

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PABLO MARTINEZ SLOMP

**DESENVOLVIMENTO DE UM SISTEMA EM MICROSERVIÇOS PARA A
CONTROLE E GERENCIAMENTO DE RPAS**

GUARAPUAVA

2023

PABLO MARTINEZ SLOMP

**DESENVOLVIMENTO DE UM SISTEMA EM MICROSSERVIÇOS PARA A
CONTROLE E GERENCIAMENTO DE RPAS**

Development of a Microservices System for RPA Control and Management

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Paulo Henrique Soares

GUARAPUAVA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

PABLO MARTINEZ SLOMP

**DESENVOLVIMENTO DE UM SISTEMA EM MICROSERVIÇOS PARA A
CONTROLE E GERENCIAMENTO DE RPAS**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação
do Curso de Bacharelado em Ciência da
Computação da Universidade Tecnológica
Federal do Paraná.

Data de aprovação: 01/julho/2023

Nome completo e por extenso do Membro 1
Título (especialização, mestrado, doutorado
Nome completo e por extenso da instituição a qual possui vínculo

Nome completo e por extenso do Membro 2
Título (especialização, mestrado, doutorado
Nome completo e por extenso da instituição a qual possui vínculo

Nome completo e por extenso do Membro 3
Título (especialização, mestrado, doutorado
Nome completo e por extenso da instituição a qual possui vínculo

**GUARAPUAVA
2023**

Quem nunca cometeu um erro nunca tentou
nada novo.
(A citação costuma ser atribuída a Albert
Einstein, mas não há indícios que confirmem
essa autoria)

RESUMO

As automações de processos tornou-se algo essencial em linhas de produção para as empresas manterem-se competitivas no mercado. A diferenciação tem sido na robotização de procedimentos administrativos por meio de RPAs (Robotic Process Automation). A partir do momento que eles vão sendo criados e o número de tarefas automatizadas cresce, a controle sobre eles e sobre as informações que eles geram torna-se cada vez mais necessário. Este documento demonstra a estrutura de um sistema em microsserviços para o controle de RPAs e visualização de dados, dentro de uma empresa do ramo financeiro, de forma que possa ser escalável e adaptável ao crescimento de robôs desenvolvidos ao longo do tempo.

Palavras-chave: rpa; automação; microsserviço; desenvolvimento.

ABSTRACT

Process automation has become essential in production lines for companies to remain competitive in the market. Differentiation has been achieved through the robotization of administrative procedures using RPAs (Robotic Process Automation). As they are created and the number of automated tasks increases, control over them and the information they generate becomes increasingly necessary. This document outlines the structure of a microservices system for RPA control and data visualization within a financial company, aiming to be scalable and adaptable to the growth of robots developed over time.

Keywords: rpa; automation; microservice; developmet.

LISTA DE FIGURAS

| | |
|--|-----------|
| Figura 1 – Estrutura atual de funcionamento dos RPAs | 12 |
| Figura 2 – Quadrante Mágico do Gartner | 15 |
| Figura 3 – Arquitetura monolítico | 18 |
| Figura 4 – Arquitetura de microsserviços | 18 |
| Figura 5 – Manutenção em microsserviços | 19 |
| Figura 6 – Desenvolvimento em módulos | 24 |
| Figura 7 – Configuração dos módulos do sistema de microsserviço | 26 |
| Figura 8 – Estrutura do Controlador | 27 |
| Figura 9 – Projeção do layout do módulo Visualizador | 28 |
| Figura 10 – Estrutura base dos documentos do BD | 28 |

LISTA DE ABREVIATURAS E SIGLAS

Siglas

| | |
|------|-----------------------------------|
| API | Application Programming Interface |
| BD | Banco de Dados |
| JSON | JavaScript Object Notation |
| RPA | Robotic Process Automation |

SUMÁRIO

| | | |
|------------|--|-----------|
| 1 | INTRODUÇÃO | 8 |
| 1.1 | Considerações iniciais | 9 |
| 1.2 | Objetivos | 10 |
| 1.2.1 | Objetivo geral | 10 |
| 1.2.2 | Objetivos específicos | 10 |
| 1.3 | Justificativa | 10 |
| 1.4 | Estrutura do trabalho | 11 |
| 2 | ANÁLISE DE DOMÍNIO | 12 |
| 3 | TRABALHOS RELACIONADOS | 14 |
| 3.1 | Mercado de RPA | 14 |
| 3.1.1 | UIPath | 14 |
| 3.1.2 | Automation Anywhere | 15 |
| 3.1.3 | Power Automate | 16 |
| 4 | REFERENCIAL TEÓRICO | 17 |
| 4.1 | RPA - <i>Robotic Process Automation</i> | 17 |
| 4.2 | Microsserviços | 18 |
| 4.3 | Python | 19 |
| 4.4 | MongoDB | 20 |
| 5 | METODOLOGIA DE DESENVOLVIMENTO | 21 |
| 5.1 | Materiais | 21 |
| 5.2 | Métodos | 22 |
| 5.3 | Arquitetura Base do Sistema | 23 |
| 6 | RESULTADOS | 25 |
| 6.1 | Estrutura do Sistema | 25 |
| 6.2 | Esquema Base da API | 28 |
| 7 | CONSIDERAÇÕES FINAIS | 30 |
| | REFERÊNCIAS | 31 |

1 INTRODUÇÃO

A formação de organizações se dá pelo relacionamento entre recursos, esforços e pessoas, as quais realizam tarefas diferentes com o intuito de alcançar objetivos específicos. Para que ocorra essa interação, a comunicação entre setores e pessoas é primordial, seja ela visual, oral ou tátil. Nessa troca de informações pode ocorrer interferências vindas de diversas variáveis, tais como os relacionamentos existentes entre os envolvidos, a diferença entre níveis de hierarquia, falhas na linguagem e o entendimento do que foi transmitido. Esses elementos são exemplos de causadores de distorções que podem ocorrer na comunicação (SOUZA, 2021). Essas deturpações podem impactar na eficiência do trabalho, tanto no fluxo quanto no resultado desse.

Na busca por melhorias na realização de processos, e assim se manter competitiva no mercado, empresas vêm há anos automatizando tarefas. Esse cenário pode ser observado no segmento industrial, onde às empresas já demonstraram que é possível ter um maior controle sobre as etapas existentes, do ganho de eficiência e tempo, além de possibilidade de extração de dados, estatísticos e de desempenho, permitindo uma melhor avaliação dos processos (SANTOS BRUNA APARECIDA DE OLIVEIRA, 2017). Porém, não é apenas em linhas de produção industrial que essa mudança tem ocorrido. Empresas têm buscado e selecionado atividades administrativas e rotineiras que possam ser automatizadas (MANZUETO, 2016).

Uma automação pode ter diferentes níveis de abrangência. O uso de softwares específicos, como um ERP ou um programa de controle e análise de dados permitem que algumas etapas de um processo sejam executadas de maneira automática, como a geração de gráficos dinâmicos ou o cálculo de uma grande quantidade de dados. Contudo, se o objetivo é a automação de várias etapas, diminuindo consideravelmente a intervenção humana, uma solução é a Automação Robótica de Processos (Robotic Process Automation (RPA)).

Um RPA é um software que tem a capacidade de realizar tarefas rotineiras e que não necessitem de análises complexas, como se fosse uma pessoa, só que com mais precisão e velocidade (SILVA, 2018). Ele é capaz de interagir com sistemas on-line e offline, lendo e interpretados dados para os quais foi programado. Para que isso ocorra é necessário que o robô tenha acesso a uma máquina e os devidos acessos, bem como as regras do processo. Dentro deste contexto, um RPA pode ser visto como um trabalhador não humano dentro das rotinas e fluxos de trabalho. Contudo, ao se decidir implementar uma automação robótica de processo é importante que sejam respondidas algumas questões, como: Qual a responsabilidade dele? Quem é o responsável por ele? Como se dará a comunicação entre as pessoas envolvidas e o robô?

As divisões de hierarquia e responsabilidades dentro de uma organização ocorre por meio de um organograma, o qual transmite a informação da responsabilidade de cada funcionário, bem como a divisão de trabalho para cada um. Além disso, com base nesses dados cada colaborador sabe com quem se comunicar dependendo do processo que está sendo re-

alizado (SOUZA, 2021). Na intenção de implementar RPAs nos processos, essas informações são cruciais na hora de desenvolvê-los, afinal, a interação entre eles e as pessoas envolvidas irá ocorrer, e para isso é preciso saber como ela irá ocorrer, bem de que maneira será controlada e gerenciada.

Um sistema que consiga controlar os RPAs, bem como filtrar e disponibilizar informações por eles geradas é de crucial importância para que a empresa saiba o que está acontecendo com os processos, ao mesmo tempo que possa tomar decisões com base dados existentes. A falta de uma aplicação que gerencie tudo isso pode acarretar problemas, como a execução de tarefas que já foram realizadas ou a queda na eficiência, pois todos os dados estariam em posse da equipe de desenvolvimento dos robôs.

Esse projeto ocorre dentro de uma empresa do ramo financeiro no interior do Paraná, a qual já possui alguns RPAs em funcionamento. Contudo, não dispõem de nenhuma ferramenta de controle e gerenciamento de informações.

1.1 Considerações iniciais

Atualmente, na empresa onde esse projeto será realizado, todo o controle e informações referentes aos RPAs está centralizando no time de automação da empresa. Se ocorrer qualquer falha ou um cenário que se diferencie do normal, apenas essa equipe terá o conhecimento sobre a situação. Isso acarreta que por mais que o robô esteja alocado em outra área, fica a cargo dos desenvolvedores verificar a situação de cada RPA, sobrecarregando a área com tarefas que saem do seu escopo. O intuito deste projeto e o desenvolvimento de uma aplicação em microsserviços que organize esses robôs bem como facilite interação deles com as pessoas envolvidas nos processos. Esse sistema será dividido em 3 módulos:

- **Controlador:** responsável pelo controle de todas as informações das tarefas a serem feitas pelos RPAs, além da interação com alguns sistemas de terceiros;
- **Serviço:** módulo acoplado ao RPA, responsável por filtrar e armazenar as informações da tarefa recebida para ser executada;
- **Visualizador:** uma aplicação web. A plataforma que expõe informações sobre as tarefas que foram ou serão feitas de forma automatizada, além da possibilidade de enviar informações novas para a execução delas.

Toda a comunicação entre os módulos será realizada por meio de Application Programming Interface (API) bem definida e cada módulo será implementado dentro de uma máquina virtual (VM - *virtual machine*) e/ou máquina física.

1.2 Objetivos

1.2.1 Objetivo geral

Desenvolver uma aplicação de microsserviços para o controle e gerenciamento de informações de RPAs utilizados por uma instituição financeira específica, localizada no estado do Paraná.

1.2.2 Objetivos específicos

1. Levantar os dados de entrada e saída dos RPAs
 - a) Padronizar dos dados de entrada e saída dos RPAs no módulo de serviço;
 - b) Padronizar no envio e recebimento de dados no módulo de controle.
2. Desenvolver de um Banco de Dados (BD) principal
 - a) Centralizar do BD no módulo de controle;
 - b) Criar uma API para a escrita e leitura do BD pelo módulo de visualização;
 - c) Criar uma API para a escrita e leitura do BD pelo módulo de serviço.
3. Analisar a comunicação dos RPAs com aplicações de terceiros
 - a) Analisar a necessidade de centralização da comunicação à aplicações de terceiros;
 - i. Centralizar da comunicação com aplicações de terceiros ao módulo de controle.
4. Desenvolver o módulo de visualização
 - a) Criar do *layout*;
 - b) Criar de um nome e logotipo.
5. Dimensionar do servidor
 - a) Máquina física *versus* Máquina virtual;
 - b) Escolher o sistema operacional.

1.3 Justificativa

O uso de RPAs para a a diminuição no tempo de execução dos processos, bem como o ganho de assertividade, tem chamado a atenção de empresas, como uma maneira de reduzir os

custos e se manter competitiva no mercado. Contudo, a implementação cega de robôs, sem o devido controle, pode ocasionar falhas a efetividade que podem ser ocasionadas, por exemplo, por falta de informações ou pelo acúmulo de funções em uma determinada equipe.

Com a implementação deste sistema, espera-se que qualquer pessoa envolvida no processo tenha acesso às informações das tarefas que os RPAs fizeram, estão fazendo e/ou irão fazer, além de disponibilizar dados para controle gerencial, tais como quantidade de processos automatizados ou as datas e horas que foram finalizados. Outro ponto é a redução de requisições e acessos paralelos realizados a sistemas de terceiros, pois elas ficarão centralizadas no Controlador. Desta forma, obtém-se um maior controle e transparência nas informações geradas por RPAs.

1.4 Estrutura do trabalho

Este trabalho está dividido da seguinte maneira:

- Capítulo 2: disserta sobre o referencial teórico;
- Capítulo 3: apresenta as tecnologias escolhidas e o levantamento do cenário de desenvolvimento
- Capítulo 4: apresenta o panorama atual do mercado de automações;disserta sobre o referencial teórico;
- Capítulo 5: descreve a metodologia de desenvolvimento;
- Capítulo 6: apresenta o cenário atual da empresa onde o projeto será desenvolvido;
- Capítulo 7: expõe os resultados do projeto e suas estruturas;
- Capítulo 8: apresenta as considerações finais.

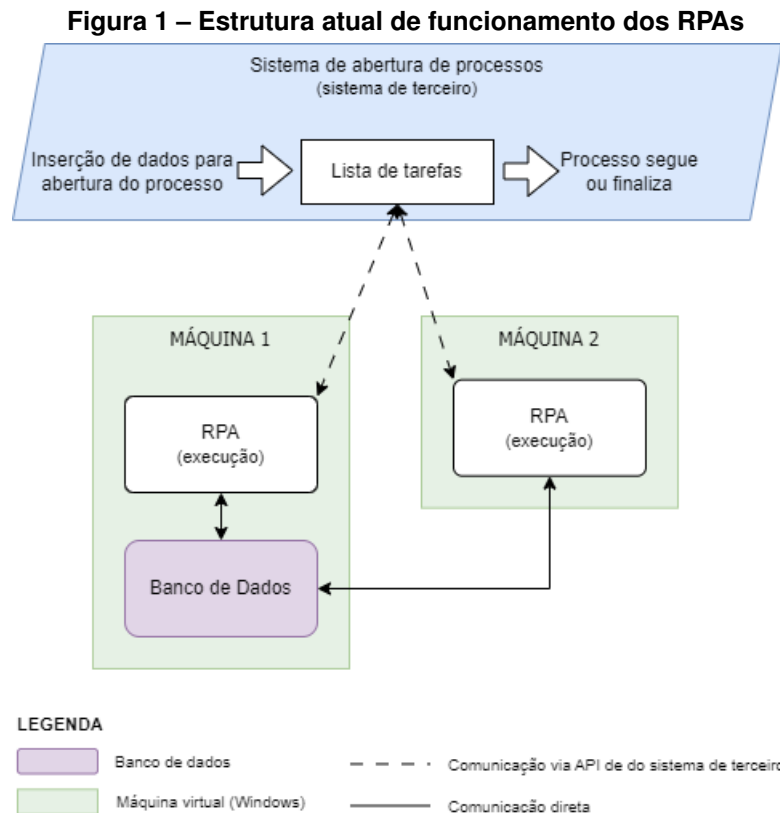
2 ANÁLISE DE DOMÍNIO

A empresa na qual esse projeto será desenvolvido tem mais de 120 anos de história e está presente em todo o Brasil. Para conseguir atender todo o território de maneira eficiente, existem várias sedes em diversos estados. A sede na qual haverá a implementação deste projeto tem mais de 40 anos de funcionamento.

Atualmente essa sede possui 2 RPAs que são responsáveis pela execução de mais de 15 processos automatizados. Cada um dos robôs trabalha como um funcionário, portanto necessitam, por exemplo, de usuário e senha para acessar os sistemas e um computador próprio, para conseguir interagir com as aplicações, seja coletando, tratando ou enviando informações.

Sempre que uma nova automação entra em produção, é importante para a equipe de desenvolvedores estar preparada para constantes manutenções e atualizações desse, já que é possível que um ou mais processos que sejam executados pelos robôs passem por alterações, seja no fluxo ou mesmo na aplicação necessária para a realização do mesmo. Devido a isso, a equipe de automação mantém seus códigos disponíveis para toda equipe, para que a constante refatoração seja feita em um ritmo sustentável e que as atualizações sejam integradas aos RPAs continuamente.

A estrutura de funcionamento básica dos RPAs pode ser visualizada na Figura 1 e seguem os seguintes passos:



Fonte: Autor.

1. A tarefa é aberta dentro de um sistema de processos, onde são preenchidas todas as informações necessárias e enviadas para a próxima etapa;
2. Os RPAs verificam suas listas de tarefas e as coletam, armazenando alguns dados em um banco de dados central;
3. Os RPAs lêem o BD e coletam as tarefas mais antigas para executar;
4. Voltam ao sistema de processos e com base no número da tarefa que pegaram no BD, coletam os dados dela;
5. Com base nesses dados, acessam os sistemas necessários, executam a tarefa e armazenam os dados no BD;
6. Retornam ao sistema de processos e dão prosseguimento à tarefa, que pode ir para outra área ou ser finalizado.

Como pode ser observado, o fluxo de funcionamento é simples e por mais que seja funcional, não é eficiente e muito menos acessível. Todas as informações geradas pelos RPAs estão no banco de dados, e não existe acesso prático para elas. Apenas a equipe de desenvolvimento das automações tem acesso a elas. Desta maneira, apenas os desenvolvedores sabem o que está sendo feito, o quanto foi feito ou se houve falha ou não. Ou seja, se alguém da empresa necessitar de uma informação, deve solicitar aos desenvolvedores e esses devem realizar essa busca. Isso acarreta em uma sobrecarga de tarefas que estão além do escopo de trabalho dos programadores, pois devem estar sempre atentos se os processos foram executados e reiniciar caso tenha dado alguma falha. Diante deste cenário, é possível visualizar que além de desenvolver os RPAs a equipe de automações fica responsável por todo o gerenciamento dos robôs bem como das informações geradas por eles.

Além disso, manutenções e atualizações podem gerar grandes pausas nas execuções automáticas, pois se uma tarefa tiver passado por alteração em seu fluxo, deve ser acessado cada um dos robôs para que eles não a executem até que sejam realizadas as devidas mudanças. Depois disso, voltar a acessar cada um deles e voltar a colocar em produção essa tarefa. Enquanto são apenas 2 RPAs, isso não aparenta ser um problema, mas ao se aumentar esse número, pode vir a ser um gargalo no processo.

3 TRABALHOS RELACIONADOS

Já é natural as empresas automatizarem linhas de produção com a utilização de robôs e/ou máquinas (SANTOS BRUNA APARECIDA DE OLIVEIRA, 2017), mas a competitividade tem impulsionado a busca por maneiras de automatizar seus processos administrativos. Essa procura resultou em um crescimento de 31% mundialmente em 2021 no mercado de RPA, e espera-se um aumento de mais de 17% para 2023. Com o objetivo de aproveitar esse crescimento, companhias desse setor têm investido cada vez mais no desenvolvimento das suas aplicações de automação (STAMFORD, 2022).

Cabe ao futuro cliente dessas companhias definir qual escolher, visto que, sendo criados processos automatizados utilizando-se a aplicação de uma dessas empresas, não é possível adquirir um módulo de controle ou de visualização de informações de outra companhia. Isso ocorre pelo fato que esses dois últimos são desenvolvidos exclusivamente para funcionarem com os RPAs criados pelo software da mesma. Tal fato se origina da quantidade de variáveis que existem ao se desenvolver uma automação. Torna-se praticamente impossível o criação de módulos de controle, por exemplo, genéricos que funcionem para diferentes aplicações de desenvolvimento RPA.

3.1 Mercado de RPA

No cenário atual, CXTODAY (2022) cita 15 empresas integrantes nesse mercado por meio do Quadrante Mágico do Gartner¹ (Figura 2), no qual 5 delas são consideradas líderes de mercado. Dessas, são três as estão em evidência atualmente.

3.1.1 UiPath

Empresa fundada na Romênia em 2005, é a líder do segmento (CXTODAY, 2022). Possui um ecossistema diversificado, com diversas integrações e bibliotecas próprias, possibilitando sua aplicação nos mais diferentes segmentos empresariais (UIPATH, 2023). Sua interface por meio de blocos possibilita que pessoas sem conhecimento aprofundado em programação consigam automatizar tarefas simples.

Para o controle dos RPAs desenvolvidos por meio da plataforma, o UiPath possui um sistema on-line no qual é possível gerenciar as automações, verificando se estão em funcionamento, agendando tarefas, controlando versões. Uma limitação ocorre no fato que essa plataforma funciona plenamente apenas em RPAs que são executados de maneira automática,

¹ *Magic Quadrant do Gartner*, é um gráfico que exhibe o posicionamento competitivo entre empresas de desenvolvimento de um mesmo mercado, dividindo-as em Líderes, Visionários, Operadores de Nicho e Competidores (GARTNER, 2023).

Figura 2 – Quadrante Mágico do Gartner



Fonte: CXTODAY (2022).

os quais possuem um valor de licença maior do que os que necessitam de uma inicialização manual.

3.1.2 Automation Anywhere

Empresa americana, fundada em 2003, tem como seu principal diferencial o foco em nuvem, podendo ser implantando tanto, por exemplo, no Google Cloud e no AWS (CXTODAY, 2022). Está presente em mais de 90 países e mais de 2,8 milhões de RPAs implementados (ANYWHERE, 2023).

Sua plataforma de gerenciamento permite a inserção de dados financeiros sobre a mão de obra que está sendo economizada com a automação, e em conjunto com o tempo de execução de cada automação, é possível a visualização, por exemplo, de informações sobre a economia gerada e do tempo ganho devida a rapidez da execução, por meio dos RPAs.

3.1.3 Power Automate

Pertencente à Microsoft, surgiu em resposta ao mercado. Anteriormente a empresa tinha apenas o Power RPA, que era incorporada a sua plataforma em nuvem Azure. O Power Automate possui uma versão gratuita para todos os usuários de Windows 10 e 11 (CXTODAY, 2022). Por pertencer à Microsoft, ele possui uma poderosa integração com todos os softwares dela, fazendo disso um grande diferencial em relação aos seu concorrentes.

A visualização e controle de informações do Power Automate é bem completa, trazendo informações como pontos de gargalos, gráficos sobre tempo e verificação de pontos de melhoria (AUTOMATE, 2023). Contudo, é importante salientar que determinadas funções ficam disponíveis apenas em planos pagos.

4 REFERENCIAL TEÓRICO

Este capítulo apresenta as definições necessárias para um melhor entendimento do projeto. Primeiramente é descrito sobre *Robotic Process Automation* (RPA), passando pelo conceito de microsserviço e sua comparação com um sistema monobloco, para então descrever alguns detalhes sobre a linguagem Python e o banco de dados MongoDB.

4.1 RPA - *Robotic Process Automation*

A automação robótica de processos pode ser descrita como a utilização de um ou mais robôs virtuais, que executam as tarefas para as quais foram programados, como se fossem seres humanos. É possível abrir, interagir, ler e enviar informações para quaisquer sistemas que se tenha acesso (SILVA, 2018). Em outras palavras, é um software previamente configurado para imitar a ação humana em tarefas repetitivas e estruturadas (JÚNIOR, 2021). Esses dois pontos são cruciais, pois, ainda, não é possível para os robôs julgar situações e tomar decisões com base nelas. De acordo com Cabral Ariel Behr (2022) as melhores tarefas para serem automatizadas possuem as seguintes características:

- Baixo requisitos cognitivos;
- Alto volume;
- Acesso a muitos sistemas;
- Baixo índice de exceções;
- Alto índice de falha humana.

Porém, a principal motivação para a implementação de RPAs está na otimização dos custos (CABRAL ARIEL BEHR, 2022). Devidamente projetados, um robô de processo pode executar uma tarefa de maneira muito mais rápida e assertiva se comparado com um ser humano. Contudo, para que se desfrute desses ganhos, é necessário um alinhamento bem estruturado dentro da empresa para que ocorra a implementação de um RPA.

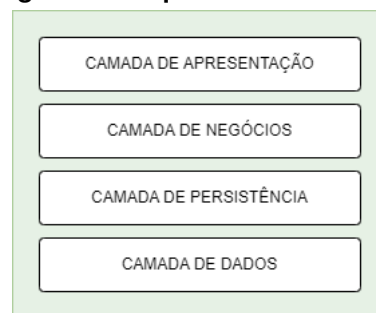
Como observador por MARIANO (2021), automações não devem ser desenvolvidas sem a devida orientação da equipe de TI, pois podem acarretar em problemas de segurança e desempenho, impactando, por exemplo, na rede da empresa. Outro ponto é a necessidade de documentação para que seja possível a realização de auditorias interna, evitando-se, por exemplo, vazamento de informações.

4.2 Microsserviços

Antes de se desenvolver uma aplicação, é importante elaborar a estrutura dela. Quantos e quais os elementos deve conter, como vão se comunicar entre si e quais propriedades de cada um deles (MEDES, 2021). Havendo o esboço dessas informações, é necessário organizá-los. Dentro dessa ordenação, dois padrões se destacam: monolítico e microsserviços.

Um sistema monolítico caracteriza-se por conter toda sua base de código compreendida em um só lugar, e os dados que alimentam-no ficam centralizados em uma base compartilhada (JÚNIOR, 2019). Ou seja, todas as camadas que o compõem fazem parte da mesma estrutura (Figura 3). Ele pode conter um único processo ou vários, porém, no segundo caso, todos eles precisam ser implementados juntos.

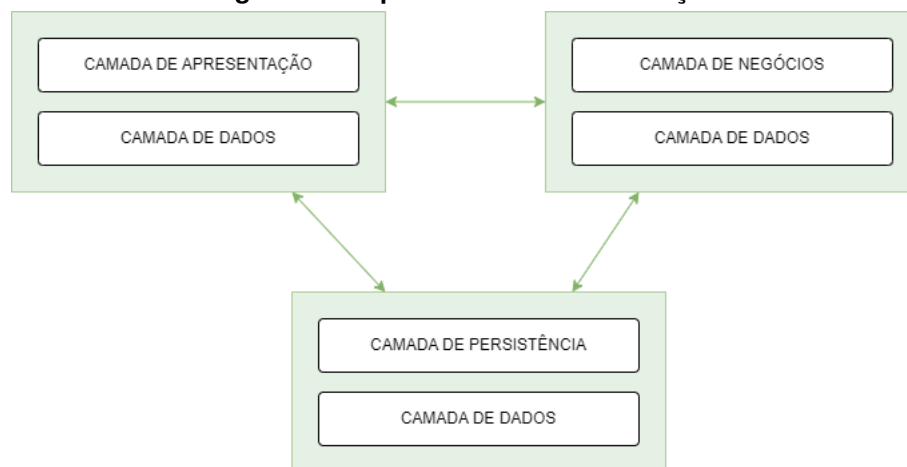
Figura 3 – Arquitetura monolítico



Fonte: (JÚNIOR, 2019) - Editada.

Por ser altamente acoplada, uma aplicação monolítica não é facilmente escalável. Diferente da sua criação, que tende a ser de fácil desenvolvimento, cada alteração que se faz em uma parte do código pode influenciar ele como um todo, tornando sua ampliação uma tarefa custosa.

Figura 4 – Arquitetura de microsserviços



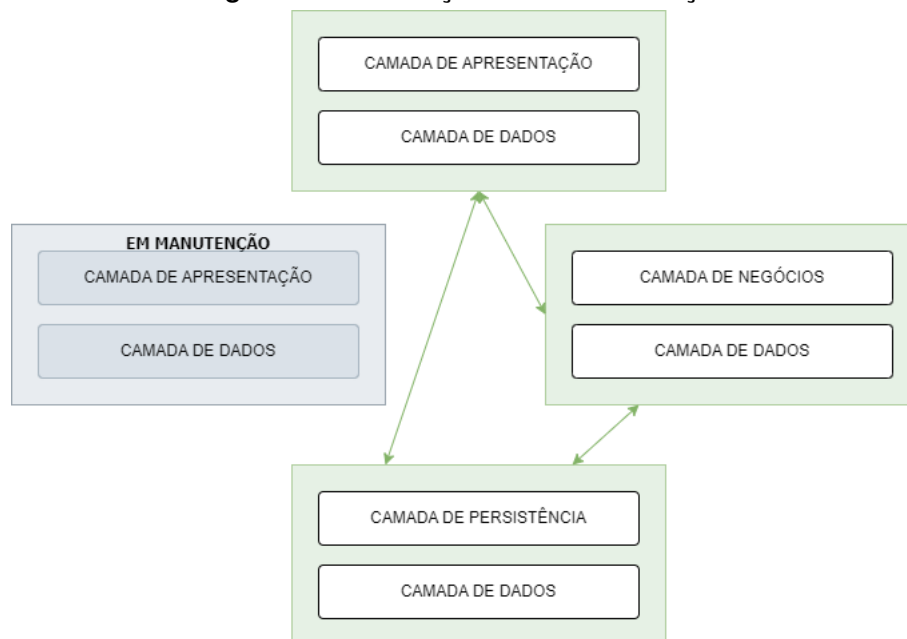
Fonte: Autor.

Por outro lado, um sistema em microsserviços tem suas camadas independentes entre si. A AWS define microsserviços como "Microsserviços são uma abordagem arquitetônica

e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas". Em outras palavras, essa arquitetura é composta por pequenas parte autônomas que se comunicam entre si para realizarem a entrega (Figura 4).

Diferentemente da estrutura monolítica, um sistema de microsserviços é altamente escalável. Como cada módulo trabalha independentemente, é possível realizar upgrades em deles sem que se necessite mexer em todos os outros, nem mesmo que seja necessário parar a aplicação para isso. Isso é possível pela possibilidade de replicar cada um dos serviços como for necessário. Por exemplo, se for necessário fazer alguma alteração na camada de negócios, é possível replicar ela. Dessa forma, ela continua funcionando enquanto se modifica a cópia. Isso sem precisar interferir nas outras camadas (Figura 5).

Figura 5 – Manutenção em microsserviços



Fonte: Autor.

Contudo, para um funcionamento fluído, a comunicação entre os serviços deve ser coesa (JÚNIOR, 2019). Cada camada realiza suas tarefa individualmente, mas para que o objetivo do processo seja alcançado é preciso que as trocas de dados entre elas esteja devidamente estruturada. Não é possível que a camada de apresentação exiba as informações que foram pedidas para a camada de negócios, se esta não envio os dados corretamente.

4.3 Python

A linguagem Python é considerada uma linguagem considerada democrática, muitas vezes sendo usada como primeiro contato à programação em escolas (DIVINO, 2021). Isso

significa que é de rápido aprendizado. Uma das características que contribui para isso é o fato dela ser flexível, permitindo o programador de maneiras diferentes, as quais são:

1. Procedural, onde código é executado na sequência na qual foi criado;
2. Orientado a objetos, a qual é organizada em estruturadas denominadas classes e objetos, permite programação multiplataforma;
3. Funcional, que organiza em blocos chamados de funções, que realizam tarefas específicas.

Outro ponto importante é o fato dela ser muito reconhecida na área de análise de dados, possuindo diversas bibliotecas voltadas para esse cenário (DIVINO, 2021). Isso a torna super versátil ao se trabalhar uma grande quantidade de dados.

4.4 MongoDB

Os bancos relacionais são os mais conhecidos, principalmente por eles implementarem do modelo ACID, que é a sigla para atomicidade, consistência, isolamento e durabilidade. Esses quatro pilares asseguram que o banco de dados continuará consistente após a execução de uma transação (SANTOS, 2019).

Contudo, quando essas características não são necessárias, nem mesmo que haja relação entre diferente grupos de dados, o uso de banco de dados relacionais se torna ilógico. Para esse cenário, existem os bancos NoSQL. Estes tem com sua principal característica é que seu armazenamento é feito por meio de documentos, os quais normalmente utilizam a formatação JavaScript Object Notation (JSON) (LIMA, 2021). E cada um desses documentos podem ser diferentes um dos outros, e armazenados em um mesmo local. Esse local, em um modelo relacional seria uma tabela, aqui é conhecido como *collection*.

5 METODOLOGIA DE DESENVOLVIMENTO

A ênfase deste capítulo está em reportar o que e como será feito para alcançar o objetivo do trabalho. Este capítulo está subdividido em duas seções, sendo uma para os materiais e outra para os métodos.

5.1 Materiais

Analisando-se as informações dos RPAs existentes na empresa, bem como os dados que são e/ou devem ser armazenados, dois pontos importantes para o desenvolvimento foram decididos: a linguagem de programação a ser utilizada e qual modelo de banco de dados aplicado no sistema.

Quanto à linguagem, foi escolhida a Python. Essa decisão foi julgada como a melhor pelas seguintes questões:

1. Atualmente o Python é usado pelo time que desenvolve às automações dentro da empresa. Por esse motivo, não é necessário que as pessoas que o compõem precisem aprender uma outra linguagem de programação, o que facilita a implementação do projeto, bem como a comunicação entre o sistema e os RPAs;
2. Possui uma sintaxe de fácil aprendizado. Isso pode facilitar em futuras contratações pela empresa, que pode vir aceitar pessoas que conhecem outras linguagens, pois podem aprender a com relativa rapidez;
3. Como o sistema vai ter múltiplos serviços, esses poderão trabalhar de maneiras diferentes, e o Python permite o programador desenvolver códigos de mais de uma maneira, como procedural e orientado a objetos;
4. Dentro da empresa utiliza-se diversos relatórios, e muitos deles podem vir a ser importantes para uma execução de tarefas mais rápidas pelos robôs. Para que essas informações sejam enviadas já filtradas, é necessário que seja prático trabalhar com esses dados.

No que diz respeito ao modelo de banco de dados, atualmente os RPAs utilizam o SQL por meio do MySQL. Contudo, foi analisado que uma das principais características desse tipo de BD não está e nem precisa ser utilizada, a de dependências e relacionamento entre os dados. Devido ao fato que as informações que serão armazenadas são apenas de transferência, ou seja, estão em um lugar, precisam ser guardadas em outro e então passadas para destino, não há necessidade de tudo o que um modelo de BD relacional disponibiliza. Além do fato que qualquer informação crucial que exista no BD do sistema, existe em outras aplicações da empresa. Outro ponto é que a quantidade e tipo de informações necessárias para cada processo

automatizado pode se diferenciar entre elas, ou seja, é necessário que o banco de dados seja flexível, possibilitando o armazenamento de diferentes informações em um mesmo local.

Como consequência a esse cenário, foi optado pela utilização de um BD NoSQL, mais precisamente pelo uso do MongoDB. Visto que a grande maioria das informações a serem armazenadas são referentes às API de comunicação entre os módulos de serviço do sistema, que são dados organizados em um JSON, é natural que o armazenamento delas ocorra dessa mesma maneira, sem a necessidade de conversões e/ou formatações.

5.2 Métodos

O presente estudo caracteriza-se pela coleta de dados para o desenvolvimento a partir de uma análise observatória e documental dos RPAs existentes na empresa onde este projeto está sendo realizado.

Para que desenvolvimento do sistema seja viável e eficiente, as seguintes informações são necessárias:

1. **Informações utilizadas pelas automações:** se existe e quais são as informações mais frequentes utilizadas;
2. **Tipos de automação:** quais tipos de automações possíveis e existentes;
3. **Aplicações acessadas:** quais as aplicações de terceiros que os RPAs acessam bem como de qual tipo são (web, embarcado, remoto...), além do levantamento das mais acessadas;
4. **Informações dos RPAs:** quais informações que a empresa deseja ter sobre os robôs e as tarefas que esses realizam;
5. **Padrão de Armazenagem de Dados:** se existe um padrão de onde e como são armazenados dados sobre os processos executados pelos RPAs;
6. **Inicialização dos Robôs:** como é feito o start de cada processo para que seja executado de forma automatizada;
7. **Frequência de Atualizações dos RPAs:** com que frequência um RPA para por atualização e quais os sistemas que mais impactam na realização delas;
8. **Nível de Alterações Possíveis:** o quanto uma automação pode ser modificada para se adaptar ao sistema;
9. **Recursos Disponíveis:** os recursos que a empresa disponibiliza para a realização e implementação do desenvolvimento do sistema de microsserviços;

A fim de elaborar a estrutura entre os módulos do sistema de microsserviços (Controlador, Serviço e Visualizador) e de se esquematizar a API, quais serão as informações básicas a serem salvas e quais as que deverão transitar entre os serviços, foram realizados 2 tipos de levantamentos:

- Acompanhamento de processos automatizados: durante várias semanas, diversas tarefas que já são executadas de forma autônoma foram observadas para o levantamento das informações que elas necessitam para que sejam realizadas, além das que são disponibilizadas após a finalizadas. Com elas em mãos, foi possível analisar também quais os dados que são comum entre elas.
- Projeção de futuras automações: com base em conversas com diferentes áreas da empresa e uma análise geral dos trabalhos que são realizados dentro de cada uma delas, optou-se por imaginar cenários futuros e quais as informações seriam necessárias para o acompanhamento de tarefas automatizadas. Isso foi realizado com o objetivo de dimensionar as informações da API, para evitar muitas modificações futuramente.

Com base nos dados que devem compor a API, é possível prever como a comunicação entre os serviços deverá ser feita, bem como quais os filtros cada um deles deverá ter para que a API seja enviada, lida e devolvida de maneira eficiente.

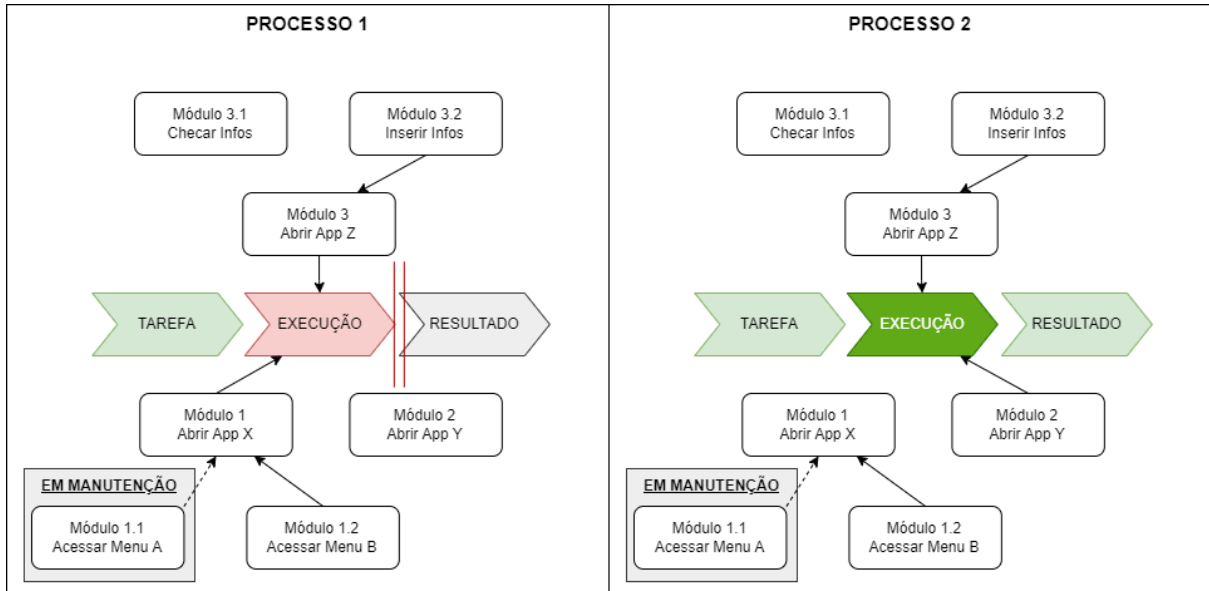
5.3 Arquitetura Base do Sistema

O objetivo é iniciar a implantação pelo módulo de controle. Como este é o centro do sistema de microsserviços ansiado, é necessário que sua estrutura esteja construída de maneira que defina as APIs de envio e recebimento de informações pelos módulos de serviço e visualização. Importante salientar que uma vez iniciado o desenvolvimento, esse deverá ser flexível o suficiente para se adequar a particularidades dos RPAs, ao mesmo tempo que determina um padrão do fluxo de informações entres os módulos. Assim que forem englobados os vários cenários de processos e automações diferentes, os futuros Robôs desenvolvidos deverão seguir o modelo concebido de leitura e escrita das APIs entre os módulos.

Devido à natureza do cenário, onde é possível ocorrer constantes mudanças em processos, aplicações e/ou RPAs, torna-se crucial que a elaboração do código seja realizada em pequenos módulos, os quais são chamados por um outro maior, responsável por estruturar e organizá-los. Dessa maneira, caso seja necessária uma manutenção, é possível realizá-la apenas em determinados módulos, evitando que interfira em outros processos. A figura 6 exemplifica esse cenário: O processo 1 precisa acessar o Menu A do Módulo 1 para ser executado, contudo esse está em manutenção. Já o processo 2 não faz uso desse módulo, então pode continuar funcionando normalmente. Esse panorama fica mais evidente no caso da necessidade de se reescrever um código, pois é possível reaproveitar todos os outros módulos que o compõem,

reformulando-se apenas o(s) necessário(s). No caso da figura, caso seja necessário reescrever o Módulo 3, é possível fazer o uso dos seus submódulos, contribuindo para um resultado mais rápido.

Figura 6 – Desenvolvimento em módulos



Fonte: Autor.

Visto isso, a estrutura interna de cada parte do sistema de microsserviços deverá seguir um sistema de módulos para que seja possível a realização de manutenções e incrementos sem que isso impacte outros serviços que estejam sendo executados.

6 RESULTADOS

Esse capítulo descreve a estrutura do sistema, bem como o relacionamento entre os módulos dos serviços e o esquema base dos dados (API) a ser transitado entre eles.

6.1 Estrutura do Sistema

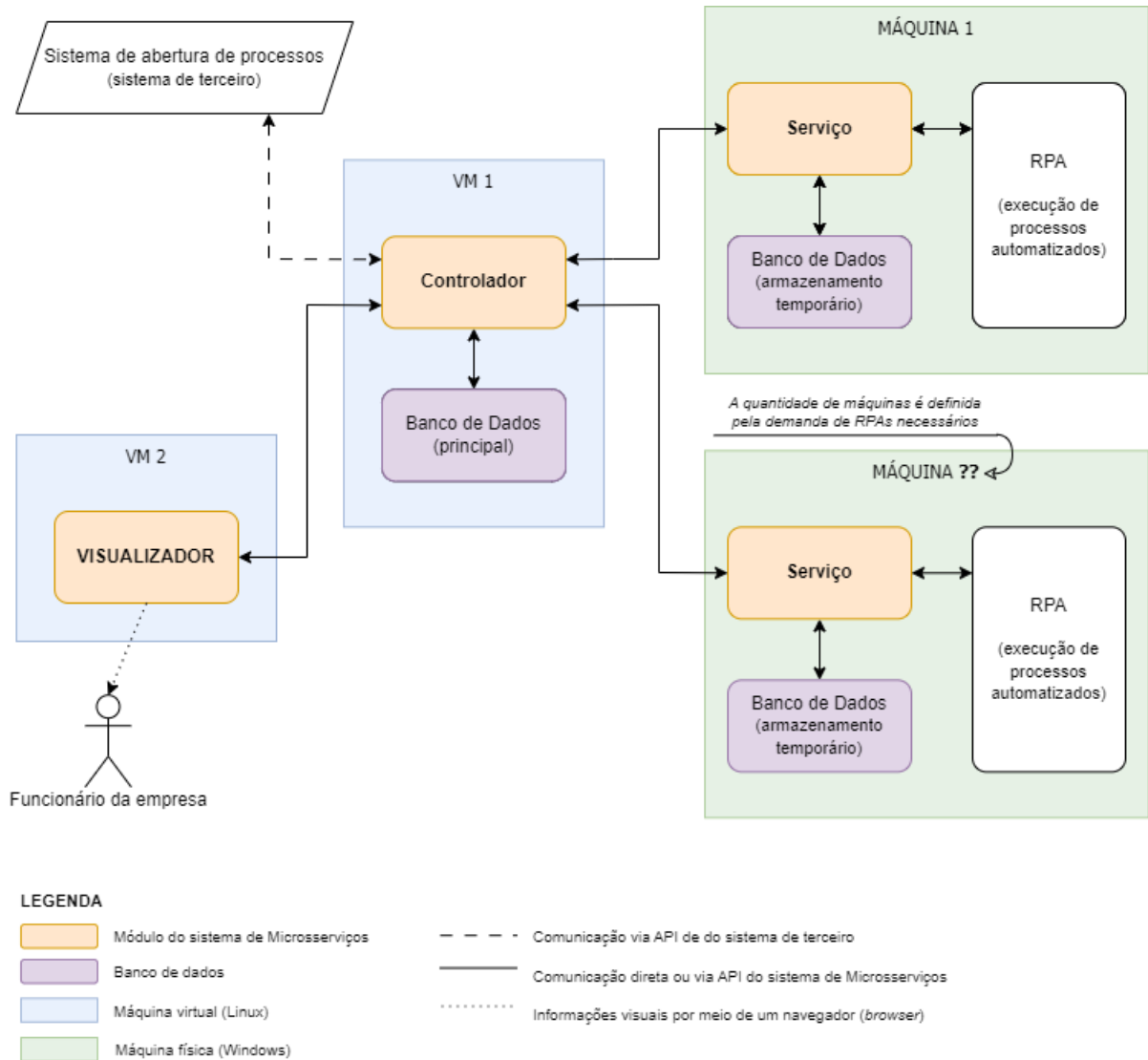
Analisando a estrutura da empresa, bem como os recursos e o atual funcionamento das automações, chegou-se à configuração exibida na figura 7. Com base nela, é possível visualizar os seguintes itens:

- Os módulos Controlador e Visualizador não necessitam estar implementados em uma máquina física, portanto serão alocados em máquinas virtuais dentro do servidor da empresa;
- Haverá um banco de dados central, o qual apenas o Controlador tem acesso;
- O módulo de serviço será implementado em todas as máquinas onde houver um RPA;
- Diferentemente das máquinas dos robôs, os módulos Controlador e Visualizador podem ser implementados em com Linux;
- A comunicação com o sistema responsável pela abertura de processos ficará a cargo do Controlador;
- Cada módulo de serviço terá um bando de dados próprio, que servirá para armazenamento de informações necessárias apenas para o processo corrente.

Devido ao fato do Controlador ser o módulo central, o desenvolvimento deverá começar por ele, pois nele serão definidas as regras de como serão estruturadas as informações de envio e recebimento da API. É nele que as informações serão filtradas para o envio ao módulo de Serviço para cada tipo de operação automatizada. Dito isso, deverá ser necessário o controle das tarefas, tais como, por exemplo: qual será enviada antes, qual está pendente ou concluída e qual robô está livre para recebe-la.

Dentro da empresa existem dois tipos de tarefas que são automatizadas: as que são agendadas e as que se iniciam pelo sistema de abertura de processos. No primeiro caso, podem existir uma tarefa que deve ser executada todo dia 10 e outra todos os dias às 9 horas. Ficará a cargo do Controlador realizar esse agendamento e enviar elas para execução no dia e hora corretos. Outro ponto é que uma tarefa não pode interferir na outra. Se uma não puder ser executada, seja por estar em manutenção ou algum problema no código, isso não pode impedir que outra seja enviada para o módulo de Serviço. Da mesma maneira que as tarefas não podem impactar no envio de informações para o módulo de Visualização. Por esse motivo,

Figura 7 – Configuração dos módulos do sistema de microsserviço

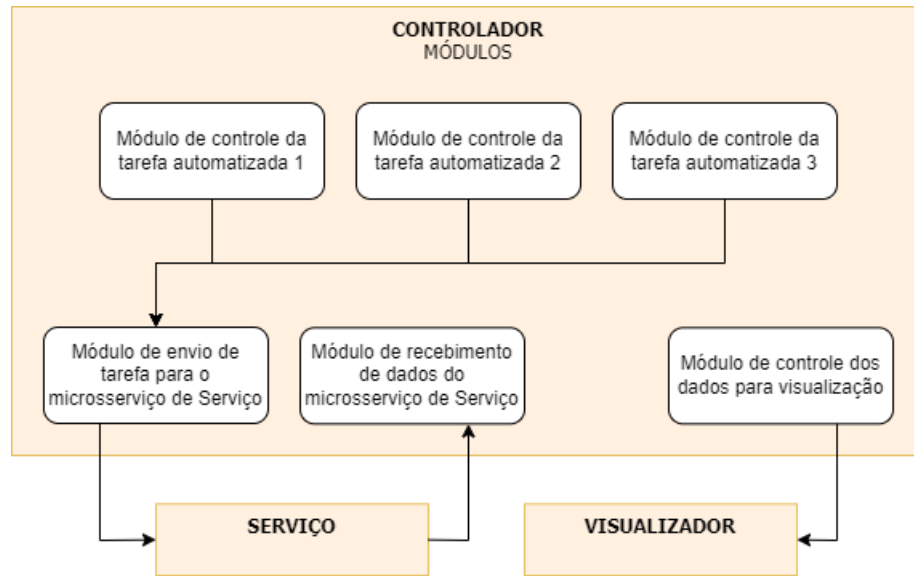


Fonte: Autor.

cada estrutura responsável por uma atividade específica dentro do controlador deverá ser independente das outras, como mostrado na figura 8. Com base nela, se o "módulo de controle da tarefa automatizada 1" estiver em manutenção, não vai impactar nas outras tarefas. O Mesmo pode ser visto para o que envia os dados para a visualização. As exceções estão nos códigos que controlam o envio e o recebimento do módulo de serviço, já que esses são em comum a todas as tarefas. Se caso ele estiver em manutenção, significa que realmente nenhuma tarefa poderá ser enviada ou recebida, por motivo de segurança.

Outro ponto é a comunicação com a aplicação de abertura de processos. Esse programa é um software de terceiro, que possui uma API própria. Para que cada robô não fique acessando ele, gerando tráfego em paralelo e impactando no desempenho do mesmo, optou-se que apenas o Controlador fizesse essa comunicação. Desse modo, ele que acessa e coleta das tarefas designadas para os RPAs, bem como faz o protocolo delas quando forem finalizadas pelos robôs.

Figura 8 – Estrutura do Controlador



Fonte: Autor.

A decisão de centralizar o bando de dados foi tomada para uma melhor organização dos dados. Caso seja necessário informações sobre um determinado tipo de tarefa realizada, não é necessário qual robô fez cada uma e ver o banco de dados deles. Centralizando no Controlador, fica fácil a manutenção e a obtenção de dados, bem como o *backup* deles.

Os módulos de **Serviço** terão como função o controle e filtro de informações enviadas para o RPA, sejam elas vindas do Controlador ou do banco de dados próprio. Por esse motivo, haverá tantos quanto o número de RPAs existentes. Dessa maneira, tem como manter os dados coesos e organizados, além de sempre devolvidos ao controlador ou enviados para o RPA de maneira correta.

Quanto ao banco de dados próprio, cada Serviço terá acesso a um. Isso é necessário para que as informações não sejam perdidas no decorrer de uma automação. Isso pode ser necessário em casos do processo ser muito longo e o RPA precise armazenar dados durante ele, ou na situação em que ocorrer algum problema durante a execução do robô. Ao ser reexecutado, é possível que continue o processo do ponto que parou.

O módulo **Visualizador** terá como função a exibição de informações para a empresa por meio de um navegador. A projeção do seu layout pode ser vista na figura 9. Seus dados virão diretamente do Controlador, o qual vai acessar o BD e enviar as informações requisitadas pelo Visualizador. O tipo e a quantidade de informações disponíveis dependerão da demanda da empresa, e serão organizadas por área e/ou tipo.

Todo o sistema deverá ser de fácil manutenção e ampliação, pois a cada nova automação os módulos deverão ser atualizados para recebe-la. Outro ponto é que deverão ser de fácil migração para o caso de uma máquina der problema. Ocorrendo isso, e havendo outra máquina disponível, os módulos serão implementados nessa nova, já podendo se comunicar entre si com mínimas alterações no código.

Figura 9 – Projeção do layout do módulo Visualizador

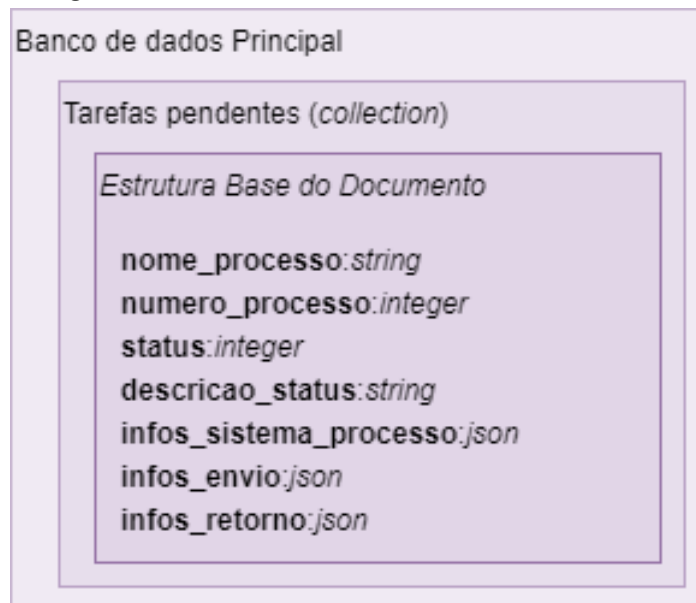


Fonte: Autor.

6.2 Esquema Base da API

Cada automação pode necessitar de diferentes dados para que sejam executadas, bem como podem gerar elementos distintos após sua execução. Devido a isso torna-se essencial que as informações enviadas pela API para Entrada e Saída do módulo de Serviço possam ser dinâmicas. O banco de dados MongoDB colabora com esse cenário, pois os documentos dentro de uma mesma *collection* não precisam serem todos iguais. Na figura 10 é mostrado uma organização base de chaves para os documentos, para que haja a possibilidade de armazenar dados diferentes sem divergir de uma estrutura em comum.

Figura 10 – Estrutura base dos documentos do BD



Fonte: Autor.

Dentre as chaves, três requerem uma descrição:

- **infos_sistema_processo:** contém as informações vindas do sistema de terceiro onde são abertos os processos, tais como hora em que a tarefa foi aberta, quem fez a requisição e para onde deverá ser enviada após a conclusão;
- **infos_envio:** informações relacionadas à tarefa, para que sejam possível ela ser executada. Um exemplo é um processo de contratação de cartão, onde é necessário saber qual a bandeira do cartão.
- **infos_retorno:** os dados que foram gerados pelo RPA após a execução.

Essas três chaves citadas acima são do tipo *json*, o que as tornam dinâmica. Ou seja, é possível modificar o conteúdo delas com base na tarefa que está sendo executada. Isso simplifica os filtros dos dados nos módulos Controlador e de Serviço, ao mesmo tempo que viabiliza a transmissão de diferentes dados entre eles.

7 CONSIDERAÇÕES FINAIS

Neste projeto foi descrito o uso de RPAs e a importância do controle e gerenciamento deles e das informações geradas por eles. Afim de desenvolver um sistema que realize essas funções, foram levantadas informações e, com base nelas, criada uma estrutura de como os módulos dessa aplicação baseada em microsserviços iram se comunicar entre si e a maneira que as informações serão armazenadas. Salienta-se que esse sistema será criado para atender exclusivamente uma empresa do ramo financeiro. Dessa forma, ela não precisará recorrer a aplicações pagas existentes no mercado ao mesmo tempo que terá algo desenvolvido de acordo com as necessidades existentes. Tendo todo essas informações já levantadas, analisadas e organizadas, espera-se que o desenvolvimento do sistema em microsserviços ocorra de maneira fluída.

REFERÊNCIAS

- ANYWHERE, A. **About Us**. 2023. Disponível em: <https://www.automationanywhere.com/company/about-us>. Acesso em: 15 de abril de 2023.
- AUTOMATE, P. **Process Advisor**. 2023. Disponível em: <https://powerautomate.microsoft.com/pt-br/process-advisor/>. Acesso em: 15 de abril de 2023.
- CABRAL ARIEL BEHR, G. S. S. P. H. D. **ROBOTIC PROCESS AUTOMATION: PROPOSTA DE DEFINIÇÃO PARA A TECNOLOGIA NO CONTEXTO CONTÁBIL**. Porto Alegre, 2022. 26 f.
- CXTODAY. **Gartner Says Worldwide RPA Software Spending to Reach \$2.9 Billion in 2022**. 2022. Disponível em: <https://www.cxtoday.com/data-analytics/gartner-magic-quadrant-for-robotic-process-automation-rpa-2022>. Acesso em: 15 de abril de 2023.
- DIVINO, B. **O que é Python?** 2021. Disponível em: <https://www.alura.com.br/artigos/python-uma-introducao-a-linguagem>. Acesso em: 27 de maio de 2023.
- GARTNER. **Gartner Magic Quadrant**. 2023. Disponível em: <https://www.gartner.com.br/pt-br/metodologias/magic-quadrant>. Acesso em: 22 de abril de 2023.
- JÚNIOR, L. M. C. M. **ROBOTIC PROCESS AUTOMATION: PROPOSTA DE DEFINIÇÃO PARA A TECNOLOGIA NO CONTEXTO CONTÁBIL**. [S.l.], 2019. 17 f.
- JÚNIOR, V. G. de F. **Automação de Processos de Negócio Utilizando Robotic Process Automation (RPA) em Um Centro de Serviços Compartilhados (CSC): Um Estudo de Caso**. Uberlândia, 2021. 68 f.
- LIMA, G. **MongoDB: O banco baseado em documentos**. 2021. Disponível em: <https://www.alura.com.br/artigos/mongodb-o-banco-baseado-em-documentos>. Acesso em: 27 de maio de 2023.
- MANZUETO, M. S. **Automação de processos: a influência dos softwares de automação de processos nas rotinas organizacionais**. 2016. 57 f. Dissertação (Mestrado em Administração de Empresas) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2016.
- MARIANO, A. F. **AUTOMAÇÃO ROBÓTICA DE PROCESSOS: UMA ANÁLISE SOBRE A GOVERNANÇA DE RPA PARA GRANDES EMPRESAS**. São Paulo, 2021. 90 p.
- MEDES, I. S. **Arquitetura Monolítica vs Microsserviços: uma análise comparativa**. Brasília, 2021. 90 p.
- SANTOS BRUNA APARECIDA DE OLIVEIRA, V. H. J. M. A. A. M. P. I. F. G. A. C. **Automação industrial em pequenas, médias e grandes empresas: Um estudo teórico**. 2017.
- SANTOS, R. de C. **Estudo Comparativo dos Sistemas Gerenciadores de Bancos de Dados: Oracle, SQL Server e PostgreSQL**. Campolide, 2019. 15 f.
- SILVA, M. C. B. Arthur Marcos da. **Automação Robótica de Processos (RPA): Estudo de Caso Através da Tarefa Administrativa Contas a Pagar**. São Paulo, 2018. 19 f.
- SOUZA, A. A. de. **Organização, processos e tomada de decisão**. Florianópolis, 2021. 109 f.

STAMFORD, C. **Gartner Says Worldwide RPA Software Spending to Reach \$2.9 Billion in 2022**. 2022. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2022-08-1-rpa-forecast-2022-2q22-press-release>. Acesso em: 15 de abril de 2023.

UIPATH. **About Us**. 2023. Disponível em: <https://www.uipath.com/company/about-use>. Acesso em: 15 de abril de 2023.