operador no sistema, inclusive eventuais tratativas, como será apresentado no tópico “4.3 Definição de Rotinas”.

**3º passo:** Com os dados e rotinas mapeados, foi feita a definição das tecnologias a serem utilizadas. Para esse projeto e como falado anteriormente, foi utilizado o *framework* Selenium como ente interativo nos elementos HTML da página e a linguagem de programação C# como responsável pelas tratativas.

**4º passo:** Foi identificada cada rotina de maneira isolada e desenvolvida a rotina no RPA para aquele contexto, ou seja, foi fracionado rotina a rotina e feito o desenvolvimento do código fonte.

**5º passo:** Observação e teste de todas as rotinas.

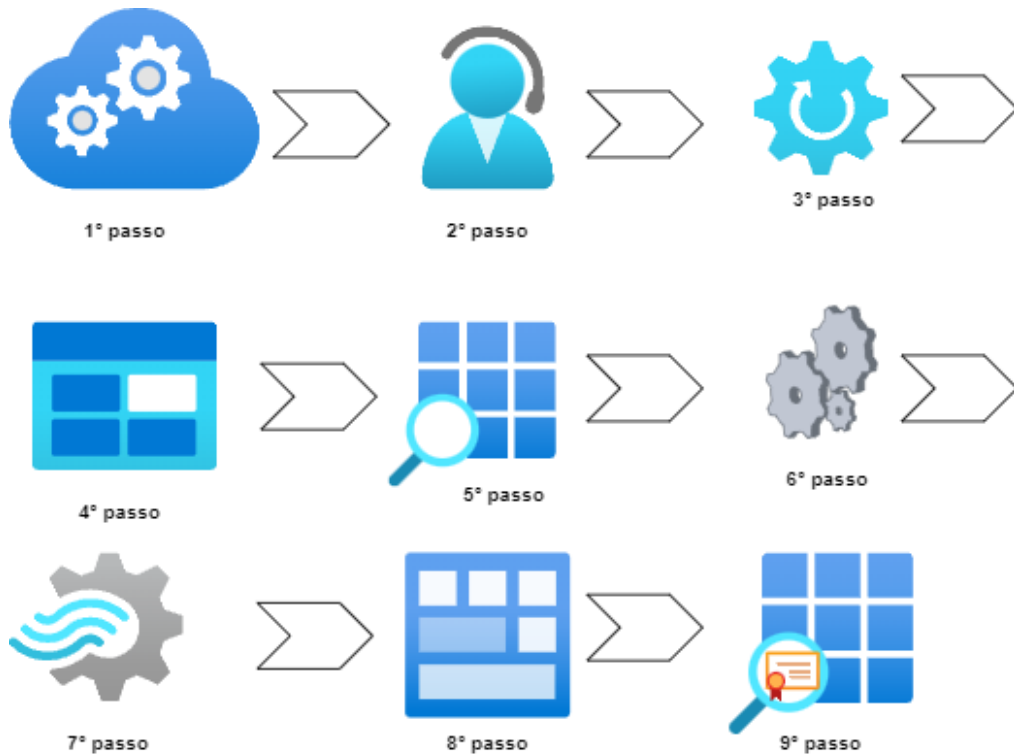
**6º passo:** Ajuste de rotinas incorretas.

**7º passo:** Ajuste de gargalos do próprio sistema web que impactavam o bom funcionamento da RPA. Em determinados momentos, o sistema web fica lento e o robô precisa ter comportamentos específicos quando isso ocorre. Geralmente, quando

o sistema web apresenta lentidão, o robô realiza alguns minutos em pausa e tenta retomar a execução em seguida.

**8º passo:** Criação de uma interface simples para iniciar a operação da RPA e registrar todos seus eventos.

**9º passo:** Observação da RPA em produção.



**Figura 3 – Esboço do passo a passo da modelagem e desenvolvimento.**

Fonte: Autoria própria

Sendo assim, após a conclusão dessas etapas, o ciclo básico da RPA se deu da seguinte forma Figura 4.

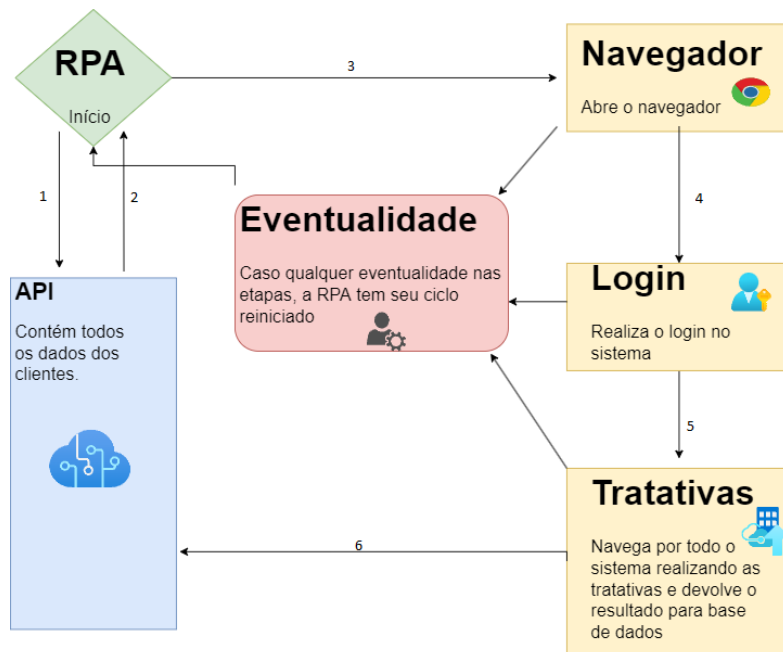


Figura 4 –

#### Fluxo básico do robô

Fonte: Autoria própria

O fluxo apresentado na Figura 4 tem o seguinte funcionamento: primeiramente é dado o início na RPA e, nesse momento, ela captura o dado do primeiro cliente na fila (seta 1); posteriormente ela inicia o navegador (seta 2) e realiza o login com acesso de um operador de *call center* (seta 3); logo em seguida os dados dos clientes são inseridos nas páginas do sistema web e dá início às tratativas (seta 4); finalmente, com essas concluídas, o resultado é devolvido à base de dados via API do cliente (seta 6).

É importante ressaltar que o robô apresenta alta estabilidade. Salvo raras exceções, ele tem sua atividade paralisada. Mas, se quaisquer eventualidades acontecer, ele será capaz de reiniciar até retomar seu fluxo normal.

#### 4.3 Definição de rotinas

Nesse tópico será apresentado parte do mapeamento do fluxo do sistema, preservando os dados da organização de maneira ética e seguindo diretrizes da LGPD (Lei Geral de Proteção de Dados Pessoais). Posteriormente, serão demonstrados pseudocódigos para algumas telas.

1º passo: a RPA abre o navegador e realiza o login com as credenciais de um operador do *call center* [Figura 5]. Nesse ponto a RPA atua inserindo as credenciais do operador no sistema web e clica no botão “Acessar”. Vale ressaltar que, caso a RPA encontre alguma eventualidade em seu ciclo, ela reiniciará para essa tela.

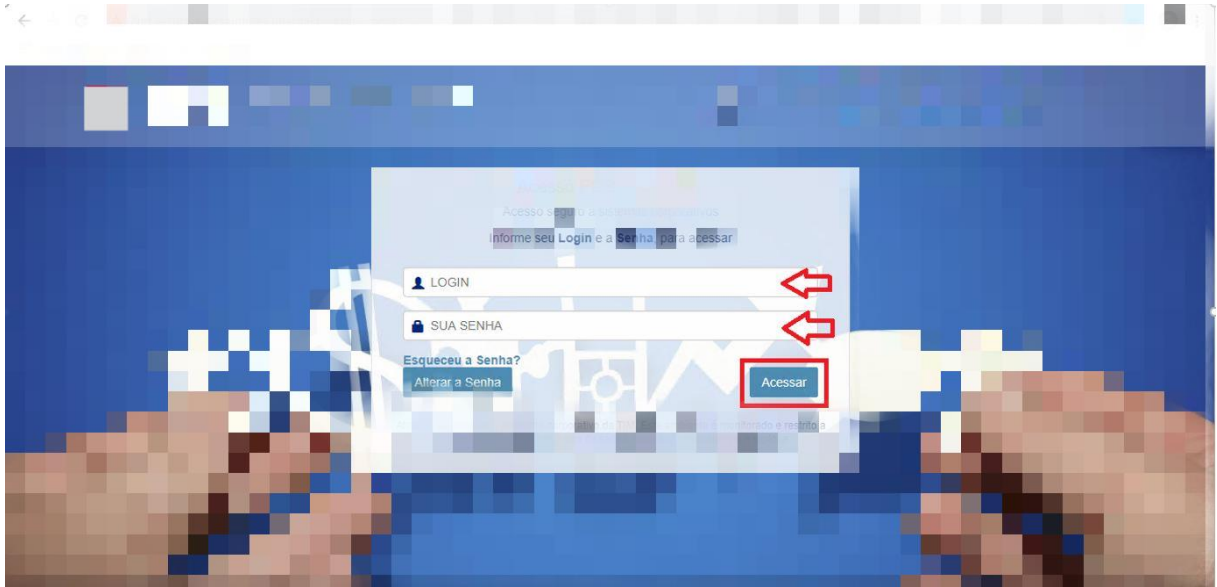


Figura 5 - 1º passo da rotina (print do sistema)

2º passo: existe o direcionamento para determinado botão [Figura 6]. Nesse momento a RPA simplesmente clica no botão “Acessos”.

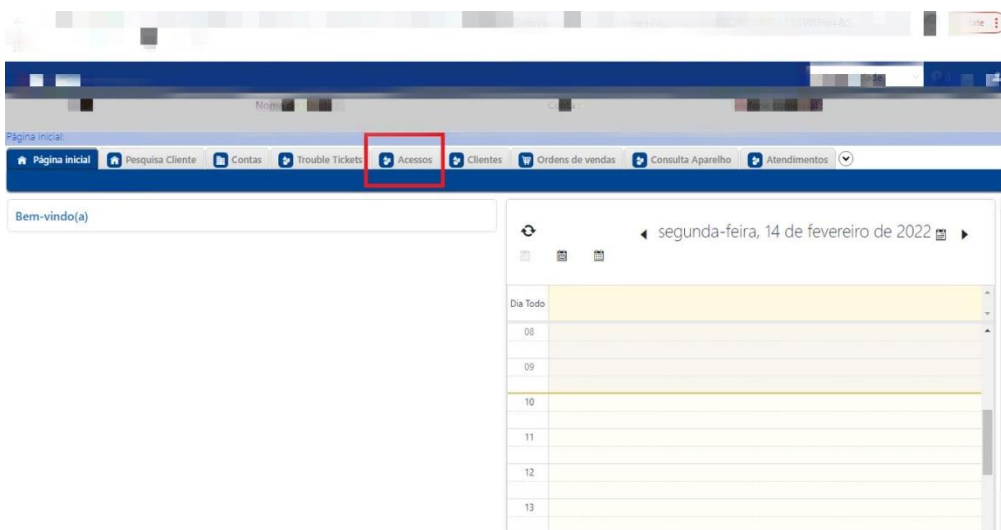


Figura 6 - 2º passo da rotina (print do sistema)

3º passo: com a aba “Acessos” selecionada, é aberta uma nova tela que deve ser direcionada ao botão “Pesquisar Acessos” [Figura 7]. A RPA identifica a nova aba e direciona para o passo seguinte.

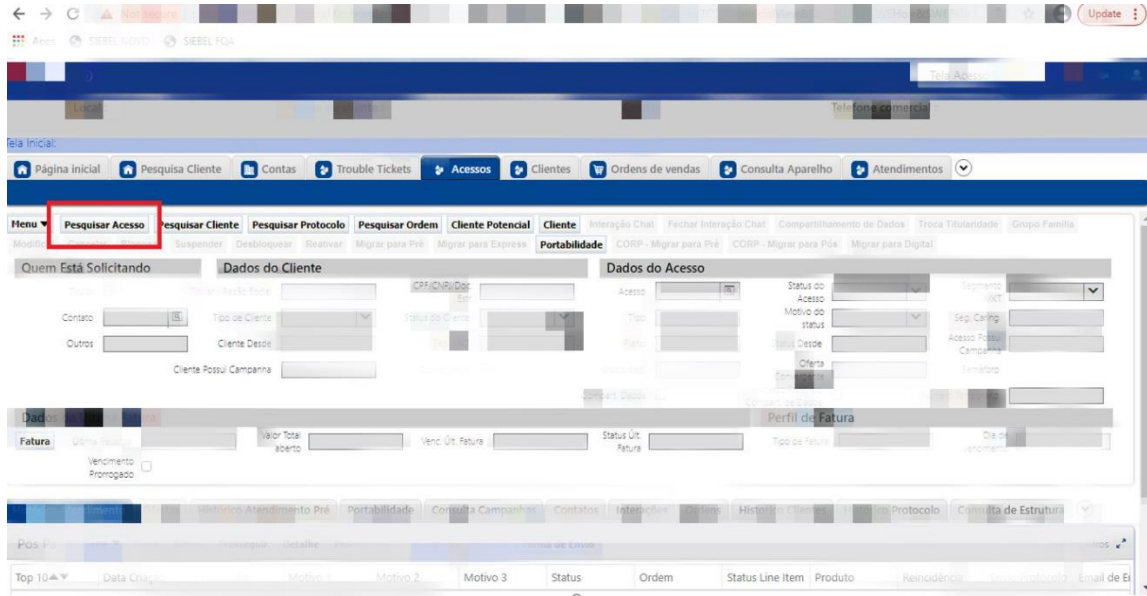


Figura 7 - 3º passo da rotina (print do sistema)

4º passo: com a aba “Pesquisar Acesso” aberta, será injetado um número de telefone do cliente e posteriormente feito o clique no botão “Ir” [Figura 8]. No presente passo pode-se destacar que a RPA apresenta uma grande vantagem, visto que ela injeta rapidamente o número de telefone sem possibilidade de erros de digitação.

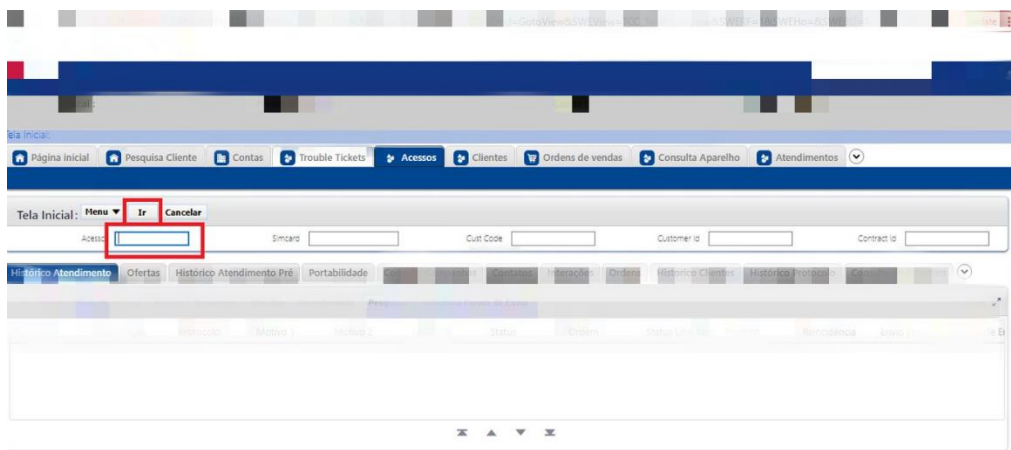


Figura 8 - 4º passo da rotina (print do sistema)

5º passo: deve-se verificar se o cliente possui *status* ativo e se ele possui CPF registrado [Figura 9]. Nessa etapa também é possível citar que a RPA apresenta uma grande vantagem, uma vez que essa faz a rápida verificação dos campos assinalados.

The screenshot displays a web application interface with a navigation menu at the top. The main content area is divided into two sections: 'Dados do Cliente' and 'Dados do Acesso'. In the 'Dados do Cliente' section, the 'Status do Cliente' dropdown is set to 'Ativo' and the 'CPF/CNPJ/Doc. Estr.' field contains the value '36'. In the 'Dados do Acesso' section, the 'Status do Acesso' dropdown is also set to 'Ativo'. Both of these dropdown menus are highlighted with red rectangular boxes.

**Figura 9 – 5º passo da rotina (print do sistema)**

Vale ressaltar, que existem cerca de 30 rotinas, porém, todas seguem essa lógica de inserir dados, captar dados e fazer eventuais tratativas até chegar no ciclo final. Logo, não é necessário demonstrar todas as rotinas para o entendimento do contexto. Como dito anteriormente, a tratativa com sucesso se dá quando o cliente tem o plano de telefonia migrado corretamente.

#### 4.3.1 Pseudocódigos das rotinas

Nessa seção serão demonstrados pseudocódigos de algumas rotinas apresentadas anteriormente para um melhor esclarecimento do funcionamento. Abaixo temos o pseudocódigo da Figura 5.

### **Pseudocódigo 1: Tela de login**

---

```
1 public void Login(Operador x){
2     driver.Navigate().GoToUrl("URLDESEJADA");
3     driver.FindElement(By.Id("UsuarioLogin")).Click();
4     driver.FindElement(By.Id("UsuarioLogin")).Clear();
5     driver.FindElement(By.Id("UsuarioLogin")).SendKeys("x.login");
6     driver.FindElement(By.Id("senhaLogin")).Click();
7     driver.FindElement(By.Id("senhaLogin")).Clear();
8     driver.FindElement(By.Id("senhaLogin")).SendKeys("x.senha");
9     driver.FindElement(By.XPath("//button[@type='submit']")).Click();
11 }
```

#### **Pseudocódigo 1 – referente a rotina da figura 7**

Nesse [Pseudocódigo 1](#) o funcionamento se dá seguinte maneira:

- Linha 2 é dado o direcionamento para URL desejada;
- Linha 3 é feito o clique no campo “UsuarioLogin”;
- Linha 4 o Selenium faz uma limpeza do campo para evitar inconsistências;
- Linha 5 é inserido o login do usuário.
- Da linha 6 a 8 é a mesma lógica anterior, porém, com a senha do usuário;
- Linha 9 é feito o clique no botão para acessar a página.

Agora, logo abaixo, segue o pseudocódigo referente à Figura 9:

## Pseudocódigo 2: Referente a “Rotina 11”

---

```
1 public boolean verificaAtivo(){
2     Thread.Sleep(5000);
3     if(driver.FindElement(By.Id("Ativo")).text() == "Ativo"
4     && driver.FindElement(By.Id("Cpf")).text() != " ") {
5         return true;}
6     else{return false;}
7 }
```

### Pseudocódigo 2 – referente a rotina da figura 11

No [Pseudocódigo 2](#), a linha 2 do método inicia com um “Thread.Sleep” pois é necessário que se garanta que os elementos da página HTML estejam carregados para o robô começar a operar na página. Logo em seguida, da linha 3 a 4, é feita uma verificação em dois campos, com as condições do usuário estar ativo e o campo CPF preenchido. E, se esses casos forem atendidos, o método retorna verdadeiro.

Vale ressaltar que existem maneiras e lógicas diferentes para esses exemplos de pseudocódigos, mas ambos exemplos buscam uma visão didática da situação.

## 4.4 Desenvolvimento

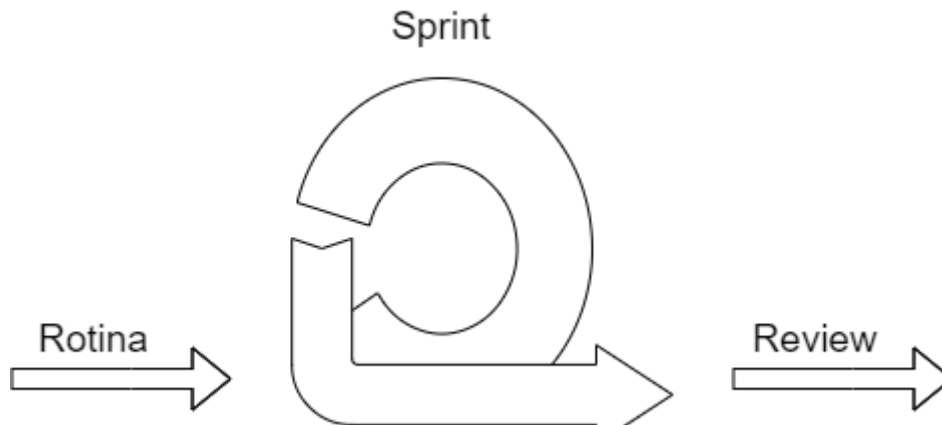
Apesar dessa RPA ter resquícios de desenvolvimento em cascata (dividido em diversas etapas sequenciadas para o desenvolvimento do *software*), durante sua construção, os gestores e equipe adotaram metodologias ágeis em paralelo.

Durante o processo de desenvolvimento, foi adotada a metodologia de desenvolvimento ágil Scrum. Segundo Soares (2004), seu objetivo é apresentar um processo conveniente para projeto. A Scrum tem uma abordagem empírica que aplica algumas ideias da teoria de controle de processos industriais para o desenvolvimento de *softwares*, reintroduzindo as ideias de flexibilidade, adaptabilidade e produtividade. Nesse sentido, a metodologia Scrum possui algumas métricas que precisam ser definidas.

Nesse projeto foi seguida uma linha que melhor se adequa ao contexto supracitado. Sendo assim, essa metodologia foi aplicada da seguinte maneira: sprints



de desenvolvimento não eram determinados por tempo ou qualquer outra métrica padrão; foram definidas *sprints* por cada tela da rotina, ou seja, o *Sprint Backlog* era parametrizado por cada rotina e, ao final do desenvolvimento daquela, era feita a *Sprint Review Meeting* na qual era apresentado as funcionalidades desenvolvidas (Figura 10). Vale ressaltar que também foi adotada a *Sprint Retrospective*, na qual cada membro da equipe expunha seus aprendizados e dificuldades.



**Figura 10 – Metodologia Scrum**

Fonte: Autoria própria

Foi adotada também a metodologia Kanban, já que essa permite que os gestores tenham controle de todo fluxo de desenvolvimento, quais rotinas já foram desenvolvidas, quais estão em desenvolvimento e quais estão pendentes. Tal metodologia possibilita ainda o conhecimento de qual membro ou equipe responsável por aquela rotina em específico (Figura 11). Segundo Oliveira (2020), o Kanban, aplicado em empresas de *software*, fornece uma melhor visualização do fluxo de trabalho, possibilitando assim melhorias nas etapas do desenvolvimento e conseqüentemente no *software*.

| PENDENTES | DESENVOLVIMENTO                      | Concluídas                           |
|-----------|--------------------------------------|--------------------------------------|
| Rotina a  | <b>Rotina "d"</b><br>Equipe 1        | <b>Rotina "g"</b><br>Equipe 1        |
| Rotina b  |                                      |                                      |
| Rotina c  | <b>Rotina "e"</b><br>Equipe 2        | <b>Rotina "h"</b><br>Equipe 2        |
|           | <b>Rotina "f"</b><br>Desenvolvedor 1 | <b>Rotina "i"</b><br>Desenvolvedor 1 |

**Figura 11 – Kanban aplicado ao projeto**

Fonte: Autoria própria

Outro método adotado em determinadas rotinas mais complexas foi a programação em pares (quando dois desenvolvedores atuam em um mesmo projeto de maneira simultânea, em suma, enquanto um desenvolve o outro observa e apoia), visto que algumas rotinas apresentavam determinado nível de dificuldade. Sendo assim, eram escalados mais desenvolvedores para trabalhar nessa de maneira simultânea.

É importante ressaltar que essas metodologias adaptadas para o contexto não foram desenvolvidas isoladamente pelo autor desse trabalho, mas sim pela equipe de desenvolvimento a qual o autor pertence e seus respectivos gestores. Sendo assim, o objetivo foi documentar e exemplificar como essas metodologias foram aplicadas no projeto.

## 5 Apresentação e análise dos Resultados

Os resultados coletados desse projeto levaram em consideração o tempo médio de trabalho de um operador de *call center* e o tempo médio que esse leva para completar um ciclo completo no sistema (fazer a migração do plano do cliente até a última tela do sistema web). Logo, possuímos as seguintes médias:

- Operador: Em média 4 minutos para concluir um ciclo e 15 operações (migrações de plano) por hora. Considerando 6 horas líquidas de trabalho por dia, o operador realiza 90 migrações por dia de trabalho.
- Robô: O robô segue a mesma média de tempo (4 minutos) para concluir um ciclo, dado as limitações/lentidões do sistema web. No entanto ele trabalha 24 horas por dia, concluindo em média 360 operações por dia.

Vale ressaltar que não existem métricas precisas em relação a erros dos operadores nesse *call center* em específico. Conquanto, Sérvio (2022) mostra que 64% dos operadores de *call center* dizem se distrair em diversos momentos ao coletar dados dos clientes. Outros 37% dos operadores citaram que esse fato acontece frequentemente, causando assim, muitos erros no cadastro manual de dados. Em contrapartida e para resolver esse problema, o robô dificilmente apresenta erros na execução das suas rotinas. Ele apenas está exposto a possíveis instabilidades de conexão ou erros dos sistemas que eles manipulam. Porém, caso essas eventualidades externas aconteçam, o robô deve possuir mecanismos que permitem contorná-las, como demonstrado na Figura 4. Adicionalmente, quaisquer erros registrados pelo robô são facilmente mapeados e implementado novos métodos na programação para lidar essa nossa situação.

Uma vez que o investimento é um fator relevante para implantação de qualquer tecnologia, é importante levantar o custo médio do desenvolvimento de uma RPA e fazer a comparação desse com o custo de um operador de *call center*. A RPA relatada nesse trabalho levou aproximadamente 200 horas para ser desenvolvida, e com 3 analistas/desenvolvedores alocados para tal. Assim, considerando-se a média do

valor/hora de um desenvolvedor<sup>1</sup> de R\$ 54,00, esta solução tem um custo de desenvolvimento total aproximado de R\$ 10.800. Vale ressaltar que não estão incluídos custos para manutenção da RPA. Por outro lado, um operador de *call center* tem o salário médio de R\$ 1.213,46 por mês<sup>2</sup>, considerando 6 horas líquidas trabalhadas. É importante mencionar que neste valor não estão incluídos despesas com transporte, alimentação e demais encargos trabalhistas como: férias, 13º salário, FGTS, dentre outros.

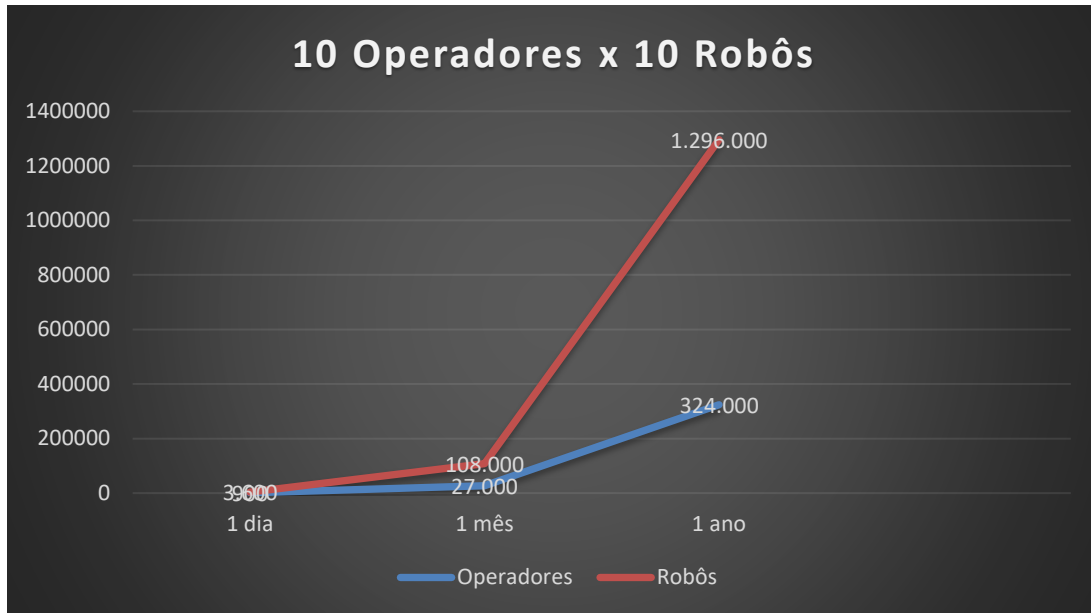
Para fins de comparação, antes do início do robô, o *call center em questão* tinha 10 operadores trabalhando 6 horas líquidas por dia. O salário de cada funcionário era de R\$ 1.600,00 por mês, sem levar em consideração os encargos e demais impostos. Assim, o custo mensal dos 10 funcionários era R\$ 16.000,00. Neste cenário, cada operador era capaz de realizar em média 90 migrações por dia. Logo, 10 operadores realizam 900 migrações de plano.

Vale ressaltar que o custo para adicionar mais robôs atuando é mínimo, uma vez que ela é capaz de ser executada em máquinas virtuais, ou seja, um servidor já existente permitiu a criação de várias máquinas virtuais. Para exemplificar, iremos considerar apenas 10 robôs RPA em execução. Assim, temos que um robô RPA é capaz de realizar 360 operações (migrações) por dia, então, totalizamos 3.600 migrações concluídas em 24 horas. O Gráfico 1 apresenta a produtividade de 10 robôs e 10 operadores ao longo de um ano.

---

<sup>1</sup> Educa mais Brasil. Salário de desenvolvedor web - carreira de desenvolvedor web. Disponível em: <https://www.educamaisbrasil.com.br/cursos-e-faculdades/desenvolvimento-web/salario-desenvolvedor-web-carreira>. Acesso em 27 de out de 2022.

<sup>2</sup> Gupy. Guia de - Operador de Telemarketing. Disponível em: <https://www.gupy.io/profissoes/operador-de-telemarketing>. Acesso em 27 de out de 2022.



**Gráfico 1 - Comparação da quantidade de operações realizadas por um operador e pela RPA**

Não foi possível calcular o custo mensal de infraestrutura para manter o servidor onde há as máquinas virtuais. Como dito acima, não foi necessário adquirir o servidor, porque já havia um no parque tecnológico da empresa. Assim sendo, concluímos que, em 24 horas, o robô RPA produz 400% a mais que um operador de *call center*.

Dessa forma, percebe-se que a RPA aplicada a esse contexto possui uma eficiência muito superior em todos os sentidos e possui um custo muito menor. Apesar de manter um tempo semelhante para se encerrar um ciclo completo (migração), a RPA consegue atuar durante longos períodos de tempo e pode funcionar em várias máquinas virtuais. Além disso, seu custo de manutenção é muito menor que se compararmos com a folha de pagamento onde há vários operadores de *call center* e está muito menos suscetível a erros. Esses fatos estão demonstrados na [Quadro 1](#).

**Quadro 1 - Comparação de métricas da RPA com um operador**

| <b>Métricas</b>                         | <b>Operador</b>             | <b>Robô</b>                             |
|---|-----------------------------|---|
| Conclusão de ciclo (migração)           | 4 minutos                   | 4 minutos                               |
| Capacidade de trabalho                  | 6 horas/dia                 | 24 horas                                |
| Quantidade de agentes                   | A critério da administração | Sem limites, apenas poder computacional |
| Erros                                   | Difícil mapeamento          | Fácil mapeamento                        |
| Custo anual de manutenção / implantação | Alto                        | Baixo                                   |

Além dessas métricas, vale citar alguns fatores indiretos que devem ser levados em consideração: um ser humano está sujeito a fatores externos (doenças, afastamentos, problemas pessoais, etc.) que afetam sua produtividade e o robô está sujeito apenas a eventualidades de infraestrutura.

Outro fator importante a ser levado em consideração é a questão que o processo documentado nesse trabalho é uma atividade extremamente repetitiva, trazendo assim impactos na vida física e psicológica do operador de *call center*. Logo, no processo em questão a RPA se mostra muito vantajosa, tanto para organização, quanto para a saúde dos operadores. Além disso, a adoção e implantação da RPA permitiu reduzir custos, ao mesmo tempo em que mantém a mesma eficiência (produtividade). Finalmente, o uso de RPAs em determinadas operações de um *call center* torna suas operações mais escaláveis.

### **5.1 Indicadores de desempenho**

Indicadores-Chaves de Desempenho ou KPI's (do inglês *Key Performance Indicators*) são indicativos que permitem às organizações entenderem as métricas do seu negócio. Sendo assim, aquelas têm a possibilidade de realizar ajustes precisos em seus processos, para atingir os objetivos desejados.

No caso das RPAs os KPI's também são essenciais, visto que eles permitem uma clara visualização do desempenho da RPA para a organização, ou seja,

forneem métricas que indicam a eficiência da RPA. Nesse sentido, Anagnoste (2018) levanta os seguintes KPI's para RPA:

- **KPI's de Automação:** acompanha a eficiência dos processos automatizados.
- **KPI's da Força de Trabalho Virtual:** acompanha a utilização geral e potencial de otimização dos robôs em execução.
- **KPI's Financeiros:** acompanha mensalmente os benefícios financeiros resultantes do uso dos "robôs".
- **KPI's dos Funcionários:** acompanha o desempenho geral da equipe: seu desenvolvimento, suas necessidades de treinamento.

Dessa forma, pode-se afirmar que as referidas KPI's nesse projeto foram atingidas da seguinte forma:

- **KPI's de Automação:** 100% do procedimento de migração foi automatizado.
- **KPI's da Força de Trabalho Virtual:** a RPA foi utilizada em todo processo, ou seja, automatizou 100% o processo de migração.
- **KPI's Financeiros:** em média R\$ 81.000 de economia mensal.
- **KPI's dos Funcionários:** sem necessidade de treinamento de equipe, uma vez que o procedimento foi automatizado por completo.

## 6 Conclusão

Pode-se considerar que a RPA é um recurso que existe desde o século passado, porém, com as demandas mercadológicas do século XXI, essa vem se tornando cada vez mais comum. Essas automações não estão restritas apenas no mundo físico através de robôs mecânicos, mas também em contextos virtuais, como relatado nesse projeto.

Uma RPA, se bem implementada, resulta em diversos fatores benéficos para uma organização, como: redução de erros e custos, maior eficiência, aumento de produção, melhor controle do fluxo, escalabilidade e outros. No entanto, para desenvolver e implantar uma RPA é necessário um amplo planejamento, como definição de tecnologias que vão integrar o projeto, mapeamento detalhado de rotinas, entendimento criterioso do contexto, alta carga de testes e organização de desenvolvimento como demonstrado nesse trabalho. Vale ressaltar que o desenvolvimento de uma RPA é uma tarefa que envolve muitas variáveis, principalmente quando trata de processos de grandes organizações. Quando se arquiteta uma RPA é necessária mão de obra qualificada, não apenas do time de TI. São necessárias também pessoas que estruturam todas as etapas das rotinas, de colaboradores capazes de gerir o desenvolvimento e dos responsáveis por testar e manter a tecnologia.

É importante ressaltar que apesar da RPA ser extremamente benéfica e atraente para as organizações, ela depende de muitos fatores para ser possível sua implementação. Podemos destacar alguns desses, como: capacidade da organização de custear uma RPA, infraestrutura de *hardware* e *software* que comporte a atuação de uma RPA, processos definidos e esquematizados, sistemas que suportem sua atuação e colaboradores qualificados (tanto por parte da organização desenvolvedora da RPA quanto da contratante do serviço).

Diante a tantos argumentos, podemos concluir que o objetivo desse trabalho foi alcançado ao se abordar todas as etapas de desenvolvimento de uma RPA em um *call center*, partindo da análise de contextos até a apresentação de resultados.



## 6.1 Limitações

A RPA atuou em um sistema web relativamente antigo (que funcionava, até então, somente com o navegador Internet Explorer), que quando foi projetado não foi levado em consideração que uma RPA atuaria no mesmo, mas sim um operador humano. Logo, esse sistema web foi estruturado levando-se em consideração especificidades de um ser humano, fato esse que torna as páginas com maior quantidade de elementos, deixando-as mais sobrecarregadas. Dessa forma, algumas limitações surgiram no projeto. A primeira delas, como disposto no [Pseudocódigo 2](#), foi a necessidade de se colocar um *thread sleep* de alguns segundos para se garantir que a página e todos elementos HTML carregassem por completo. Visto que, em alguns casos, as páginas do sistema eram relativamente lentas, devido a quantidade de elementos. Para que um robô funcione bem interagindo com uma página web, é imprescindível que todos os elementos estejam carregados na tela. O *thread sleep* faz com que o robô espere alguns segundos antes de executar a próxima rotina.

Outro fator a ser observado é a questão dos identificadores dinâmicos de campo HTML (identificador HTML é um código interno que cada componente/objeto da página possui). Nesse sentido tornava-se necessário contornar a situação através da linguagem de programação C# ou interagir com outro identificador estático do elemento HTML.

Vale ressaltar também que o sistema, em diversos casos, contava com *pop-ups* de notificação. Assim, apesar da RPA tratá-los com facilidade, esses não seriam necessários caso o sistema fosse desenvolvido focado na atuação da RPA. Além disso, *pop-ups* deixam a atuação da RPA lenta.

## 6.2 Trabalhos futuros

Como trabalho e projeto futuro seria relevante arquitetar e desenvolver um sistema que trate exatamente do mesmo processo documentado, porém, levando em consideração as especificidades de uma RPA. Em outras palavras, ele deveria ser focado no desempenho da RPA e não nas particularidades de um sistema desenvolvido para um humano operar.

Além disto, seria interessante comparar como a RPA poderia se sair melhor em um sistema desenvolvido especificamente para sua atuação e relatar as dificuldades ainda encontradas. Finalmente, outra sugestão para trabalhos futuros é o desenvolvimento de outra RPA envolvendo o mesmo processo, porém, utilizando tecnologias diferentes.

## Referências

- ABBOTT, R. B., & BOGENSCHNEIDER, B. (2018). **Should robots pay taxes? tax policy in the age of automation**. Harvard Law & Policy Review. 2018.
- MARIANO, A. F. **Automação robótica de processos: uma análise sobre a governança de RPA para grandes empresas**. 2020. 92 f. Diss. Tese (Mestrado) - Fundação Getúlio Vargas Escola de Administração de Empresas de São Paulo, São Paulo, 2020.
- CASEY, k. **Robotic process automation (RPA): how it works**, 2019. Disponível em <https://enterpriseproject.com/article/2019/10/rpa-robotic-process-automation-how-it-works>. Acesso em 9 de ago de 2022.
- MARQUES, Cícero Fernandes. **Estratégia de gestão da produção e operações**. Curitiba, 2012.
- WILLCOCKS, Leslie; LACITY, Mary; CRAIG, Andrew. **The it function and robotic process automation**. The Outsourcing Unit Working Research Paper Series, 2015.
- LACITY, MC; WILLCOCKS, LP. **Robotic process automation at Telefónica O2**. MIS Quarterly Executive, 2015.
- KIRCHMER, M; FRANZ, P. **Value-driven Robotic Process Automation (RPA), a process-led approach to fast results at minimal risk**. International Symposium on Business Modeling and *Software Design*. 2019.
- SANTIAGO, A.; RODRIGUEZ, A. **Automation of a business process using robotic process automation (RPA): A case study**, J.C. Figueroa-García et al. (Eds.): WEA 2017, CCIS 742. 2017.
- GIRÃO, Bruna. **Automação Robótica de Processos (RPA): análise da aplicação de RPA para o processo de retorno de reclamação e solicitação de clientes na hyperion services**. 2021. Monografia - Centro Universitário Christus, Unichristus, Fortaleza, 2021.
- MARIANO, Alex. **Automação robótica de processos: uma análise sobre a governança de RPA para grandes empresas**.2020. Tese - Escola de Administração de Empresas de São Paulo, Fundação Getulio Vargas, São Paulo, 2020.
- SHERMAM, Amanda; RODRIGUES, Ana; **A influência da implantação de RPA (Robotic Process Automation) nos processos relacionados a emissão de notas em uma empresa do ramo metalúrgico**. 2018. Monografia – Universidade Federal do Rio Grande do Sul, 2018.
- JÚNIOR, Vicente. **Automação de processos de negócio utilizando Robotic Process Automation (RPA) em um centro de serviços compartilhados (csc): um estudo de caso**. 2021. Monografia – Universidade Federal de Uberlândia, Uberlândia, 2021.

Selenium. **O projeto Selenium de automação de navegadores**. 2022. Disponível em: <https://www.selenium.dev/pt-br/documentation>. Acesso em 7 de ago de 2022.

Microsoft. **Um tour pela linguagem C#**. 2022. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>. Acesso em 8 de ago de 2022.

DEVARAJAN, Yuvaraja. **A study of robotic process automation use cases today for tomorrow's business**. 2018. International Journal of Computer Techniques. 2018

SÉRVIO, Gabriel. **Call center: metade dos atendentes admitem falhas em contatos com os clientes, diz estudo**. 2022. Disponível em: <https://olhardigital.com.br/2022/03/17/pro/call-center-metade-dos-atendentes-admitem-falhas-em-contatos-com-os-clientes-diz-estudo/>. Acesso em 27 de out de 2022.

SOARES, Michel. Metodologias ágeis extreme programming e scrum para o desenvolvimento de *software*. 2004. **Revista Eletrônica de Sistemas de Informação**, v. 3, n. 1, 2004.

OLIVEIRA, Luisa. **Modelo de gerenciamento ágil de projetos utilizando a metodologia kanban: aplicação em uma empresa de *software***. 2020. Monografia - Universidade Federal de Santa Catarina (UFSC), 2020.