

Proposta de Automatização, junto ao Instituto Nacional do Seguro Social (INSS) de Santa Maria, de Processos Administrativos de Despacho e Notificação de Benefícios Cancelados ou Suspensos

Guilherme Painko Scalcon, Alexandre Zamberlan
Curso de Ciência da Computação
UFN - Universidade Franciscana
Santa Maria - RS
g.painko@ufn.edu.br, alexz@ufn.edu.br

Resumo—Este projeto tem como foco a automatização e padronização de processos (tarefas) administrativos por meio de robôs de software, especialmente a técnica de *Robotic Process Automation (RPA)*, que executa tarefas repetitivas e suscetíveis a erros humanos em sistemas informatizados. O contexto de aplicação envolve processos do Instituto Nacional do Seguro Social (INSS), incluindo subprocessos manuais relacionados à análise e ao despacho de benefícios, como aposentadorias, auxílios e pensões. Esses subprocessos, até então, são realizados por funcionários servidores e estão sujeitos a falhas. A proposta justifica-se pela necessidade de reduzir erros e tornar o fluxo de trabalho mais eficiente, eliminando atividades manuais repetitivas. O objetivo foi projetar e implementar um sistema capaz de automatizar e padronizar as tarefas de despacho e notificação aos segurados, contribuindo para a melhoria da gestão de benefícios do INSS. Os resultados alcançados foram: i) a modelagem, simulação e análise dos fluxos do INSS por meio da *Business Process Model and Notation (BPMN)*, com o intuito de compreender os fluxos atuais e propor melhorias; e ii) a modelagem e implementação de um sistema Web para atender à automatização das tarefas de despacho e notificação.

Palavras-chave: Robôs de Software; RPA; Padronização; Sistema de Informação.

I. INTRODUÇÃO

O tema de pesquisa deste projeto de Trabalho de Conclusão de Curso (TCC) insere-se no campo da automação e padronização de processos por meio do uso de robôs de software. Esses robôs, em essência, são programas ou *scripts* desenvolvidos para executar tarefas repetitivas, demoradas ou suscetíveis a erros humanos em sistemas informatizados, promovendo maior eficiência e precisão.

Dentre os diversos tipos de robôs de software existentes, destaca-se, para o contexto desta pesquisa, o uso de Robôs de Processos (*Robotic Process Automation - RPA*). Esses robôs são projetados para interagir com a interface de usuário de aplicativos de maneira semelhante à interação humana. Por meio de técnicas de simulação de cliques, preenchimento de formulários, navegação entre telas e até manipulação de dados, os RPA oferecem uma solução eficaz para automatizar processos que antes dependiam exclusivamente de intervenções humanas, garantindo maior produtividade e reduzindo falhas operacionais.

Essa abordagem tecnológica tem ganhado destaque em diversos setores, especialmente em contextos onde a automação de tarefas rotineiras e padronizadas é fundamental para otimizar fluxos de trabalho e reduzir custos. A escolha do RPA para este projeto reflete a busca por soluções inovadoras e alinhadas às necessidades de sistemas informatizados modernos.

O INSS possui diversos fluxos de trabalho, como os processos administrativos relacionados à geração e manutenção (cancelamento ou suspensão) de benefícios, incluindo aposentadoria programada, aposentadoria especial, aposentadoria por incapacidade permanente, auxílio por incapacidade temporária, auxílio-acidente, auxílio-reclusão, salário-maternidade, salário-família e pensão por morte.

Dentro desses fluxos, existem subprocessos realizados manualmente por servidores, tais como: análise de benefícios cancelados ou suspensos; elaboração de despachos ao cidadão segurado, contendo informações como tipo de benefício, erro que ocasionou o cancelamento ou suspensão, Lei Federal infringida, número do benefício, nome do segurado e toda a documentação apresentada pelo segurado em processos anteriores; notificação ao segurado ou a seu representante legal.

Devido à natureza manual, repetitiva e demorada desses subprocessos, bem como à possibilidade de ocorrência de erros por parte dos servidores, este trabalho justifica-se pela proposta de automatizar e padronizar os subprocessos de elaboração de despachos e notificações, visando maior eficiência, precisão e agilidade nos fluxos administrativos do INSS.

O Objetivo Geral é propor o projeto, a implementação de um sistema que automatize e padronize os subprocessos despacho e notificação. Como Objetivos Específicos, assume-se: i) entender o fluxo administrativo de manutenção de benefício junto ao INSS; ii) identificar os modelos de montagem a serem utilizados para a padronização; iii) pesquisar, compilar e testar bibliotecas de geração de *Robotic Process Automation*; iv) criar uma modelagem de um dos processos existentes no INSS utilizando *Business Process Modeling Notation (BPMN)* e fazer simulações de antes e depois

da automatização; v) projetar e implementar um sistema de coleta de informações do processo de manutenção do benefício cancelado; vi) implementar um módulo dentro do sistema para a automatização e padronização por meio de robôs de software.

II. REVISÃO BIBLIOGRÁFICA

Nesta seção, encontram-se explicações sobre os fundamentos necessários para o cumprimento do objetivo deste trabalho. Na Figura 1, é possível visualizar todos os conceitos e suas relações, tendo como foco de trabalho Robôs de Processo executando em um Sistema Web rodando em uma arquitetura-cliente servidor em uma *Local Area Network* (LAN).

A. Automatização

Automação, segundo Frank Lamb [1], é o uso de comandos programáveis e equipamentos mecanizados para substituir atividades manuais que exigem decisões e respostas humanas. Trata-se de uma evolução que vai da mecanização à computação, com o objetivo de reduzir o esforço sensorial e mental, além de otimizar processos e garantir maior assertividade na produtividade. O termo, surgido na década de 1940, descreve sistemas nos quais controles automáticos substituem o trabalho humano [1].

A automação abrange uma ampla gama de tecnologias destinadas a reduzir a intervenção humana em processos, por meio da predeterminação de critérios de decisão e ação, que são incorporados em máquinas ou sistemas computacionais. Ela pode ser encontrada em diversos setores, como linhas de produção industrial, onde são utilizados sistemas de controle com realimentação e robótica para aumentar a produtividade, reduzir custos operacionais e criar vantagens competitivas.

O principal benefício da automação é a melhoria da eficiência e precisão das operações, resultando em menos erros, maior qualidade e economia de mão de obra [2].

B. Robôs de Processos (*Robotic Process Automation - RPA*)

OS Robôs de Processos ou RPA são tecnologias de software projetados para automatizar processos repetitivos e baseados em padrões, substituindo tarefas humanas por atividade de computacionais [3]. Robôs de processo substituem seres humanos em tarefas que são geralmente demoradas e sujeitas a erros manuais, como por exemplo:

- entrada e extração de dados, que envolve copiar e colar informações de um sistema para outro, como preenchimento de formulários, importação de dados de planilhas ou documentos para sistemas do INSS;
- processamento de transações, que trata da execução de operações com base em regras definidas, como aprovações de pedidos, processamento de análise de benefícios juntos ao INSS;
- geração de relatórios e notificações, que envolve compilar dados de várias fontes (base de dados de benefícios)

para gerar relatórios automáticos e enviar notificações ou despachos para segurados do INSS;

- validação de dados, que trata de Verificar consistência e precisão de informações em sistemas distintos, destacando dados incorretos ou incompletos para revisão humana, quando necessário;
- rotinas de monitoramento e auditoria, que foca no monitorar processos e realizar auditorias automatizadas para garantir conformidade e identificar falhas.

No contexto do INSS, por exemplo, robôs podem automatizar a análise de benefícios, a elaboração de despachos e notificações ao segurado, tarefas que atualmente requerem tempo dos servidores públicos.

C. Ferramentas para RPA

As ferramentas apresentadas fazem parte do ecossistema Python, já que o sistema Web (LAN) foi projetado e desenvolvido utilizando a linguagem Python e os *frameworks* Django e Bootstrap.

1) *PyAutoGUI*: De acordo com seu site de referência [4], essa ferramenta é um módulo Python de automação baseado em Interface Gráfica (GUI) e multiplataforma para usuários de sistemas informatizados. Usado para controlar o mouse e o teclado de forma automatizada [5], *PyAutoGUI* auxilia na automatização de interações com outros aplicativos, sejam eles Web ou não. Está disponível no *Package Installer for Python* (PIP), que por sua vez, é um gerenciador de pacotes para o ecossistema Python [6]. Registra-se que um pacote contém todos os arquivos que são necessários para a implementação de um programa ou um módulo [6]. O PIP é um recurso que possibilita a instalação facilitada de pacotes via terminal ou ambiente *shell*, por meio do comando: `pip install pyautogui`.

2) *SikuliX*: O *SikuliX* é uma ferramenta de automação visual, que por meio de reconhecimento de imagens, permite automatizar tarefas no computador [7]. A ferramenta é considerada (*WYSIWYS-Tool: What You See Is What You Script*), ou seja, o que o programador visualiza é o que ele codifica [7]. Para o *SikuliX* encontrar a imagem na tela, é necessário o uso da biblioteca *OpenCV* (também presente no ecossistema Python), pelo motivo dele ser baseado no método *matchTemplate* do *OpenCV*. Para utilizar imagens com os recursos do *SikuliX*, é necessário realizar o armazenamento de imagem, preferencialmente no formato PNG, em algum local do sistema.

3) *Robocorp*: Plataforma que oferece ferramentas e infraestrutura para RPA, com foco em automação a partir de robôs que agilizam tarefas repetitivas em ambientes empresariais [8]. O *Robocorp* permite a criação, gerenciamento e implementação de robôs capazes de realizar vários processos como, processamento de dados, integração de sistemas, coleta de itens de páginas Web, entre outras funcionalidades possíveis de implementação. É possível utilizar escrever funções em Python e utilizar dentro dos *scripts* nos *frameworks*

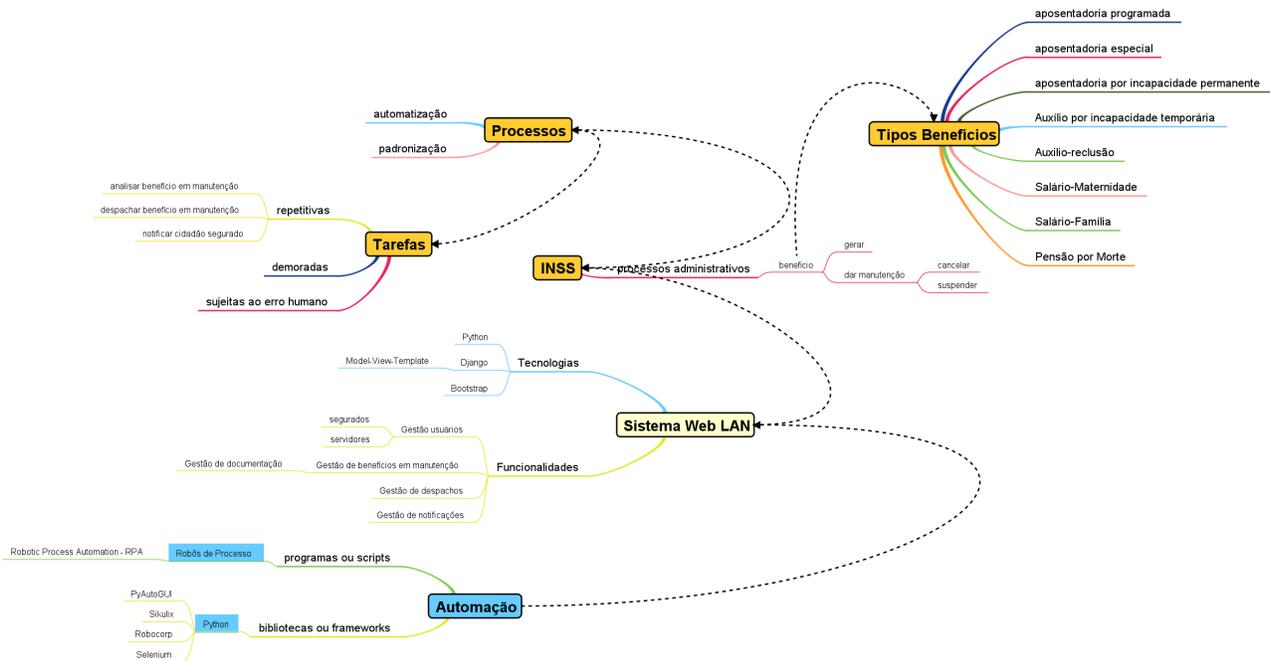


Figura 1. Mapa mental do trabalho.

utilizando ambientes virtuais, os quais, podem facilitar as configurações dos robôs criados utilizando Robocorp [9].

4) *Selenium*: É uma biblioteca de automação suportada por diversas linguagens como JavaScript, Python, C++, HTML (por meio de um *driver* de navegador) [10]. No caso do contexto deste trabalho, ela permite emular interações humanas com um sistema *desktop* ou em página Web. Segundo Palharini e Zamberlan [10], Selenium é útil quando se precisa extrair e/ou injetar elementos em sistemas, como sites que são dinâmicos e interativos, como páginas geradas por JavaScript. O Selenium pode ser usado para simular cliques, preenchimento de formulários, *scroll*, entre outras interações [11].

D. Desafios do uso de Robôs

Dentro da implementação de robôs, existem desafios que podem surgir, tais como:

- Processos ineficientes replicados por automação: Caso um fluxo de processo predefinido seja baseado em processos que contêm erros ou ineficiências, os robôs replicarão essas falhas, o que pode gerar custos adicionais e desperdício de recursos [12];
- Interação humano-robô e gestão de exceções: A gestão de exceções limita a autonomia dos robôs, pois o usuário pode precisar intervir em situações que exigem cognição, intuição ou decisões situacionais [12];
- Integração de sistemas e robustez da infraestrutura: Para uma automação eficiente, a infraestrutura tecnológica subjacente deve ser robusta e plenamente compatível

com as soluções de RPA [12];

- Governança organizacional e estratégia: Estruturas de governança podem tornar-se mais complexas com a adoção da RPA. Cada organização deve decidir se a RPA será usada como uma solução temporária ou como uma parte integrante de sua capacidade estratégica [12];
- Impactos no trabalho humano e necessidade de requalificação: As iniciativas estratégicas para implementar a RPA devem levar em consideração o engajamento dos trabalhadores, o desenvolvimento de novas competências e as decisões relacionadas à alocação de recursos humanos [12];
- Evolução tecnológica e integração de IA: A evolução da RPA inclui o uso de módulos de inteligência artificial, permitindo a análise de mudanças nos processos e funcionalidades dos robôs, bem como transformações no mercado da RPA [12].

E. Framework Django e o modelo arquitetural Model-View-Template - MVT

O MVT do Django é o padrão arquitetural, uma variação do padrão MVC (*Model-View-Controller*) adaptada ao Django para organizar e separar a lógica de uma aplicação Web. No MVT [13]:

- Model representa a camada de dados, onde são definidos os modelos de dados que mapeiam para as tabelas do banco, permitindo criar, ler, atualizar e deletar dados;
- View é responsável pela lógica de processamento dos

dados e pela resposta que o usuário visualizará. Ela recebe as requisições HTTP, processa dados, interage com os modelos (caso necessário) e retorna uma resposta, que geralmente inclui renderizar uma página HTML;

- Template é a camada de apresentação, onde o conteúdo é estruturado para exibição ao usuário final. Ele utiliza o sistema de *template* do Django para definir a estrutura HTML e permite o uso de variáveis e lógica de apresentação, como repetições e condicionais, para exibir dados dinâmicos.

Esse padrão ajuda a manter o código organizado e permite que cada camada seja independente, facilitando o desenvolvimento, a manutenção e a escalabilidade do sistema [13].

F. Universo INSS e os fluxos de manutenção de benefícios

De acordo com o próprio site [14], o Instituto Nacional do Seguro Social (INSS) foi criado em 27 de junho de 1990, por meio do Decreto número 99.350, a partir da fusão do Instituto de Administração Financeira da Previdência e Assistência Social (IAPAS) com o Instituto Nacional de Previdência Social (INPS), como uma autarquia vinculada ao então Ministério da Previdência e Assistência Social (MPAS). O INSS possui um extenso fluxo de processos administrativos, que incluem a geração, manutenção (como cancelamento ou suspensão) e análise de benefícios. Esses processos envolvem várias etapas realizadas manualmente, como análise de benefícios, coleta de documentos, criação de despachos (indicando o tipo de benefício, o erro que gerou o cancelamento ou suspensão, entre outros detalhes), e elaboração de ofícios e notificações para defesa em casos de possível fraude.

Dado que esses processos são manuais, repetitivos, demorados e suscetíveis a erros, este trabalho justifica-se pela proposta de automatizar e padronizar os subprocessos de despacho e notificação, buscando maior eficiência e precisão.

G. Business Process Model and Notation - BPMN

BPMN é um padrão internacional utilizado na modelagem de processos de negócio [15]. Ele fornece uma notação gráfica, de fácil entendimento, para representar o fluxo de atividades e decisões dentro de um processo, sendo útil tanto para profissionais técnicos quanto para gestores de negócios. Sua principal finalidade é criar um entendimento comum sobre como os processos de uma organização funcionam, facilitando a comunicação entre as partes interessadas. De acordo com Von Rosing e colegas [15], a notação do BPMN utiliza elementos como eventos, atividades, *gateways* e fluxos, organizados de maneira padronizada, permitindo uma visualização clara e objetiva do processo, desde seu início até a conclusão.

Na prática, o BPMN é aplicado para identificar, analisar e otimizar processos organizacionais, promovendo melhorias na eficiência. Ele permite mapear processos com-

plexos, identificar gargalos e simular mudanças antes de implementá-las. Além disso, é frequentemente utilizado como base para automatização de processos por meio de ferramentas de BPM (Gestão de Processos de Negócio) [15].

H. Trabalhos Relacionados

Esta seção tem como objetivo elencar e discutir alguns trabalhos que possuem tecnologia semelhante a tecnologia utilizada na solução ou por possuírem contexto técnico deste estudo.

O trabalho desenvolvido por Hoffmann e colegas [12], pesquisou a *Robotic Process Automation* (RPA) como uma ferramenta eficaz no âmbito das transformações digitais. Através de análise de ferramentas, esses mesmos autores identificaram quatro características principais que definem o RPA: automação de processos previamente realizados por humanos, utilização de módulos tecnológicos e operadores de fluxo de controle, integração com ecossistemas de Tecnologia de Informação (TI) estabelecidos e implementação ágil através de projetos. O estudo destacou os benefícios da RPA, como aumento de eficiência, escalabilidade e conformidade, ao mesmo tempo em que apontou os desafios relacionados à governança e à necessidade de integração estratégica para maximizar seu impacto organizacional. Além disso, os autores sugerem que o RPA pode evoluir para formas mais avançadas de automação, especialmente quando combinado com tecnologias de inteligência artificial.

Já o trabalho de Palharini e Zamberlan [10] apresenta a construção de um sistema automatizado de rastreamento de conteúdo em sites e/ou portais jurídicos. O sistema utiliza um *crawler* (bot de pesquisa) para coletar metadados, como título, descrição e data de publicação. Ele foi desenvolvido para facilitar a busca de informações por profissionais do Direito nos Tribunais de Justiça do Brasil. O projeto empregou a metodologia Scrum juntamente da técnica Kanban para gestão de atividades, a linguagem Python, os *frameworks* Django, Bootstrap e Scrapy, a biblioteca Beautiful Soup e a API Selenium. Como resultado há um sistema piloto disponível em <https://buscadir.lapinf.ufn.edu.br/> com gestão de usuários, áreas do Direito, fontes de pesquisa, instituições e buscas.

Com o aumento da competitividade no mercado e o avanço tecnológico, a gestão de processos de negócio tem se tornado cada vez mais relevante. Nesse contexto, o trabalho de Richetti [16] apresenta um estudo de caso focado na otimização de processos por meio de técnicas de modelagem e simulação, utilizando a notação *Business Process Model and Notation* (BPMN). O projeto buscou aprimorar a qualidade e eficiência das atividades relacionadas à prestação de serviços da empresa. Todo o processo de modelagem e simulação foi realizado no software Bizagi Modeler, destacando o potencial da BPMN para alcançar melhorias significativas no desempenho operacional.

I. Considerações sobre Revisão Bibliográfica

Com base na revisão realizada, observa-se que a ferramenta SikuliX não atende ao escopo deste projeto, uma vez que ele se baseia na manipulação de textos. Apesar de o PyAutoGUI ser uma ferramenta poderosa para automação, sua aplicação é mais adequada a tarefas de curto prazo e de menor complexidade. Em sistemas Web de longo prazo, soluções que utilizam APIs ou ferramentas de RPA dedicadas, como Selenium ou plataformas específicas de automação, tendem a ser mais robustas e confiáveis.

Para automatizar processos envolvendo manipulação de textos em formatos .doc ou .docx em um sistema Web desenvolvido com Python-Django, a escolha entre Robocorp e Selenium depende das necessidades específicas de automação. O Selenium é ideal para automações relacionadas à interação com navegadores, como preenchimento de formulários, cliques em botões e navegação entre páginas. Contudo, ele não oferece suporte nativo para manipulação de arquivos .doc ou .docx.

Por outro lado, o Robocorp é uma solução de RPA mais abrangente, capaz de realizar automações que envolvem interfaces de usuário, manipulação de arquivos e integração com diversos sistemas. Utilizando a biblioteca RPA.Document da Robocorp, é possível extrair, ler e modificar documentos do Microsoft Word de maneira direta e eficiente, tornando-o uma escolha mais apropriada para o escopo deste projeto.

III. METODOLOGIA DO TRABALHO

Este trabalho caracteriza-se como uma pesquisa aplicada, voltada para a resolução de um problema prático: a automatização e padronização de dois processos junto ao INSS. A pesquisa é de natureza quantitativa, pois envolve a coleta e análise de dados numéricos referentes aos processos existentes e ao desempenho do sistema proposto.

Em relação à gestão da pesquisa, a metodologia *Scrum* [17] adaptada é utilizada, bem como a técnica *Kanban* para a organização das atividades assumidas no cronograma. Por fim, as ferramentas ou materiais a serem utilizados são: Trello; GitHub; Visual Studio Code; Linguagem Python; *framework* Django; biblioteca Robocorp. Além disso, a ferramenta Bizagi é utilizada para a análise e compreensão dos processos que ocorrem no INSS.

No contexto de Scrum, há elementos básicos como papéis, eventos, artefatos e as regras que os conectam. Dessa forma, o *product owner* é o funcionário servidor responsável em gerenciar benefícios em manutenção, pois ele que vai priorizar os itens no *product backlog* (lista das funcionalidades mapeadas e priorizadas). O *scrum master* é o professor orientador, uma vez que atua como facilitador do Scrum, garantindo que os envolvidos no desenvolvimento do sistema sigam as boas práticas da metodologia. O *dev team* é formado pelo aluno e pelo professor orientador, que trabalham em par no

desenvolvimento do sistema, de forma que os itens do *sprint backlog* sejam completados.

Em relação aos eventos da metodologia Scrum, há os *sprints*, que são semanais (máximo 15 dias), em que um item do *product backlog* deve ser apresentado e avaliado pelo cliente. Ao final de cada entrega, um *sprint planning* é realizado, para o próximo item do *product backlog*, bem como um *sprint review*, garantindo os encaminhamentos futuros tendo como base as autorizações do *scrum master* e do *product owner*.

A. Proposta do sistema

O sistema proposto é desenvolvido com tecnologias do ecossistema Python, integrando Bootstrap e Django para a construção de sua interface e estrutura. Seu principal objetivo é automatizar a geração de despachos utilizando robôs de processos, evitando erros e otimizando tarefas repetitivas.

O funcionamento do sistema prevê que o funcionário servidor responsável insira os dados necessários para a criação do despacho. Com base nessas informações, o sistema deve gerar automaticamente o despacho e a notificação para o beneficiário, incluindo toda a documentação exigida. A única etapa manual restante é a impressão e o envio da notificação ao beneficiário por parte do funcionário servidor, simplificando o fluxo de trabalho e garantindo maior precisão e eficiência no processo.

B. Modelagem da Solução

A metodologia Scrum sugere que a modelagem e implementação do solução seja a partir do mapeamento de aspectos funcionais do sistema, facilitando assim o entendimento das funções básicas dos processos existentes, suas relações e suas interdependências. Esse entendimento, tanto pela equipe de desenvolvimento, quanto pelo cliente. Dessa forma, a Figura 2, apresenta dois pacotes (Gestão básica e Módulo de automatização), que, por sua vez, são compostos por funcionalidades, como Gerenciar benefício em manutenção, gerenciar despacho (tipo), gerar texto padrão, compilar documentos e gerar notificação, por exemplo.

Por sua vez, a Figura 3 ilustra o fluxo de um benefício que recebeu uma denúncia ou que foi identificado irregularidade no processo, até a notificação da irregularidade ao beneficiário.

Finalmente, a Figura 4 contempla questões de aspectos estruturais do sistema, em que é possível organizar o sistema, via o modelo MVT, em pacotes, classes (com os principais métodos, coerentes ao diagrama de casos de uso da Figura 2) e suas inter-relações e/ou dependências. Na figura, destaca-se o pacote *models* que possui as classes: Benefício, Tipo-Benefício, Documentacao, Despacho, Usuario, TipoUsuario e Aviso).

O diagrama de caso de uso, apresentado na Figura 2, detalha um sistema de Gestão Básica com integração de

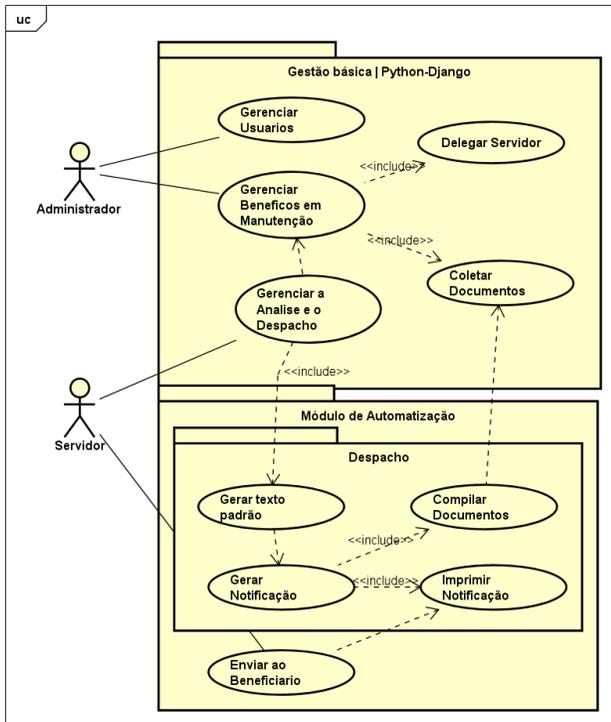


Figura 2. Funcionalidades básicas do sistema Web e do módulo de automação, em Diagrama de Casos de Uso.

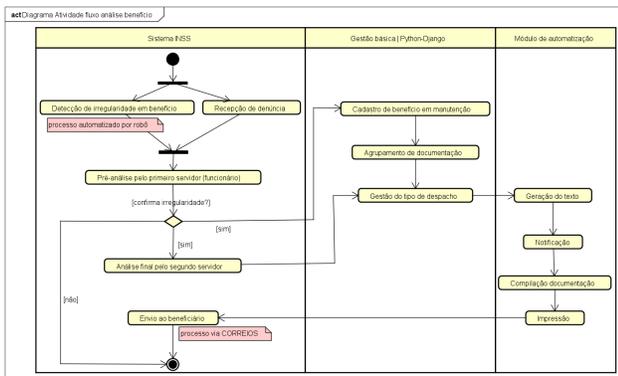


Figura 3. Fluxo principal, desde a chegada de benefício em manutenção, até a notificação aos beneficiários, em Diagrama de Atividade.

um Módulo de Automação, implementado em Python com Django. Segue uma descrição dos principais elementos e interações:

- Os atores do sistema são o administrador e o servidor. O administrador é responsável por tarefas de gerenciamento, incluindo a administração de usuários, que envolve cadastrar, alterar ou remover usuários no sistema, e a gestão dos benefícios em manutenção, supervisionando e ajustando-os para garantir o funcionamento adequado. Já o servidor atua diretamente nos processos de automação, com a função de analisar os

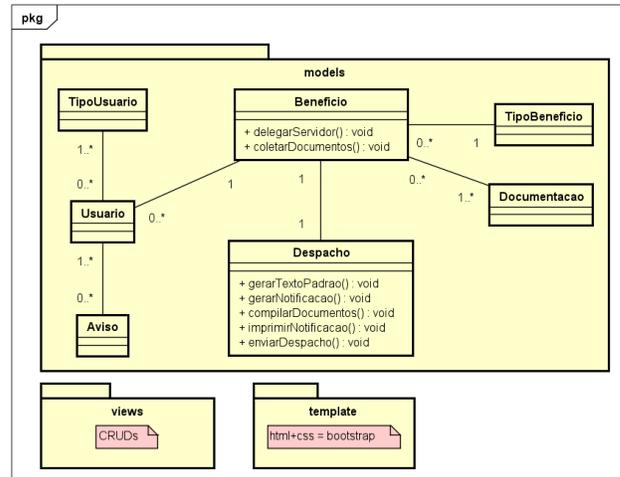


Figura 4. Aspectos estruturais do sistema, em Diagrama de Domínio.

despachos gerados pelo sistema automatizado, assegurando a precisão das informações e seu envio correto aos beneficiários;

- O caso de uso do sistema está dividido em duas áreas principais: Gestão Básica e Módulo de Automação. Na Gestão Básica, o administrador pode Gerenciar Usuários, realizando operações como cadastro, alteração e exclusão de usuários, e Delegar Servidores, atribuindo benefícios em manutenção para análise aos servidores. Além disso, o administrador tem acesso à funcionalidade de Gerenciar Benefícios em Manutenção, incluindo o benefício no sistema. No Módulo de Automação, os processos são automatizados para aumentar a eficiência. Esse módulo inclui a funcionalidade de Gerar Texto Padrão, que cria documentos ou notificações com base em modelos predefinidos, e Gerar Notificação, que também pode incluir o processo de Imprimir Notificação. Outras atividades automatizadas incluem Compilar Documentos, organizando os arquivos necessários, e Enviar ao Beneficiário, garantindo que o despacho automatizado seja entregue ao destinatário final. Essas funcionalidades são interligadas para assegurar que tanto as operações manuais quanto as automáticas funcionem de forma integrada e eficiente.

O diagrama de domínio, apresentado na Figura 4, descreve a estrutura principal do sistema, abordando as classes e seus relacionamentos. No pacote *models*, existem classes centrais do sistema. A classe *TipoUsuario* está associada a *Usuario*, permitindo que um usuário tenha um ou mais tipos definidos (administrador ou servidor). A classe *Usuario*, por sua vez, pode estar associada a múltiplos *Avisos*, que representam mensagens ou notificações relevantes. Cada *Usuario* pode estar relacionado a um ou mais *Benefícios*, sendo que cada *Beneficio* está obrigatoriamente associado

a exatamente um *Usuário*. A classe *Beneficio* possui dois métodos principais: *delegarServidor* e *coletarDocumentos*, que descrevem suas funcionalidades. Ela também está relacionada a *TipoBeneficio*, representando diferentes categorias de benefícios. Além disso, *Beneficio* está vinculado a uma ou mais instâncias da classe *Documentacao*, que agrupam os documentos necessários para o benefício. A classe *Despacho* complementa as operações relacionadas aos *Benefícios*, oferecendo funcionalidades por meio de métodos como *gerarTextoPadrao*, *gerarNotificacao*, *compilarDocumentos*, *imprimirNotificacao* e *enviarDespacho*. Cada *Despacho* está associado exclusivamente a um único *Beneficio*.

Complementando a estrutura, o pacote *views* é responsável pelas implementações das operações de *Create*, *Read*, *Update*, *Delete* (CRUD) que gerenciam os dados do sistema, enquanto o pacote *template* é responsável pela interface de usuário, com a utilização de tecnologias como HTML, CSS e Bootstrap, garantindo *design* moderno e responsivo.

C. Modelagem de Processo

A modelagem de processos é uma etapa fundamental para analisar e melhorar processos de negócios, especialmente em contextos administrativos como despacho e notificação de benefícios cancelados ou suspensos. Neste trabalho, utilizou-se a ferramenta Bizagi Modeler com a metodologia BPMN, a qual permite uma representação gráfica clara das etapas, decisões e interações entre atividades [15].

Em relação a dinâmica da modelagem de processos, é preciso destacar que há a representação gráfica das atividades de um processo de negócio (no caso, processos administrativos de despacho e notificação de benefícios cancelados ou suspensos), identificando as etapas, decisões e interações entre elas. A modelagem segue a metodologia BPMN, que define os elementos gráficos e a forma de conectar as etapas. Registra-se que há passos dessa modelagem: i) definir o escopo do processo; ii) identificar *stakeholders*.

Inicialmente, o escopo do processo trata do benefício que recebeu denúncia ou foi identificada alguma irregularidade. Como *.stakeholders* principais, participam dois servidores públicos: um com função de chefia, responsável pela análise e aprovação final, e outro servidor ordinário, encarregado das etapas operacionais iniciais.

Na Figura 5, apresenta-se o diagrama de processo modelado com a ferramenta Bizagi Modeler, sem a implementação do módulo de automatização. O fluxo está organizado em duas linhas principais: a primeira linha refere-se ao Sistema INSS, abrangendo as etapas realizadas por sistemas institucionais ou servidores responsáveis pela recepção e detecção de irregularidades; e a segunda linha representa as atividades conduzidas pelo servidor responsável pela análise do benefício, incluindo a definição do tipo de processo, coleta e seleção de documentos, geração de texto e impressão. Além disso, o diagrama contempla eventos temporizados e pontos de decisão

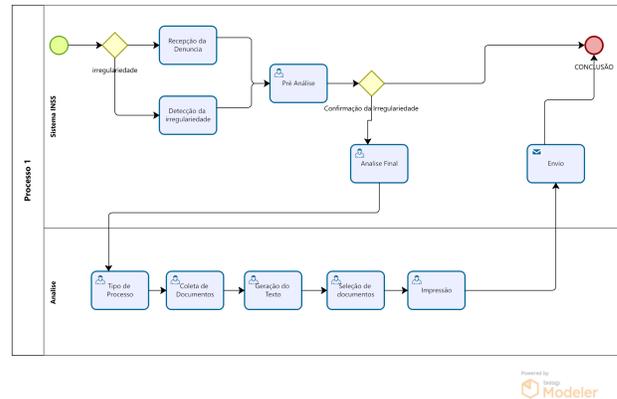


Figura 5. Modelagem de processo via Bizagi Modeler.

Na Figura 6, apresenta-se o diagrama de processo modelado com a ferramenta Bizagi Modeler, que inclui a presença de um módulo de automatização. O processo está dividido em três linhas (*lanes*) principais: a primeira linha refere-se ao Sistema INSS, englobando as etapas realizadas por outros sistemas institucionais ou por servidores responsáveis por detectar e encaminhar irregularidades; a segunda linha diz respeito às atividades executadas no sistema desenvolvido em Python com Django, sendo de responsabilidade direta do servidor que atua na gestão do benefício; e a terceira linha representa as ações automatizadas, que serão realizadas pelo robô, como geração de texto, envio de notificações e compilação de documentos. O diagrama também contempla eventos temporizados e decisões.

D. Infraestrutura

A infraestrutura necessária para a implementação do sistema proposto engloba alguns aspectos.

O sistema foi implementado em um ambiente baseado no sistema operacional Windows, pois este é o sistema operacional utilizado no INSS. Essa escolha garante compatibilidade com o ambiente corporativo já existente, além de facilitar a integração e a manutenção do sistema pelos profissionais responsáveis. O Windows também oferece suporte para ferramentas de desenvolvimento, servidores Web e gerenciamento de permissões.

O sistema utiliza o banco de dados SQLite para armazenar todas as informações relevantes, incluindo os usuários cadastrados e os benefícios em manutenção que estão em análise. Essa escolha garante uma organização eficiente dos dados, além de proporcionar acessibilidade e facilidade na consulta e manipulação das informações, atendendo às necessidades do sistema de forma prática e confiável.

O sistema obedece o modelo cliente-servidor, no qual a camada cliente (*frontend*) é responsável pela interação com o usuário final, utilizando *frameworks* como o Bootstrap

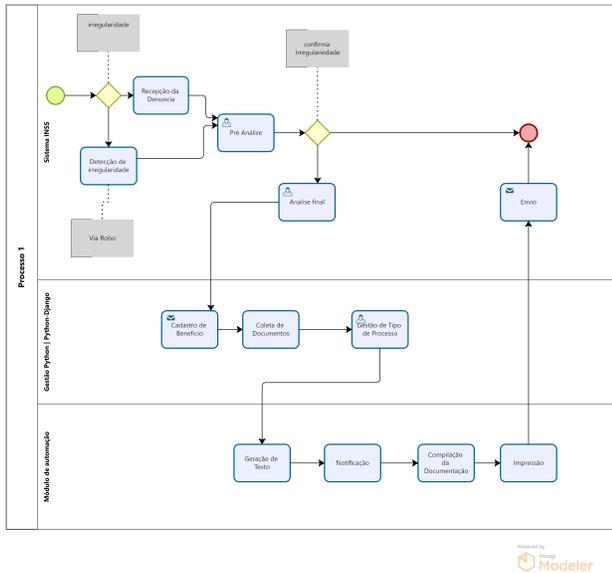


Figura 6. Modelagem de processo via Bizagi Modeler.

para responsividade. O *backend* é gerenciado pelo Django, rodando em servidores locais configurados no Windows, conforme a demanda. Essa arquitetura modular facilita a manutenção e a escalabilidade do sistema.

IV. RESULTADOS

Como resultados, pode-se destacar a análise das simulações comparativas dos dois cenários: sem a presença do módulo automatizado e com a implementação desse módulo. Além disso, todas as interfaces de interação com usuário e com o próprio módulo de automatização.

A. Resultados da Simulação Bizagi

A Figura 7 apresenta os resultados da simulação realizada na ferramenta Bizagi, considerando o processo sem a implementação do módulo de automatização. A simulação foi executada com 1.000 instâncias de processo, resultando em um tempo mínimo de 1 hora, tempo máximo de 3 horas e 39 minutos, e um tempo médio de 3 horas, 3 minutos e 3 segundos por instância. Com base nesses dados, estima-se que o tempo total necessário para concluir a análise das 1.000 instâncias foi de aproximadamente 127 dias, 3 horas e 6 minutos.

A Figura 8 apresenta os resultados da simulação realizada na ferramenta Bizagi, considerando o processo com a presença do módulo de automatização. A simulação foi executada com 1.000 instâncias de processo, evidenciando um tempo mínimo de 1 hora e 5 minutos, um tempo máximo de 3 horas e 16 minutos e um tempo médio de 2 horas, 14 minutos e 33 segundos por processo. Com base nesses dados, estima-se que o tempo total necessário para concluir

Informações do cenário
 Nome: Cenário 1
 Unidade de tempo: Minutos
 Duração: 03/03/2020

Nome	Tipo	Instâncias concluídas	Instâncias iniciadas	Tempo mín.	Tempo máx.	Tempo méd.	Tempo total
Processo 1	Processo	1.000	1.000	1h	3h 39m	3h 3m 3s	127d 3h 6m
Irregularidade	Gateway	1.000	1.000				
Recepção de Denúncia	Atividade	505	505	0	0	0	0
Detecção de Irregularidade	Atividade	495	495	0	0	0	0
Pré-Análise	Atividade	1.000	1.000	1h	1h	1h	41d 16h
Confirmação de Irregularidade	Gateway	1.000	1.000				
CONCLUSÃO	Finalizar evento	1.000					
Análise Final	Atividade	774	774	1h 52m	1h 52m	1h 52m	60d 4h 48m
Tipo de Processo	Atividade	774	774	10m	10m	10m	5d 9h
Coleta de Documentos	Atividade	774	774	10m	10m	10m	5d 9h
Geração de Texto	Atividade	774	774	10m	10m	10m	5d 9h
Seleção de documentos	Atividade	774	774	10m	10m	10m	5d 9h
Impressão	Atividade	774	774	2m	2m	2m	1d 1h 48m
Envio	Atividade	774	774	5m	5m	5m	2d 16h 30m
NoneStart	Iniciar evento	1.000					

Figura 7. Resultado da simulação sem o módulo de automatização.

a análise de todas as instâncias foi de aproximadamente 93 dias, 10 horas e 41 minutos.

Informações do cenário
 Nome: Cenário 1
 Unidade de tempo: Minutos
 Duração: 03/03/2020

Nome	Tipo	Instâncias concluídas	Instâncias iniciadas	Tempo mín.	Tempo máx.	Tempo méd.	Tempo total
Processo 1	Processo	1.000	1.000	1h 5m	3h 16m	2h 14m 33s	93d 10h 41m
NoneStart	Iniciar evento	1.000					
ExclusivoGateway	Gateway	1.000	1.000				
Recepção de Denúncia	Atividade	488	488	5m	5m	5m	1d 16h 45m
Detecção de Irregularidade	Atividade	511	511	5m	5m	5m	1d 18h 35m
Pré-Análise	Atividade	1.000	1.000	1h	1h	1h	41d 16h
ExclusivoGateway	Gateway	1.000	1.000				
Cadastro de Benefício	Atividade	531	531	5m	5m	5m	1d 20h 15m
Coleta de Documentos	Atividade	531	531	2m	2m	2m	17h 42m
Gestão de Tipo de Processo	Atividade	531	531	2m	2m	2m	17h 42m
Análise Final	Atividade	531	531	1h 52m	1h 52m	1h 52m	41d 7h 12m
NoneEnd	Finalizar evento	1.000					
Geração de Texto	Atividade	531	531	2m	2m	2m	17h 42m
Notificação	Atividade	531	531	2m	2m	2m	17h 42m
Completção de Documentação	Atividade	531	531	2m	2m	2m	17h 42m
Impressão	Atividade	531	531	2m	2m	2m	17h 42m
Envio	Atividade	531	531	2m	2m	2m	17h 42m

Figura 8. Resultado da simulação com o módulo de automatização.

Com base na análise dos resultados obtidos nas simulações dos dois modelos — o primeiro sem o módulo de automatização e o segundo com sua implementação —, observa-se uma melhoria de desempenho aproximada de 26%, o que representa um ganho significativo em termos operacionais. A adoção do módulo de automatização não apenas pode reduzir a carga de trabalho e a responsabilidade dos funcionários servidores, como também possibilita o registro formal e estruturado do fluxo de trabalho e do histórico temporal de cada processo dentro do sistema. Essa automação tende a minimizar falhas humanas, promovendo maior eficiência, precisão e rastreabilidade nos procedimentos de despacho. Finalmente, registra-se que isso é uma projeção de tempo e desempenho, caso o módulo tenha 100% de eficácia em situação real.

B. Sistema Web

Decidiu-se apresentar e discutir algumas interfaces, juntamente com seus respectivos códigos, a fim de ilustrar

o sistema Web construído integrado com o módulo de automatização.

Na Figura 9, é apresentada a página responsável pelo controle de despachos (criação, listagem, edição e exclusão). Nessa página, encontra-se o botão “Cadastrar despacho”, que ao clicá-lo, o usuário é direcionado ao formulário de cadastro de despacho. Outra funcionalidade é, ao clicar no botão de “Arquivo despacho”, o sistema gera um arquivo pdf com o despacho.

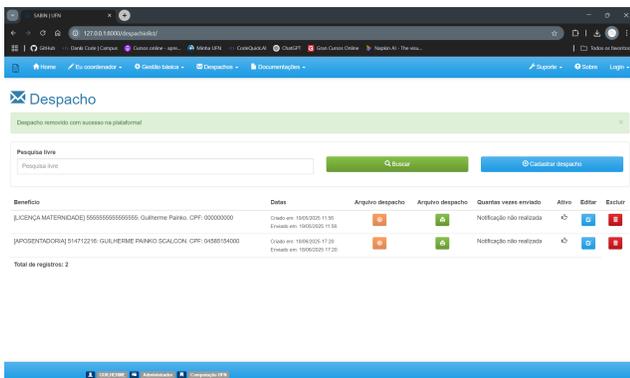


Figura 9. Página despacho.

Ao ser direcionado para a página de cadastro, Figura 10, o usuário deve selecionar o benefício desejado, definir o status dele e clicar em gravar, isso fará executar o robô responsável pela coleta dos documentos.

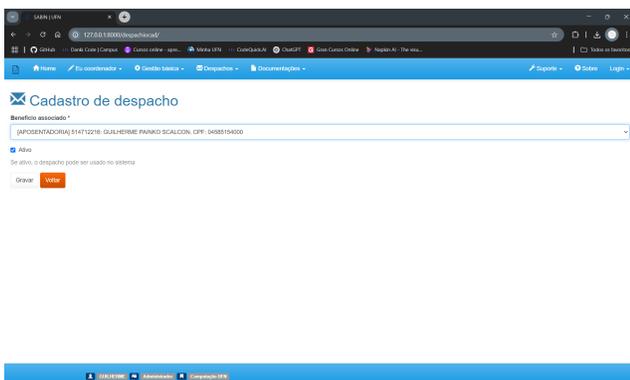


Figura 10. Cadastro de despacho.

Na figura 11, é mostrado o código responsável por definir a tarefa de coleta de documentos, definindo os sites que o robô deve visitar para coletar os arquivos para o despacho.

A Figura 12 possui o código onde é definido os sites que o robô deve buscar para realizar o *download* dos documentos. Neste ponto, registra-se que os sites utilizados são meramente de demonstrativos e que geram arquivos pdfs de forma automatizada para esses tipos de testes e avaliações.

A Figura 13 apresenta o código onde é montado o texto do despacho para *download*.

```
class DespachoCreateView(LoginRequiredMixin, StaffRequiredMixin, CreateView):
    model = Despacho
    fields = ['beneficio', 'is_active']
    success_url = 'despacho_list'

    def form_valid(self, form):
        # Etapa Executa robô Sabin via RCC (Robocorp)
        try:
            sabin_path = os.path.abspath(os.path.join(os.path.dirname(__file__), "../projeto/sabin/"))
            subprocess.run(
                ["rcc", "run"],
                cwd=sabin_path,
                check=True
            )
        except subprocess.CalledProcessError as e:
            print("Erro ao executar robô Sabin:", e)
            # mensagens.warning(self.request, 'Robô Sabin falhou.')
        return super().form_valid(form)
```

Figura 11. Tarefa de coleta de documento.

```
1 *** Settings ***
2 Library RPA.HTTP
3 Library RPA.FileSystem
4 Library DateTime
5
6 *** Tasks ***
7 Run | Debug | Run in Interactive Console
8 Coletar Documento de Teste
9 # Garante que a pasta de destino exista
10 Create Directory ../uploads/despachos
11 # Gera timestamp para renomear o arquivo
12 ${timestamp} = Get Current Date result_format=yyyyMMdd
13 ${new_name} = Set Variable despacho_${timestamp}.pdf
14
15 # Faz o download e salva já com o nome personalizado
16 Download https://www.w3.org/WAI/ER/tests/xhtml/testfiles/resources/pdf/dummy
17 ... target_files ../uploads/despachos/${new_name}
18
19 Log Documento salvo como "${new_name}"
20
```

Figura 12. Configuração robô.

```
152 # colocar logo do LAP
153 story.append(Imagem_lap)
154
155 # Título do documento
156 story.append(Paragraph("${despacho.beneficio.tipo_beneficio}/NM", title_style))
157 story.append(Spacer(1, 20))
158
159 # Texto principal do atestado
160 evento_titulo = get_attr(despacho.beneficio.numero_beneficio, 'titulo', despacho.beneficio.tipo_beneficio) # Usar titulo se existir,
161 sendo none
162
163 texto_ateestado = f"""
164 O Beneficiário beneficiário nº {despacho.beneficio.numero_beneficio} está atualmente em fase de análise.
165 """
166
167 story.append(Paragraph(texto_ateestado, justify_style))
168 story.append(Spacer(1, 40))
169
170 # data_texto = "Santa Maria, { inscricao.evento.data_inicio.strftime('%d de %B de %Y')}."
171 # story.append(Paragraph(data_texto, right_style))
172 story.append(Spacer(1, 50))
173
174 # Texto final explicativo
175 texto_final = f"""
176 {despacho.beneficio.numero_beneficio} tem o data limite de 30 dias para se apresentar a uma aps do INSS ././
177 """
178
179 story.append(Paragraph(texto_final, justify_style))
180 story.append(Spacer(1, 40))
```

Figura 13. Criação pdf.

V. CONCLUSÕES

Neste trabalho, discutiu-se a automatização de processos, o uso de robôs para automação e as ferramentas necessárias para a implementação de robôs de processos, além dos desafios envolvidos em sua adoção. Foi abordado também o *framework* Django, com seu padrão arquitetural MVT, que sustenta a estrutura e a funcionalidade do sistema. Além disso, explorou-se o contexto do INSS e seus processos manuais, destacando os trabalhos realizados para transformar essas operações em soluções automatizadas. A análise incluiu as tecnologias empregadas e as melhorias geradas, oferecendo uma visão abrangente sobre as inovações e os impactos das soluções propostas.

Além disso, apresentou-se a proposta para a automação de processos no INSS, com base na metodologia Scrum e na técnica Kanban. A proposta foi suportada por um cenário de simulação BPMN via a ferramenta Bizagi, em que foi possível entender a dinâmica de funcionamento

dos despachos e notificações realizados até então, e projetar melhorias de fluxo em um sistema com módulo de automação.

Registra-se que o sistema possui funcionamento baseado em tecnologia Web, ou seja, opera diretamente em navegadores de Internet. No entanto, sua utilização é restrita a uma rede local com acesso exclusivo ao setor responsável pelos despachos e notificações referentes aos processos em manutenção ou cancelados junto ao INSS. Ainda assim, permanecem três desafios críticos a serem resolvidos na próxima etapa do projeto: i) implementação de um visualizador integrado ao aplicativo, possibilitando abrir, comentar e validar documentos diretamente no sistema, eliminando assim a dependência de ferramentas externas; ii) disponibilização do acesso aos sistemas internos do INSS para a coleta automatizada de documentos; iii) organização dos documentos coletados pelo robô de acordo com o respectivo benefício.

REFERÊNCIAS

- [1] Frank Lamb. *Automação industrial na prática-série Tekne*. AMGH Editora, 2015.
- [2] Breno Augusto Bezerra Tavares. *Sistema de Computação e Automação*. Lauro de Freitas. Disponível em <https://repositorio.pgsscogna.com.br>: Trabalho de Conclusão de Engenharia da Computação - UNIME, 2022.
- [3] Willian Yudji Suzuki Saito. “Desenvolvimento de um robô utilizando RPA (Robotic Process Automation) e software de BI (Business Intelligence) para melhorar um processo na área de engenharia de uma empresa”. Em: (2024).
- [4] *PyAutoGUI*. URL: <https://pypi.org/project/PyAutoGUI/> (acesso em 27/09/2024).
- [5] Al Sweigart. “Pyautogui documentation”. Em: *Read the Docs* 25 (2020).
- [6] *PIP: Package Installer for Python*. URL: <https://pypi.org/project/pip/> (acesso em 27/09/2024).
- [7] *Framework SikuliX*. URL: <https://sikulix-2014.readthedocs.io/en/latest/basicinfo.html> (acesso em 20/09/2024).
- [8] Robocorp. *Robocorp: Open Source RPA Tools*. Accessed: 2024-11-08. 2024. URL: <https://robocorp.com>.
- [9] David Heller. *Automating with Python and Robocorp: A Guide to Robotic Process Automation (RPA)*. Documentation for Robocorp, an open-source Python-based RPA platform. 2021. URL: <https://robocorp.com/docs>.
- [10] Eduardo Pavani Palharini e Alexandre Zamberlan. *Web Crawler Para Portais Da Área Do Direito*. Santa Maria, RS, Brasil. Disponível em <https://tfgonline.lapinf.ufn.edu.br>: Trabalho de Conclusão de Curso Sistemas De Informação - Universidade Franciscana (UFN), 2023.
- [11] *Selenium Web Driver*. URL: <https://www.selenium.dev/> (acesso em 02/03/2024).
- [12] Peter Hofmann, Caroline Samp e Nils Urbach. “Robotic process automation”. Em: *Electronic markets* 30.1 (2020), pp. 99–106.
- [13] *Framework Django*. URL: <https://www.djangoproject.com/>. (accessed: 01.09.2021).
- [14] *INSS*. URL: <https://www.gov.br/inss/pt-br/acesso-a-informacao/institucional/institucional> (acesso em 07/11/2024).
- [15] Mark Von Rosing, Henrik Von Scheel e August-Wilhelm Scheer. *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM, Volume 1*. Vol. 1. Morgan Kaufmann, 2014.
- [16] Fabricio Richetti e Ricardo Frohlich Da Silva. *Otimização De Processo De Uma Prestadora De Serviço Através Da Modelagem E Simulação De Processos De Negócio Com Uso Da Notação Bpmn*. Santa Maria, RS, Brasil. Disponível em <https://tfgonline.lapinf.ufn.edu.br>: Trabalho de Conclusão de Curso Sistemas De Informação - Universidade Franciscana (UFN), 2016.
- [17] Tomasz Wykowski e Justyna Wykowska. *Lessons learned: Using Scrum in non-technical teams*. Set. de 2019. URL: <https://www.agilealliance.org/resources/experience-reports/lessons-learned-using-scrum-in-non-technical-teams/>.