

Robotic Process Automation (RPA) aplicada à parametrização de religadores microprocessados

Tema: Sistemas de Controle, Automação e Proteção

Autores: Rodrigo Moreira Costa

Co-Autores: -

Empresa: CEMIG Distribuição S.A

Resumo

Este trabalho apresenta o desenvolvimento de uma ferramenta RPA (Robotic Process Automation) utilizando Python, destinada à parametrização automática de religadores microprocessados. A ferramenta foi projetada para otimizar e automatizar o processo de configuração desses dispositivos, que são essenciais para a proteção e controle em sistemas de distribuição de energia elétrica.

A solução proposta utiliza bibliotecas do Python para interagir com os softwares de parametrização dos religadores, permitindo a configuração precisa e eficiente dos parâmetros operacionais. Através de uma memória de cálculo feita em um arquivo Excel, contendo os dados de configuração do religador, a ferramenta consegue gerar automaticamente o devido arquivo de parametrização do dispositivo, reduzindo significativamente o tempo de configuração e minimizando erros humanos. Além disso, a flexibilidade da solução permite adaptações futuras para diferentes modelos de religadores e requisitos específicos de configuração.

A automação da atividade de parametrização de religadores microprocessados através de RPA representa um avanço significativo na gestão de sistemas de distribuição de energia, proporcionando grandes benefícios em termos de tempo, precisão e segurança. Aumenta-se o nível de produtividade através de uma operação ininterrupta e em tempo real, possibilitando que o corpo técnico se volte para as atividades de maior nível técnico e cognitivo.

1. Introdução

Este trabalho apresenta uma solução que utiliza a RPA para a configuração, também chamada de parametrização, de religadores microprocessados. Os religadores microprocessados, dispositivos importantes para a redução dos índices de duração das interrupções nas redes de distribuição, são cada vez mais utilizados pelas concessionárias de energia elétrica e na Companhia Energética de Minas Gerais (CEMIG) não é diferente.

São centenas de equipamentos que mensalmente são instalados, com funções diversas e muita das vezes distintas, contemplando dezenas de fabricantes e modelos diferentes. Existe, por outro lado, a demanda pela revisão, também chamada de reajuste, da parametrização desses religadores. Através da necessidade crescente deste serviço, viu-se a oportunidade de se criar uma ferramenta que automatizasse esse processo.

Desta forma, será destacado no decorrer deste trabalho o desenvolvimento de um robô, ou *bot*, que realiza automaticamente a parametrização da proteção dos religadores microprocessados da CEMIG. Esta solução já é aplicada a todos os novos religadores que entram na rede da CEMIG Distribuição (CEMIG D), contemplando mais de 12 fabricantes diferentes desses equipamentos que são utilizados pela empresa. Independente do horário ou do dia da semana, o *software* fica à disposição para ser executado através de um computador *desktop* que é exclusivo para a sua operação. Basta inserir um arquivo Excel (.xlsm) padronizado, que contém os dados obtidos no estudo de proteção, em determinados diretórios que se localizam na rede corporativa. A partir disso, o *bot* reconhece essa entrada e sai do estado de ociosidade e inicia a sequência de automação.

Os motivadores da idealização deste projeto, as alternativas observadas, o ciclo de operação, o nível de produtividade alcançado e as perspectivas futuras serão destacados no decorrer deste texto.

2. Desenvolvimento

2.1 O que é a RPA e como ela pode ser importante nos processos repetitivos?

RPA é uma sigla derivada de *Robotic Process Automation*, ou Automação Robótica de Processos, traduzindo para o português. É uma tecnologia que usa robôs, ou *bots*, que são programados para automatizar tarefas repetitivas que os seres humanos realizam rotineiramente. Esses *bots* podem “imitar” o comportamento humano e suas ações nas atividades computacionais do dia a dia. Esse tipo de tecnologia de *software* pode entender e interpretar o que se passa em uma tela, realizar interações através de movimentos no mouse, cliques e digitações completas, identificar e extrair dados, seja de uma base de dados interna ou da própria tela do computador, preencher formulários, mover arquivos e tudo mais que um ser humano comum possa fazer de forma ordenada e repetitiva. Utilizar a RPA no ambiente corporativo, onde várias atividades repetitivas são realizadas, pode gerar um aumento na produtividade, na performance de diversos processos, além de reduzir os erros humanos que são inerentes a essas atividades.

2.2 Parametrização de religadores microprocessados e atividade repetitiva

Nas concessionárias de energia elétrica existem áreas que são voltadas à atividade de planejamento da proteção da rede elétrica. Nesses locais, são realizados os estudos de proteção necessários à instalação dos religadores trifásicos e monofásicos. Na CEMIG, após o término desses estudos, é gerada a chamada ficha de parametrização, nos formatos Excel e PDF, que é o documento base para que o religador seja configurado. A configuração desses equipamentos é feita pela elaboração de arquivos padronizados para cada modelo, de acordo com as especificações que são inseridas na ficha pelos engenheiros de proteção. Para cada modelo diferente há um tipo específico de *software* e é nesse *software* que serão elaborados tais arquivos, que posteriormente serão “descarregados” no religador, fazendo-os funcionar de acordo com o estudo de proteção feito pelos engenheiros.

Até pouco tempo atrás, essa configuração de parâmetros de proteção em cada arquivo de cada *software* era feita manualmente. Em cada equipamento o técnico abria um determinado *software*, inserindo manualmente os dias, *pick-ups*, curvas de proteção de fase e terra, configurações de *hotline*, dentre outros diversos parâmetros.

Nesta atividade, o corpo técnico dispendia boa parte do seu tempo para ler e interpretar o que estava contido na ficha para depois inserir nos campos respectivos de cada *software* os dados de proteção. Uma atividade de “copiar e colar” que é suscetível a erros de digitação e que precisava ser feita com extremo cuidado pelos responsáveis, visto que um número a mais ou a menos poderia comprometer seriamente a parametrização e causar problemas na proteção da rede. Ademais, o próprio engenheiro de proteção

que elabora a ficha do equipamento também tinha que se atentar aos limites de cada dispositivo e inserir corretamente os dados planejados, pois uma curva ou *pick-up* diferente do planejado causaria sério risco para o sistema de proteção.

Desta forma, enxergando uma possibilidade de melhoria no processo, a ideia de se tentar fazer arquivos de parametrização automaticamente surgiu, porém as ferramentas que foram imaginadas no primeiro instante não surtiram o efeito desejado, visto que um outro objetivo requerido para o *software* era o de implementar gatilhos lógicos que pudessem também ajudar na identificação de erros nas fichas e aperfeiçoar o próprio processo interno, eliminando erros de escritas ou valores que extrapolavam determinados limites de cada equipamento.

2.3 Rapidez, eficiência e lógica: o uso do Python na construção de um software de RPA

A solução final para um *software* que fizesse automaticamente os arquivos de ajustes dos religadores surgiu quando se identificou o potencial do Python, uma linguagem de programação que apresenta diversos tipos de bibliotecas voltadas a essa atividade. Percebeu-se que a criação de um *software* nessa linguagem possibilitaria atender os requisitos desejados, que eram a criação de uma automação veloz, confiável, com gatilhos lógicos que evitariam possíveis erros e fosse de fácil implementação.

A partir disso, foi feito um trabalho voltado para que fosse possível:

- A leitura das fichas (também chamadas de memória de cálculo) no seu formato em Excel;
- A abertura do *software* específico do modelo do religador, escolhendo o arquivo padronizado correto;
- A inserção dos dados de proteção, através de ferramentas de leitura de tela, comando de mouse e teclado;
- A conclusão da elaboração do arquivo e a exportação do mesmo para uma pasta compartilhada, onde os técnicos responsáveis poderiam utilizar o arquivo de parametrização já pronto e finalizado.
- A repetição do ciclo.

2.4 Bot Auto Arquivo: o software que parametriza religadores microprocessados de forma automática

A partir da solução encontrada e com o desenvolvimento contínuo, o *software* “Auto Arquivo” foi concebido. Atualmente são 12 modelos diferentes de religadores já contemplados pelo *software*, 11 trifásicos e um monofásico, englobando quase que a totalidade dos religadores da rede CEMIG, salvo raras exceções. O *bot* opera em um processo cíclico, observado na Figura 1, gerando vários arquivos de parametrização dos 12 diferentes modelos de acordo com a demanda. Quando o engenheiro libera uma ficha para o robô, o *script* reconhece essa atualização e já inicia o processo de elaboração do arquivo. É feito o reconhecimento do modelo e o armazenamento dos dados em variáveis do Python. Logo após, é feito um “pente fino” nessa ficha, ou seja, operações lógicas diversas são feitas para validar que a ficha está apta a gerar um arquivo. Por exemplo, se o valor do *pick-up* é válido, se a curva escolhida é a padrão utilizada pela CEMIG D, se os dados de dial e tempo de atuação foram devidamente inseridos, dentre outros. Caso não haja conformidade, é gerado um *log* de erros e o arquivo não é elaborado. Os técnicos que acompanham a operação ficam cientes através de um aplicativo auxiliar, que informa os *logs* de erros e demais informações da operação do *bot*.

A partir do aplicativo auxiliar, observado na Figura 2, o técnico pode acompanhar em tempo real quais e quantos arquivos foram feitos e quais precisam de alguma correção. Os arquivos não conformes são revistos e posteriormente voltam ao *bot*. Caso não haja nenhum erro, o algoritmo continuará a sua operação selecionando o arquivo padrão correto para aquele modelo/religador e começará a inserção de dados, utilizando bibliotecas para o comando do mouse e teclado e para a identificação de imagens. Após a inserção correta dos dados, o arquivo é salvo e posteriormente exportado para uma pasta compartilhada

na rede corporativa, onde será futuramente repassado às oficinas e “descarregado” nos religadores da rede CEMIG.



Figura 1 - Ciclo de atuação do *bot* Auto Arquivo

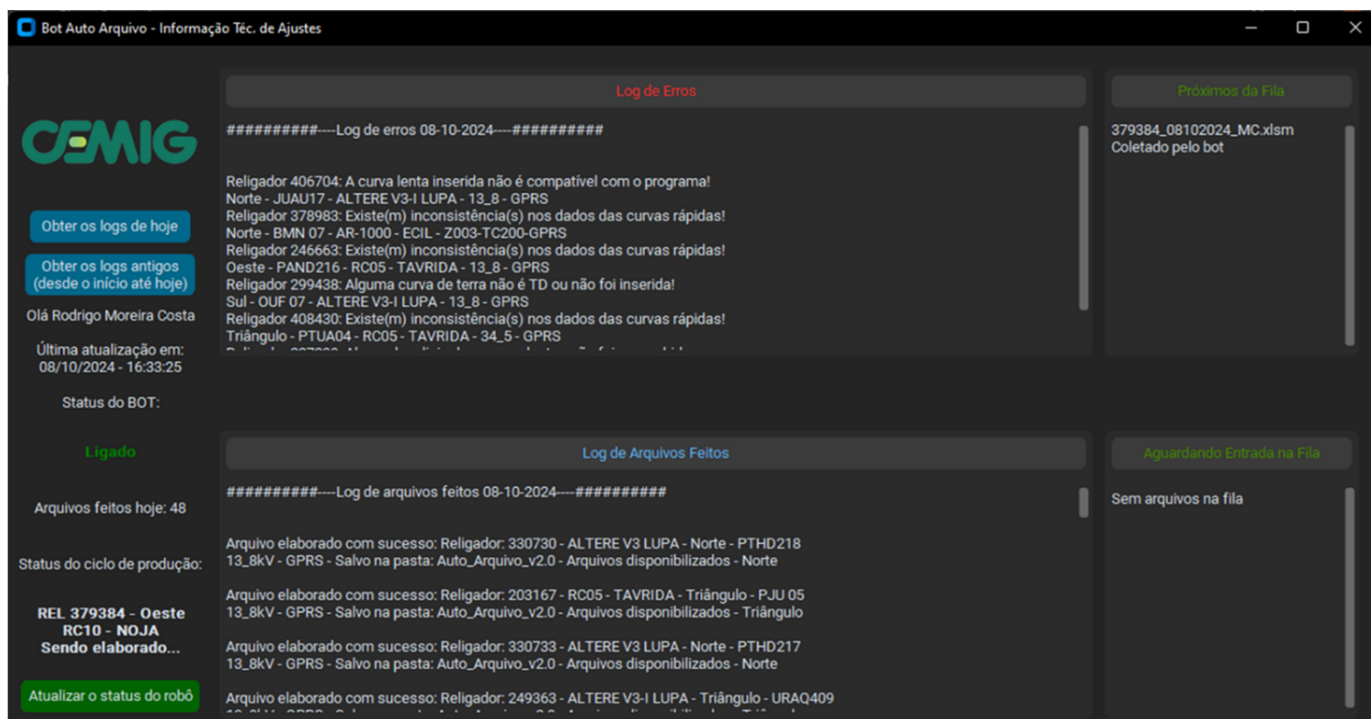


Figura 2 - Aplicativo auxiliar que informa o status da operação do *bot* aos técnicos

2.5 Bot Auto Arquivo: A lógica por trás da ferramenta

Para que todo o processo anteriormente destacado fosse possível de ser colocado em prática, linhas de programação foram necessárias. Quando um arquivo Excel com os dados de algum religador é colocado na rede, o programa desperta para iniciar a sua operação. Para isso, um *loop* infinito é realizado, realizando a leitura constante dos diretórios onde esses arquivos são deixados pelos técnicos responsáveis. Esses arquivos são movidos, através da função *shutil.move*, para um diretório interno do computador que roda a aplicação. A partir daí, utilizando principalmente a biblioteca *pandas*, todo o conteúdo do Excel é lido. Esse Excel, que é a ficha de ajustes que os Engenheiros produzem, contém todos os dados necessários para a parametrização dos religadores, conforme observado na Figura 3, a seguir:

CEMIG		ESTUDO DE COORDENAÇÃO DA PROTEÇÃO PLANEJAMENTO DA OPERAÇÃO DE MÉDIA TENSÃO AJUSTES DE RELIGADOR											
Salvar como: Completar dados		Salvar PDF		Chamado Ajuste		Chamado de fúteis		Chamado de desequilíbrio		Elab. por: Engenheiro		Edição: * DV	
Malha: Centro		Condição: NF		Eqpto: 999999		Alimentador (es): TESTE 123		NS/Cham: TESTE		SIOM: RELI		Espécie Nº 0 0	
Motivador: Instalação		Data:		Coord.:		End.:		TESTE					
Fabricante: TESTE		Modelo: TESTE		Comando: TESTE		Telecontrolado: SIM		By-Pass: FACA					
Corrente Fase A: 139		Icc Trifásico no ponto instalad: 2345		Bitola de cabo crítica no trecho: 10 CA		Nominal Admiss. 201A 274 A		Vinculado à acessante: NÃO					
Corrente Fase B: 157		Icc Fase-Fase no ponto mais dist: 1696		Pickup da proteção de terra do equip. a montante: 70									
Corrente Fase C: 145		Icc Fase-Terra min ponto mais dist: 190		Tempo definido da proteção de terra a montante: 10									
AJUSTE DE FASE				AJUSTE DE TERRA									
Pick-Up		Dial		Pick-Up		T. Máx							
Mínimo Máximo		Mínimo Máximo		Mínimo Máximo		9,8							
Principal: 200 200				Principal: 20 50		Tempos sugeridos T.Max.F[s]							
Alternativo: 195 274				Alternativo: 50 50		Lenta Rápida							
Sugerido: 200				Sugerido: 65		9 1							
Grupos de Ajuste NF										Fluxo Normal = Fluxo Inverso			
										FASE			
										TERRA			
										Temp o Morto			
Grupo 1 -										10			
200 3L										20			
IEC MUITO INVERSO										-			
Dial T. Adic. Dial T. Adic.										-			
0,32										9,5			
Grupo 2 -										10			
225 3L										20			
IEC MUITO INVERSO										-			
Dial T. Adic. Dial T. Adic.										-			
0,28										9,5			
Grupo 3 -										10			
200 1R + 3L										20			
IEC MUITO INVERSO IEC INVERSO										20			
Dial T. Adic. Dial T. Adic.										-			
0,32 0,01										9,5 1			
Grupo 4 -										-			
630 1L										-			
T. DEFINIDO Lenta 20										-			
630 1L										-			
T. DEFINIDO Lenta 20										-			

Figura 3 - Modelo de uma ficha de ajustes de proteção que será lida pelo *bot*

Após a leitura das células que contêm os dados que serão utilizados na parametrização, a automação salva essas informações em variáveis do Python e abre o *software* referente ao fabricante/modelo do religador. É nesse *software* que são inseridos os dados da parametrização e que é produzido o arquivo de configuração do religador. Como a CEMIG trabalha com mais de uma dezena de modelos distintos, para cada um deles a atuação do robô é diferente. Existem funções e maneiras de se inserir os dados distintas, e por isso foi criada uma função específica no Python para cada modelo.

Tais funções são baseadas na biblioteca *pyautogui*, imensamente importante e voltada para a RPA. É através dela que é possível “imitar” o comportamento humano, sendo possível controlar o mouse e o teclado. Além disso, a leitura de imagens da tela, incluída na biblioteca, auxilia nas movimentações e cliques do mouse. A biblioteca *time* irá acrescentar tempos de parada do programa, nas situações em que é necessário um momento de espera para carregamentos de páginas ou para um clique ou digitação serem efetivamente reconhecidos.

Desta forma, os dados armazenados nas variáveis são inseridos nos campos devidos do *software*, utilizando principalmente as funções *pyautogui.write* e *keyboard.write*. Ainda é importante salientar que esse passo do ciclo de operação só é realizado após os dados obtidos no Excel passarem por uma extensa lista de comparações lógicas que irão atestar se os dados solicitados pelo Engenheiro estão aptos a serem inseridos na parametrização. Por exemplo, se o dial de alguma curva está vazio ou fora dos limites pré-esta-

belecidos, a automação irá rejeitar o arquivo e informar, via *log* de erros no aplicativo de acompanhamento, o erro encontrado.

Essa ligação entre o *bot* e o aplicativo é feita de forma muito simples: via arquivos .txt que estão na rede corporativa. A automação insere a linha de erro, através das funções *open(nome_do_diretorio)* e *nome_do_diretorio.write* que são nativas do Python, e o técnico que acompanha o ciclo de atividades do robô irá receber a informação na sua interface. Isso vale também para o contador de arquivos feitos no dia e para o *log* de arquivos produzidos, que são outras funcionalidades do aplicativo de acompanhamento.

Após o término da inserção dos parâmetros, um arquivo é gerado, salvo e armazenado em diretórios específicos na rede, em pastas que representam cada malha que a rede CEMIG D é dividida (Centro, Mantiqueira, Sul, Oeste, Leste, Norte, Triângulo). A partir disso, os técnicos retiram esse arquivo dos diretórios e os mandam para as oficinas, que irão posteriormente inseri-los nos religadores, configurando-os.

Na Tabela 1, a seguir, algumas das bibliotecas do Python que viabilizaram a construção do *bot* e suas funcionalidades são destacadas:

Tabela 1 - Bibliotecas do Python que viabilizaram a construção desse *bot* e suas funcionalidades

Bibliotecas	Funcionalidade
pandas	pandas.read_excel() lê o conteúdo de células do Excel. Primordial na coleta de dados das fichas que são alocados nas variáveis do python.
pyautogui	pyautogui.click(coord_x,coord_y) , pyautogui.locateCenterOnScreen(imagem) , pyautogui.press(botão) , pyautogui.write(texto) , pyautogui.rightClick(coord_x,coord_y) , pyautogui.hotkey(botão,botão) são algumas das possibilidades dessa poderosa biblioteca que pode movimentar e realizar cliques no mouse, digitar textos e pressionar botões, além de localizar imagens ou textos na tela do computador. É a base de toda a operação do <i>bot</i> na elaboração dos arquivos.
datetime	O datetime.now() informa a data e hora naquele determinado instante. Usado para que seja possível calcular o tempo total de atuação do <i>bot</i> .
pyperclip	pyperclip.copy() e pyperclip.paste() são os famosos “ctrl + c” e “ctrl + v
win32	As funções VK_CAPITAL e GetKeyState são capazes de identificar se o “Caps Lock” está ativo ou não, sendo possível depois ativá-lo/desativá-lo.
pygetwindow	pygetwindow.getWindowsWithTitle(x) é capaz de selecionar uma determinada janela x do Windows, que posteriormente pode ser maximizada ou minimizada.
keyboard	Dá acesso total às teclas do teclado, sendo uma função auxiliar para os casos em que o pyautogui não apresenta uma solução eficaz. Ex.: keyboard.write()
shutil	shutil.move() poderá mover os arquivos entre pastas no computador, ideal para a movimentação das fichas e arquivos de ajustes.
time	Utilizando time.sleep(x) é possível pausar a execução do script por x segundos, o que pode ser necessário em janelas de carregamento ou para diminuir a velocidade do programa, evitando cliques ou digitação indevidos.
os	O modulo “os” coleta informações úteis do sistema. Por exemplo, para saber quem é o usuário que logou na máquina basta usar os.getlogin() .
sys	sys.exit() realiza o desligamento forçado do <i>bot</i> e do ciclo de operações. Utilizado em casos de instabilidade ou falta de confiança do <i>bot</i> em realizar alguma tarefa, por exemplo a não possibilidade de se identificar alguma imagem necessária à operação correta do <i>script</i> .

2.6 Bot Auto Arquivo: Os níveis de produtividade alcançados

A construção da automação evoluiu desde o seu início até o momento presente, onde se consolidou como ferramenta principal na parametrização dos religadores microprocessados. Inicialmente, apenas um fabricante/modelo era contemplado e gradualmente as demais funções do Python com os outros modelos foram desenvolvidas. Desta forma, os números de produtividade da ferramenta também cresceram gradualmente até alcançar o patamar atual. A seguir, são apresentados alguns quantitativos sobre o desempenho da aplicação:

- Estima-se que mais de 5500 arquivos de parametrização foram produzidos pelo robô até meados de novembro de 2024;

- Em 24/05/2024 um contador foi implementado no *bot*, para quantificar a produção da aplicação. Desta data até 22/11/2024 foram produzidos 4215 ajustes de proteção;
- Realizando a média aritmética, foram 4215 ajustes produzidos em 127 dias, totalizando uma média de 33 arquivos/dia;
- Em média, cada arquivo demora 10 minutos para ser produzido, esse valor varia para cada tipo de *software*. Alguns podem ser feitos mais lentamente que outros, pois demandam mais cliques e digitação ou tem mais telas para serem abertas e configuradas;
- Tendo em vista esse valor, economiza-se por dia, em média, 330 minutos de trabalho humano ou 5 horas e 30 minutos, cerca de 70% da jornada diária de trabalho de um funcionário da CEMIG;
- Nas situações de pico de produtividade, por duas vezes foram elaborados 103 arquivos em um só dia. Considerando o mesmo valor de 10 minutos para a elaboração de cada arquivo, são 1030 minutos economizados em atividade humana repetitiva, ou 17 horas e 10 minutos. São mais de dois dias completos de jornada de trabalho de um funcionário da CEMIG.

A seguir, podemos observar na Figura 4 e na Figura 5 os gráficos de produção diária e média de produção diária alcançadas pelo *bot*. Como a atividade tem uma variação considerável de demanda, os números de produção também acompanham essa flutuação. Os picos de produtividade, como já citado, foram de 103 arquivos de parametrização produzidos em um só dia. A média máxima obtida até agora foi de 34 arquivos/dia e atualmente está na faixa dos 33 arquivos/dia.

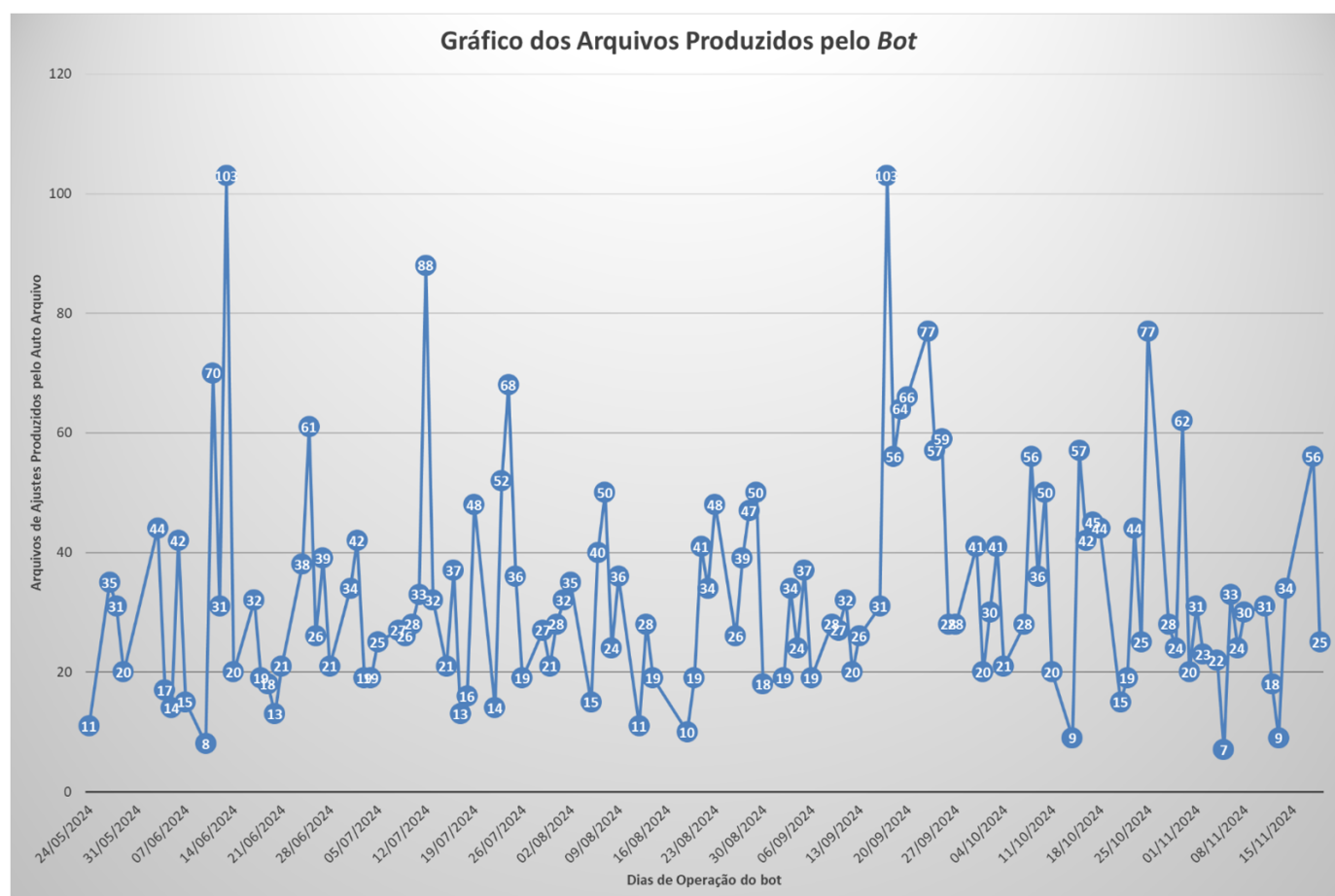


Figura 4 - Produtividade diária do *bot* Auto Arquivo

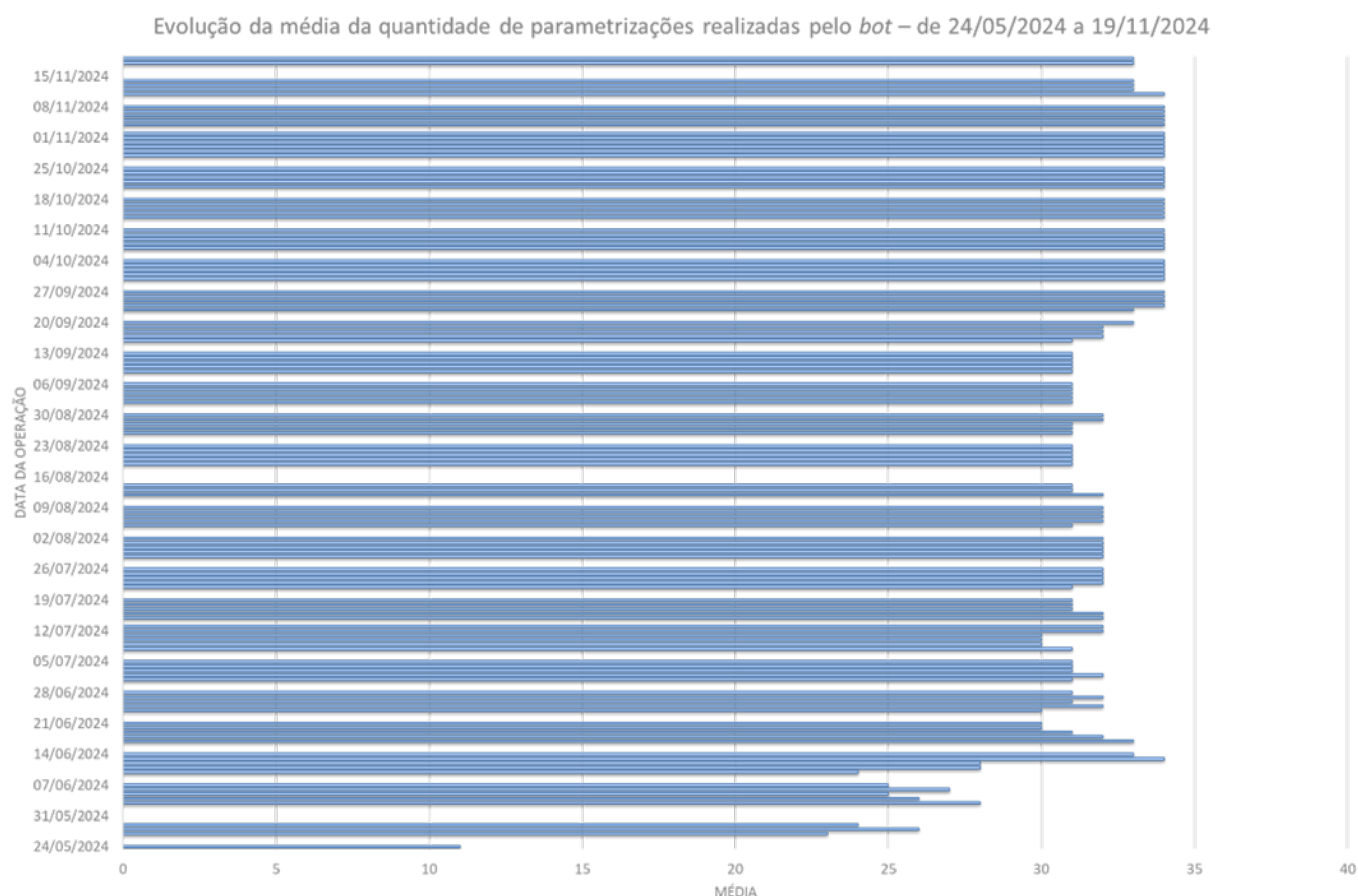


Figura 5 - Evolução da média diária da produtividade do bot Auto Arquivo

2.7 O próximo passo: IA e aumento da automatização do processo

Para obter camadas a mais de segurança na coleta dos dados que são obtidos pela automação através da ficha de ajustes, o uso da inteligência artificial será de grande importância. Nas memórias de cálculo é comum que o engenheiro escreva, à mão livre, informações pertinentes àquele ajuste que ele elaborou. Uma dessas informações é o tempo máximo de fase, que é inserido em um campo específico da ficha, mas também é detalhado nas observações do engenheiro.

Para evitar incompatibilidades, será utilizada uma API (*Application Programming Interface*) do ChatGPT para o Python que será útil nesse tipo de situação. Essa API informará para uma variável Python se o engenheiro inseriu ou não algum texto relativo ao tempo máximo de fase à mão livre e qual o seu valor. Essa informação deve ser comparada com aquela que o próprio engenheiro já informou no campo específico e a partir daí é possível evitar divergências de valores ou até mesmo informar a ausência desse valor no campo específico. É uma simples pergunta para a IA e uma forma adicional de identificar um erro que futuramente poderia causar uma falha de proteção nos religadores da rede. Além desse exemplo, a própria IA poderá, futuramente, fazer uma avaliação de todos os valores pertinentes dos parâmetros de proteção, identificando outras inconsistências ou valores não usuais, alertando o engenheiro de forma a revisar o seu estudo de proteção.

Além disso, o processo de disponibilização dos ajustes de proteção dos religadores diz respeito não só à elaboração do ajuste em si, mas também a sua inserção em um sistema corporativo e demais modificações no sistema SAP. Uma vez que o engenheiro disponibiliza a ficha de proteção e o técnico a libera, o sistema

identifica essa situação, elabora o ajuste de proteção e o robô dá o seu trabalho como finalizado, aguardando novos arquivos para serem produzidos. Um próximo passo dessa automatização seria a inclusão das outras duas etapas nesse ciclo de trabalho do *bot*, que faria a inserção dos arquivos no sistema corporativo e realizaria as modificações no SAP. Os estudos sobre a viabilidade dessa integração já começaram e as possibilidades iniciais foram animadoras. É possível que nos próximos meses apenas uma pessoa se dedique a operação da ferramenta RPA, podendo se dedicar também a outros trabalhos paralelos, graças a essa nova solução. Desta forma, uma atividade que era feita por até 4 pessoas, no atendimento aos períodos de maior demanda, se reduziria a apenas 1 pessoa, evitando o seu serviço repetitivo, abrindo possibilidades para que um trabalho mais técnico e menos mecânico seja feito, reduzindo drasticamente os erros e aumentando a produtividade do processo como um todo.

3. Conclusão

Como pôde ser observado no decorrer deste trabalho, a utilização de uma solução baseada em RPA para a automação de atividades humanas repetitivas foi de grande valia para o processo de parametrização de religadores microprocessados da CEMIG D.

O desenvolvimento do *bot* Auto Arquivo trouxe maior confiabilidade e velocidade na elaboração das parametrizações, além de evitar que um corpo técnico inteiro fosse responsável por uma atividade repetitiva e monótona.

A aplicação deste tipo de solução pode ser disseminada para outros processos dentro da empresa e este projeto já é um precursor de novas ideias. Uma delas é a criação de um outro robô, que tem como função definir curvas de proteção de religadores automaticamente. Desta forma, o processo de coordenação da proteção e seletividade poderia ser agilizado e ao mesmo tempo obteria uma confiabilidade baseada em cálculos precisos, que só a capacidade computacional pode entregar. O próprio *bot* Auto Arquivo está em processo de melhorias e integração de novas funções, como já citado.

Desta forma, é nítido observar a importância dos projetos de desenvolvimento de *software* nos processos da área da distribuição de energia elétrica, em especial nos sistemas de proteção. O patamar de execução de diversas atividades se eleva com o uso da capacidade computacional, dentre elas a própria inteligência artificial, sendo essencial para as concessionárias de energia elétrica que queiram se destacar e apresentar os melhores resultados possíveis para os seus clientes e administradores.

4. Referências bibliográficas

PyAutoGUI's documentation. Acesso em 20/11/2024, disponível em <https://pyautogui.readthedocs.io/en/latest/index.html>.

Pandas documentation. Acesso em 25/11/2024, disponível em <https://pandas.pydata.org/docs/>.

Shutil - High-level file operations. Acesso em 27/11/2024, disponível em <https://docs.python.org/3/library/shutil.html>.