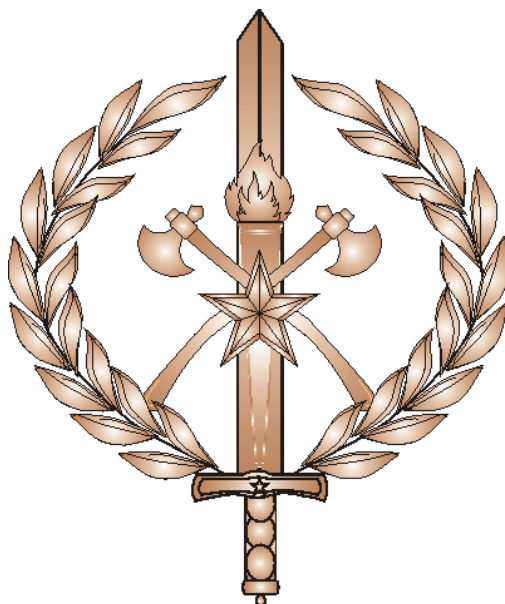


**CORPO DE BOMBEIROS MILITAR DO DISTRITO FEDERAL
DEPARTAMENTO DE ENSINO, PESQUISA, CIÊNCIA E TECNOLOGIA
DIRETORIA DE ENSINO
CENTRO DE ESTUDOS DE POLÍTICA, ESTRATÉGIA E DOCTRINA
CURSO DE ALTOS ESTUDOS PARA OFICIAIS**

Maj. QOBM/Compl. JOÃO DE **ALMEIDA** NETO



**AUTOMATIZAÇÃO DA GERAÇÃO DE RESSUPRIMENTO DE
MATERIAIS POR MEIO DE *WEBSCRAPING* DO PORTAL PAINEL DE
PREÇOS E INTEGRAÇÃO COM SISTEMA LABORTI**

BRASÍLIA
2021

Maj. QOBM/Compl. JOÃO DE **ALMEIDA** NETO

**AUTOMATIZAÇÃO DA GERAÇÃO DE RESSUPRIMENTO DE
MATERIAIS POR MEIO DE *WEBSCRAPING* DO PORTAL PAINEL DE
PREÇOS E INTEGRAÇÃO COM SISTEMA LABORTI**

Monografia apresentada ao Centro de Estudos de Política, Estratégia e Doutrina como requisito para conclusão do Curso de Altos Estudos para Oficiais do Corpo de Bombeiros Militar do Distrito Federal.

Orientador: Ten. Cel. RRM/PTTC Luís Cláudio de **AQUINO** Alencar

BRASÍLIA
2021

Major QOBM/Compl. JOÃO DE **ALMEIDA** NETO

**AUTOMATIZAÇÃO DA GERAÇÃO DE RESSUPRIMENTO DE
MATERIAIS POR MEIO DE *WEBSCRAPING* DO PORTAL PAINEL DE
PREÇOS E INTEGRAÇÃO COM SISTEMA LABORTI**

Monografia apresentada ao Centro de Estudos de Política, Estratégia e Doutrina como requisito para conclusão do Curso de Altos Estudos para Oficiais do Corpo de Bombeiros Militar do Distrito Federal.

Aprovado em: ____/____/____.

BANCA EXAMINADORA

Ten-Cel. QOBM/Comb. JEANN WILSON AGUIAR CAVALCANTE
mat. 1400122
Presidente

Ten-Cel. QOBM/Comb. LEONARDO MONTEIRO LOPES
mat. 1400146
Membro

Profa. ZILTA DIAS PENNA MARINHO
Membro

Ten. Cel. RRM/PTTC Luís Cláudio de **AQUINO** Alencar
Orientador

RESUMO

Em um estudo é sobre a implementação de um algoritmo de *webscraping* que realize a tarefa de obtenção de preços públicos de licitação listados pelo sistema de compras governamentais do Governo Federal, no Laboratório de Análises Clínicas da Policlínica Médica do CBMDF. Foi utilizada a linguagem de programação *python* e o módulo de *webscraping* *SELENIUM* para a construção do algoritmos. Apesar de o algoritmo ter demonstrado a capacidade de reduzir cerca de 30% o tempo gasto na pesquisa no referido portal, o ganho de tempo foi ainda mais significativo, já que ele se atualiza automática e periodicamente, reduzindo a carga de trabalho dos recursos humanos que antes desempenhavam essa tarefa.

Palavras-chave: Programação. *Python*. Automatização de Tarefas. Licitação Pública. Balizamento de Preços.

LISTA DE ILUSTRAÇÕES

Figura 1 - Metodologia de Ponto de Ressuprimento.....	21
Figura 2 - Processo de Processamento de Pedidos.....	22
Figura 3 - Etapas do processo de <i>Webscraping</i>	28
Figura 4 - <i>Script</i> principal do algoritmo.....	29
Figura 5 - <i>Script</i> contendo os localizadores dos elementos.....	30
Figura 6 - <i>Script</i> contendo as ações realizadas em cada localizador.....	31
Figura 7 - Processo atual de pesquisa de preços no Painel de Preços	36
Figura 8 - Tabela final do algoritmo com aferição do tempo de execução....	37
Figura 9 - Desempenho aferido na pesquisa de preço – Experimentos.....	39

SUMÁRIO

1. INTRODUÇÃO.....	9
1.1 Definição do problema.....	11
1.2 Justificativa.....	14
1.3 Objetivos.....	15
1.3.1 Objetivo geral.....	15
1.3.2 Objetivos específicos.....	15
1.4 Hipóteses/Questões.....	16
1.5 Definição de termos.....	16
2. REVISÃO DE LITERATURA.....	18
2.1 Automatização de Tarefas.....	18
2.2 Gestão de Estoques.....	19
2.3 A linguagem HTML.....	23
2.4 <i>Webscraping</i>	25
3. METODOLOGIA.....	26
3.1 Apresentação.....	26
3.2 Procedimentos.....	27
4. RESULTADOS E DISCUSSÃO.....	33
4.1 Desenvolvimento e Execução do Algoritmo.....	33
4.2 Processo atual: estimativa através do mapeamento do processo.....	35
4.3 Experimento 2: Execução real do processo atual.....	38
4.4 Discussão dos Resultados.....	39
5. CONSIDERAÇÕES FINAIS.....	41
6. RECOMENDAÇÕES.....	43
REFERÊNCIAS.....	44
ANEXOS.....	47
APENDICES.....	49

1. INTRODUÇÃO

A automatização de tarefas é uma tendência decorrente do avanço da tecnologia de informação no campo da gestão de processos corporativos. Se de um lado a automatização pode suscitar reações negativas, como o medo da substituição da mão de obra humana por robôs, por outro, é uma oportunidade de aumentar a produtividade com pouco investimento.

A automatização é um sistema constituído por dispositivos mecânicos ou eletrônicos, utilizado em fábricas e estabelecimentos comerciais, em telecomunicações, em instituições hospitalares e bancárias etc., destinado à operacionalização e controle dos processos de produção, que dispensa a intervenção direta do homem (MICHAELIS, 2021).

A dispensa da intervenção direta, aos olhos do gestor, pode significar redução de custos com pessoal, mas ganho de eficiência. Em tempos nos quais o termo *accountability*¹ vem se tornando frequente na administração pública, o questionamento quanto à eficiência da operação das organizações é o cotidiano dos noticiários.

Nesse contexto, os órgãos públicos necessitam de dados que permitam gerar informação gerencial, e dessa forma conseguir otimizar seus processos, monitorar seu desempenho e até mesmo fazer comparações com outros órgãos.

No entanto, o acesso a esses dados oriundos de bancos de dados públicos não é uma tarefa trivial. Pinto (2006), ao analisar uma estratégia de integração entre os bancos de dados do Sistema Único de Saúde (SUS), relata uma série de dificuldades para tal tarefa. Uma delas é da natureza dos bancos de dados, que possuem dados “brutos”, ou seja, precisam ser trabalhados, lapidados e discutidos para gerar informações relevantes, necessitando grande quantidade de trabalho para limpar as variáveis e torná-las palatáveis ao entendimento do público geral.

1 Na palavras de Rocha(2011), *accountability* diz respeito aos “...processos de avaliação e responsabilização permanente dos agentes públicos que permitam ao cidadão controlar o exercício do poder concedido aos seus representantes”.

Dados brutos encontrados nesses diferentes bancos de dados podem apresentar problemas de variabilidade de formatação, como por exemplo, um mesmo dado possuir rótulos diferentes:

As opções de resposta da variável “sexo” são codificadas como ‘1. Masculino’ e ‘3. Feminino’, no Sistema de Informação Hospitalar (SIH-SUS), como ‘M’ e ‘F’, no Sistema de Informação sobre Nascidos Vivos (SINASC), e como ‘1. Masculino’ e ‘2. Feminino’, no (SIM) Sistema de Informação sobre Mortalidade (PINTO, 2006, p. 66).

Dentre esses dados necessários, destacam-se dados sobre o preço praticado em licitações públicas, uma vez que, estes são úteis na construção de balizamento de preços de futuras aquisições.

A forma como os preços são disponibilizados pelos órgãos públicos não é diferente; são, em sua maioria, iniciativas localizadas, não padronizadas e de difícil integração. Causa dificuldade, lentidão e aumento da possibilidade de erro na realização de consultas pelos órgãos da administração pública.

A Policlínica Médica do Corpo de Bombeiros Militar do Distrito Federal possui um serviço de Análises Clínicas estabelecido há mais de 20 anos, subordinado à Seção de Farmácia e Laboratório da Policlínica Médica, hierarquicamente subordinada à Diretoria de Saúde e ao Departamento de Recursos Humanos do CBMDF: a SULAC/POMED. Ao longo de sua existência este serviço tem se destacado por um processo constante de melhoria de qualidade, e a presente proposta de pesquisa pretende contribuir com este processo otimizando a gestão de suprimentos do setor.

A SULAC/POMED participa do processo de aquisição de insumos por meio de licitação pública, realizando as etapas de Oficialização da Demanda, Estudo Técnico Preliminar, Mapa de Riscos e Pedido de Execução. Nestes processos, uma etapa crítica é a obtenção de preços públicos para balizamento do preço estimado: é um processo demorado, pois envolve entrada manual de dados, como especificação do material e quantidades, item a item, em aplicações de internet que consomem muito tempo. Uma abordagem mais profunda sobre esse tema será apresentada na seção Definição do Problema.

Na justificativa, apresenta-se os motivos pelos quais a automatização da tarefa de pesquisa de preço se relaciona com os objetivos estratégicos do CBMDF, bem como os possíveis benefícios da sua implementação.

Existe uma série de ferramentas tecnológicas disponíveis em código aberto na internet que permitem automatizar muitas tarefas atualmente desempenhadas por seres humanos. Na seção denominada objetivos serão apresentadas diretrizes para a verificação da aplicabilidade e vantajosidade da utilização de tecnologia de automatização de tarefas em um processo administrativo importante da Corporação, que é a pesquisa de preços públicos para balizamento de preços, com foco em uma técnica de obtenção de dados chamada *Webscraping*.

A literatura consultada para subsidiar esse trabalho compreende três grandes áreas: Automatização de Tarefas, Gestão de Estoque, a linguagem HTML e *Webscraping*. A seção que descreve a metodologia apresenta a classificação da pesquisa, bem como os meios e métodos utilizados durante seu desenvolvimento.

Diante do exposto, a finalidade da presente pesquisa foi estudar o emprego de uma solução tecnológica para melhorar o tempo de execução do processo de gestão de suprimentos do laboratório do CBMDF. Os resultados apresentados permitirão a análise da viabilidade desta solução pelos atores envolvidos corporativamente neste processo.

1.1 Definição do problema

O problema de pesquisa deste trabalho trata de determinar se a automatização de tarefas é capaz de reduzir o tempo gasto na pesquisa de preços de processos de aquisição do laboratório da Policlínica Médica.

A pesquisa de preços para licitação, cuja norma de execução é fundamentada na Lei Federal 8.666/1993, é uma atividade que depende do trabalho humano de maneira intensiva. Para obter os preços de licitações ocorridas no passado, o militar responsável, no atual desenho do processo,

precisa manter-se por longo tempo operando uma aplicação gráfica disponibilizada pelos órgãos gerenciadores da licitação.

No âmbito do CBMDF, esta pesquisa é feita através de dois portais públicos: o Mapa de Preços de Notas Fiscais Eletrônicas da Secretaria de Fazenda do Distrito Federal (MAPADEPRECOS, 2021), e o Portal de Preços do Ministério da Economia (PAINEL DE PREÇOS, 2021).

O primeiro deles é um portal bastante rudimentar, principalmente porque não utiliza uma padronização de materiais, o que dificulta a comparação entre os itens da licitação; no entanto, segundo o Decreto nº 39.453, de 14 de novembro de 2018, do governo do Distrito Federal, sua utilização é obrigatória. No que diz respeito à situação específica da SULAC/POMED, sua utilização é ainda mais complicada, pois os preços constantes nas Notas Fiscais Eletrônicas não especificam o regime de contratação, que pode incluir somente insumos como ser combinado a prestação de serviços especializados.

Os laboratórios clínicos da iniciativa privada possuem um perfil diferente de contratação de insumos, fato observado pela equipe de farmacêuticos bioquímicos da SULAC/POMED, com experiência de mais de 20 anos no setor. Ao contrário dos órgãos públicos, esses laboratórios optam por não celebrar contratos longos, possuem maior facilidade de dispor de patrimônio, como equipamentos, e maior flexibilidade para contratar diferentes fornecedores e prestadores de serviço.

Há ainda o problema da não especificação do regime de contratação (para o setor de laboratório clínico, existem os regimes de compra do equipamento e kits diagnóstico, compra dos kits diagnóstico com aluguel dos aparelhos e prestação de serviços, e fornecimento de kits com cessão de equipamentos em regime de comodato), causando uma variação de preços muito grande, o que dificulta o balizamento.

Dessa forma, laboratórios privados trabalham com preços muito diferentes da realidade dos órgãos públicos, o que torna a pesquisa para balizamento de preços para licitação no portal da SEFAZ/DF ainda menos confiável.

Em virtude da precariedade das informações deste portal, suas pesquisas de preço em geral não retornam nenhum item aderente às especificações da SULAC/POMED, ou fornecem preços muito díspares da realidade do serviço público. No entanto, como o decreto citado demanda apenas 1 (um) preço obtido por esse portal, sua obtenção não é uma etapa limitante do processo. A pesquisa dos demais preços (o decreto demanda três preços para compor o preço médio final), pelo Portal de Preços ainda consome a maior parte do tempo despendido na atividade de balizamento de preços.

Ainda que a Secretaria de Fazenda ofertasse a pesquisa de preços por meio da aplicação de uma interface gráfica, e sim uma interface de aplicação, essa pesquisa também poderia ser automatizada.

Para o presente estudo, optou-se por tentar automatizar a pesquisa de preços no segundo portal supracitado, o Portal de Preços do Ministério do Planejamento, Orçamento e Gestão, denominado “painel de preços”, já que este portal é utilizado pelo CBMDF para realizar os pregões da Corporação, além de ser utilizado pela maioria dos hospitais públicos, civis e militares do país, o que proporciona um banco de dados atualizado e robusto para utilização como balizador de preços. Ademais, esse portal utiliza um sistema de cadastro de materiais, o CATMAT, o que facilita muito a comparação. O MPOG possui uma equipe de cadastro de material, e as especificações são muito detalhadas e condizentes com a lógica comercial.

A operação do laboratório da SULAC/POMED requer a licitação de insumos que variam pouco ao longo do tempo, ou seja, habitualmente dispensa a inclusão de novos exames. Estes ocorrem apenas em situações excepcionais, como por exemplo, a atual crise sanitária provocada pela pandemia do Sars-Cov-2; dessa maneira, o cadastro existente do CATMAT atende perfeitamente.

Conforme descrito anteriormente, a utilização do portal do Ministério da Economia consome muito tempo para realização por se tratar de uma aplicação gráfica destinada ao usuário humano de computadores. Dada a complexidade do tema, a SULAC/POMED, em geral, opta por designar um

oficial farmacêutico para a realização da pesquisa, o que resulta em uma competição por recursos humanos entre as atividades de pesquisa de preço e as atividades finalísticas.

Com o advento da revolução tecnológica que o mundo vivencia atualmente, torna-se apropriado questionar se a tecnologia pode ajudar a otimizar os processos da SULAC/POMED. Uma aplicação que automatiza a pesquisa de preços pode agilizar a confecção dos processos licitatórios e de execução de contratos, liberando recursos humanos fundamentais para o desempenho da missão fim da organização. Logo, esta pesquisa se embasa no seguinte problema: a implantação de um algoritmo de automatização da tarefa é uma solução viável e de fato reduz o tempo de pesquisa de preços de materiais do Laboratório de Análises Clínicas da Policlínica Médica do CBMDF?

1.2 Justificativa

A pesquisa de preço em órgãos governamentais é um requisito legal para compor os processos de licitação conforme disposto na Lei Federal 8.666/1993, a lei das Licitações. Ela é disponibilizada por meio de interfaces gráficas disponibilizadas pelo Ministério do Planejamento, no âmbito federal, e por outras plataformas na esfera estadual.

O Decreto nº 39.453 de 14 de novembro de 2018 do governo do Distrito Federal, em seu art. 4º, orienta que a pesquisa de “preços públicos referentes a aquisições ou contratações similares realizadas pelo Distrito Federal e demais entes públicos”, junto com a pesquisa no portal de Notas Fiscais eletrônicas, é obrigatória na composição do balizamento de preços para licitação.

O decreto citado foi regulamentado pela Portaria Nº 514, de 16 de novembro de 2018, da Secretaria de Estado de Planejamento, Orçamento e Gestão do Distrito Federal, no que diz respeito aos procedimentos administrativos básicos para realização de pesquisa de preços na aquisição de bens e contratação de serviços em geral.

A plataforma de compras governamentais do governo federal é a mais completa base de dados de preços públicos disponível, devido ao grande cadastro de participantes e fornecedores. Esse cenário torna a plataforma federal uma fonte abastada e atualizada para a pesquisa de preços no CBMDF, inclusive a forma recomendada pela Diretoria de Aquisições de Materiais (DIMAT).

No entanto, a interface disponibilizada por meio de uma aplicação de pesquisa na internet, requer muito tempo de navegação entre as telas do sistema, demandando um tempo significativo para a pesquisa. Na experiência do setorial, o tempo gasto na pesquisa de preço é um problema que afeta os processos de novas contratações e renovações de contrato.

Além do tempo despendido, o trabalho de pesquisa de preços pode ter impactos em variáveis relacionadas à gestão de recursos humanos por diversos motivos: trata-se de uma tarefa repetitiva, facilmente dispersa, uma vez que é realizada via internet, e não se alinha com a missão fim da Corporação.

A pesquisa de preços é realizada anualmente para formação de preços em pelo menos 2 (dois) processos novos de aquisição de insumos e em 6 (seis) processos de renovação de contrato.

Ressalta-se que um trabalho desta natureza aloca o laboratório da Policlínica Médica em concordância com o Planejamento Estratégico da corporação, mais especificamente com o objetivo estratégico número 6 que visa garantir a infraestrutura apropriada às atividades operacionais e administrativas. A rotina atual demanda tempo e recursos humanos escassos que poderiam ser utilizados em processos mais críticos e intensivos em mão de obra, como o atendimento ao paciente.

Em suma, essa monografia apresenta importância estratégica ao discutir, estudar e propor uma solução estratégica para o controle de estoque de produtos laboratoriais para a Policlínica Médica do CBMDF, pois uma

solução que reduza o tempo gasto na pesquisa de preços públicos tornaria mais ágil a contratação de fornecedores e liberaria recursos humanos para realizar maior volume de tarefas técnicas.

1.3 Objetivos

1.3.1 Objetivo geral

Verificar a redução do tempo gasto na pesquisa de preços de materiais do Laboratório de Análises Clínicas da Policlínica Médica do CBMDF praticados em licitações constantes no Portal de Preços do Ministério do Planejamento, Orçamento e Gestão por meio de um algoritmo de automatização da tarefa.

1.3.2 Objetivos específicos

- Aferir o tempo gasto com a mesma tarefa utilizando a metodologia atual, de pesquisa manual dos preços na aplicação “Painel de Compras” do Ministério do Planejamento.
- Criar um algoritmo de *webscraping* para o site <paineldeprecos.planejamento.gov.br>.
- Adicionar à ferramenta nativa de controle de estoque do sistema de gestão laboratorial o cadastro de materiais do Ministério da Economia.
- Testar uma aplicação de busca de preços automatizada a partir da lista de demanda oriunda do sistema de gestão.
- Formatar um documento de pesquisa de preço como produto desta integração.
- Aferir o tempo gasto com a adoção do novo processo na aquisição de uma lista padronizada de ressuprimento.

1.4 Questões Norteadoras

A presente pesquisa será embasada pelas seguintes questões:

- É possível realizar uma rotina de automatização de tarefas de pesquisa de preços no painel de compras do Governo Federal?
- O sistema de gestão laboratorial existente admite a inserção do código de cadastro de materiais do Ministério do Planejamento (código CATMAT) à lista de materiais em estoque?
- É possível utilizar uma aplicação de automatização de tarefas para buscar os preços públicos oriundos da listagem de itens para reposição do estoque emitida pelo sistema de gestão laboratorial existente?
- A aplicação de automatização é capaz de emitir um documento de pesquisa de preço formatado nos moldes do que hoje é demandado pela gestão do CBMDF?
- Ao aplicar a rotina de automatização de tarefas, é possível aferir o tempo gasto na realização e compará-lo às rotinas existentes?

1.5 Definição de termos

A fim de facilitar a compreensão dos leitores, define-se os seguintes termos:

ACCOUNTABILITY: controle, fiscalização, responsabilização, prestação de contas.

API: sigla em inglês que significa Interface de Programação de Aplicações, estrutura padronizada de oferta de dados de um banco de dados via construção de aplicações.

BIG DATA: grande volume de dados estruturados e não estruturados que são gerados a cada segundo, fruto do desenvolvimento tecnológico recente que permitiu a obtenção e utilização dessa massa de dados.

CBMDF: instituição militar, organizada com base na hierarquia e na disciplina, em conformidade com as disposições legais, destina-se a realizar serviços específicos de bombeiros, na área do DF, incluindo extinção de incêndios, serviços de busca e salvamento, atividades de defesa civil e atendimento pré-hospitalar, entre outros.

SEI: Sistema Eletrônico de Informações, documenta os processos de grande parte dos órgãos da administração pública.

SCRIPT: Documento de texto especificando instruções para um computador escritas em uma linguagem de programação

WEBCRAPING: técnica utilizada para extrair, transformar e produzir informações a partir de dados existentes na internet.

2. REVISÃO DE LITERATURA

2.1 Automatização de Tarefas

Com o desenvolvimento tecnológico sem precedentes visto nos últimos anos, muitos questionamentos surgem em relação à abrangência e aos benefícios e prejuízos da automatização sobre os salários e empregos.

Após analisar mais de 800 ocupações e mais de 2000 atividades de trabalho, Chui (2016) afirma que a automatização vai eliminar poucas ocupações na próxima década, mas vai afetar a maioria delas; tudo vai depender do tipo de trabalho que elas desempenham, em especial nas áreas de saúde e finanças. A automatização vai ter impacto nas atividades ou tarefas, de forma que as ocupações terão que repensar o modo como atuam.

O CBMDF não foge ao destaque que Chui (2016) faz sobre o potencial de automação dos setores de saúde e finanças. Por se tratar de uma organização multitarefa e possuir um sistema próprio de saúde, é possível verificar a inserção da automatização de atividades no setor, como se vê na marcação de consultas e nos pedidos de ressarcimento. Portanto, é de se esperar que cada vez mais tarefas e práticas sejam automatizadas.

Em sua análise, Chui (2016) mostrou também que as ocupações analisadas gastam entre 64% e 78% do seu tempo em tarefas suscetíveis de serem automatizadas. Essa suscetibilidade é impactada por cinco fatores: viabilidade técnica: é possível automatizar a tarefa com a tecnologia disponível; custos da automação: custos envolvidos na mudança do processo em curso pelo processo automatizado; escassez relativa das habilidades: facilidade de se encontrar no mercado profissionais que realizam a tarefa. Quanto mais escasso, maior a recompensa pela automação; benefícios da automação além da substituição do trabalho: como a automação melhora a eficiência do processo, além da evidente redução do custo com salários; regulação e aceitação social: entraves jurídicos e atuação de entidades laborais.

Ainda segundo esse autor, as atividades com maior viabilidade de serem automatizadas envolvem Trabalho Físico Previsível (repetitivo), Processamento

de Dados e Coleta de Dados. O tempo gasto em todas as ocupações dos Estados Unidos por estes três grupos de tarefas equivale a 51% do tempo total gasto com todas as demais tarefas, como gestão, expertise aplicada, interação com *stakeholders* e trabalho físico não previsível (CHUI, 2016).

Na rotina da SULAC/POMED, há intensa demanda por Trabalhos Físicos Não Previsíveis e Expertise Aplicada, como coleta de sangue, análise visual por microscopia e supervisão de equipes de trabalho. Além disso, os oficiais do setor desempenham várias atividades que demandam relação com *stakeholders*, que envolvem atendimento ao público externo (reclamações, supervisão de fornecedores e controle de qualidade) e interno (demandas da ouvidoria interna, execução de contratos de fornecimento, participação em comissões, entre outras).

Cabe destacar que a pesquisa de preços públicos para licitação, objeto deste trabalho, envolve apenas tarefas de coleta e processamento de dados, categorias de tarefas que o estudo de Chui (2016) afirma que sua automatização é viável.

2.2 Gestão de Estoques

A gestão de estoques diz respeito à execução de uma série de atividades chave para otimizar o nível de ressuprimento de insumos de uma organização. Na visão de Peinado e Graeml (2007), por nível de ressuprimento entende-se uma quantidade de itens em estoque que permite manter o abastecimento durante o intervalo de processamento de uma nova aquisição.

Além disso, a esse nível adiciona-se uma medida de precaução, na forma de estoque de segurança. Trata-se de uma quantidade marginal de estoque para fazer frente a incertezas, como por exemplo, erros no processamento do pedido, crise de oferta do produto ou fracasso de licitação; tudo isso baseia-se em uma série de consumo passado.

Assim, a variável tempo é chave na gestão dos estoques: quanto maior for o tempo entre as aquisições, maior deverá ser o nível de ressuprimento.

Conforme assinala Ballou (2001), níveis maiores de estoque aumentam a exposição aos riscos do aumento dos custos do serviço de estocagem, como obsolescência, perda de validade, custos com seguro, aluguel de armazéns etc. Dessa maneira, uma aquisição realizada em menor tempo possibilita a diminuição da necessidade de grande volume de estoque, e, conseqüentemente, os custos associados.

No caso específico da atividade de pesquisa de preços, o Decreto nº 39.453, de 14 de novembro de 2018, do governo do Distrito Federal determina que os preços públicos pesquisados tenham validade máxima de 60 (sessenta) dias (DISTRITO FEDERAL, 2018). Esse prazo pode fazer com que os preços sejam invalidados até mesmo durante o processo de licitação, demandando a realização de uma nova pesquisa de preço.

O tempo gasto na atividade de pesquisa de preços é, portanto, um componente crítico da composição do tempo de ressuprimento dos materiais, em virtude da legislação determinar um período de validade. Não fosse isso, bastaria realizar uma única pesquisa de preços e consultá-la sob demanda.

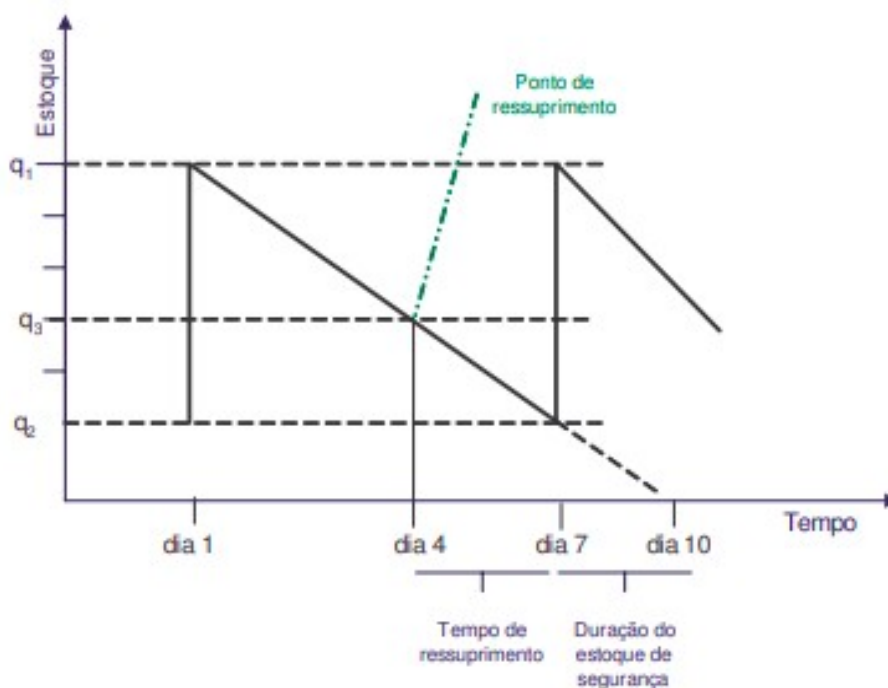
Para contornar a imprevisibilidade do tempo de consumo, uma das metodologias mais utilizada é a do Ponto de Ressuprimento, que pode ser obtido a partir da demanda média, adicionado de uma medida de desvio dita Estoque de Segurança. O ponto de ressuprimento pode ser calculado pela seguinte fórmula (PEINADO; GRAEML, 2007):

$$PR = (D \times TR) + ES$$

Onde: PR: Ponto de Ressuprimento, é o mesmo que Nível de Ressuprimento, quantidade de um item que aponta a necessidade de se iniciar um processo de aquisição; D: Demanda pelo material no período, obtida do consumo histórico; TR: Tempo de Ressuprimento, tempo entre o pedido e o ressuprimento; ES: Estoque de Segurança.

A dinâmica do controle do estoque, segundo a metodologia de Ponto de Ressuprimento, pode ser visualizada na Figura 1. Nela, percebe-se a reposição do estoque para o nível máximo, q_1 , após realização de pedido no nível do ponto de ressuprimento, q_3 , e atingir o nível mínimo de estoque, q_2 .

Figura 1 - Metodologia de Ponto de Ressuprimento



Fonte: Peinado e Graeml (2007, p. 467).

A título de exemplo, se a SULAC/POMED realiza 1.000 (mil) unidades do insumo “Teste para Colesterol” por mês, essa é a Demanda pelo material no período. O Tempo de ressuprimento quando se obriga um processo licitatório, na experiência do setorial, gira em torno de 6 (seis) e 12 (doze) meses, perfazendo uma média de 9 (meses). Essa informação impulsiona o gestor de estoque na melhor tomada de decisão, optando por um Estoque de Segurança de 6 (seis) meses, para fazer frente a esta imprevisibilidade. Dessa forma, o Ponto de Ressuprimento é igual a:

$$\text{PR} = (1.000 \times 9) + (1.000 \times 6) = 15.000 \text{ unidades do item.}$$

Quando o almoxarife (ou o sistema de informação) percebe que o nível do item “Teste para Colesterol” atingiu 15.000 (quinze mil) unidades, é o momento ideal de se iniciar um novo processo de compra.

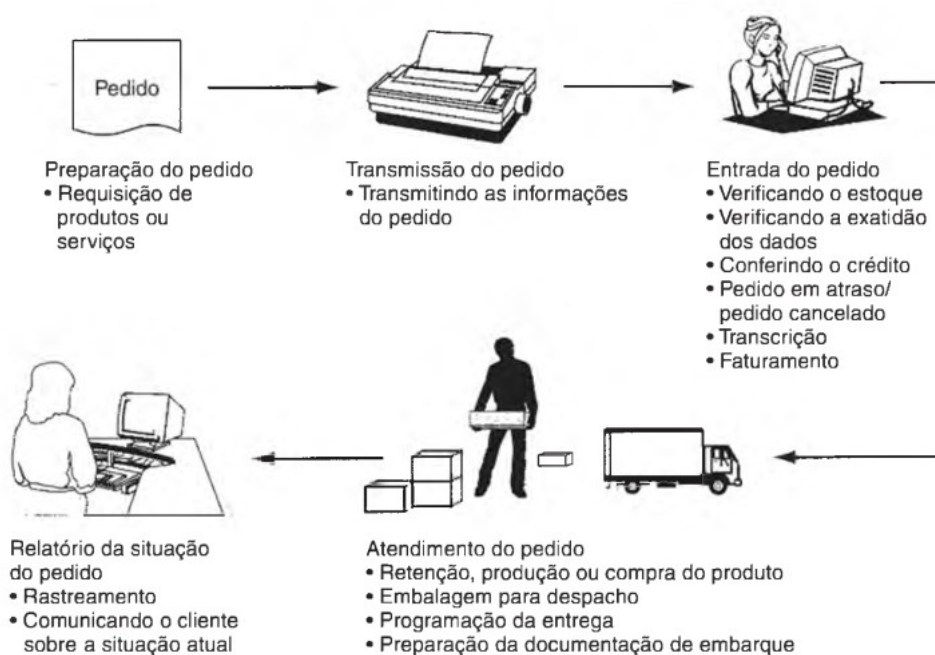
De acordo com Peinado e Graeml (2007), o estoque de segurança deve ser consumido durante o tempo de ressuprimento, isto é, não pode exceder a demanda média total estimada para esse período.

Ballou (2001) afirma que a manutenção de um estoque de segurança é um seguro contra a imprevisibilidade do tempo de ressuprimento. Dessa forma, depreende-se que o tempo de ressuprimento reduzido, além de ser a solução para evitar quebras de estoque, reduz o custo de estocagem ao diminuir o nível do estoque de segurança.

No entanto, essa proteção só é efetiva para itens que já foram consumidos e em quantidades esperadas, uma vez que a única fonte de dados para estimativa são os eventos passados. Logo, diante de um evento imprevisível e inesperado, como a pandemia de Sars-Cov-2, o estabelecimento de um estoque mínimo pode não ser uma medida efetiva para a manutenção do abastecimento do estoque.

O tempo de processamento de pedidos é uma etapa crítica do processo de ressuprimento (BALLOU, 2001). A Figura 2 esquematiza as etapas de processamento de um pedido, e é possível, por analogia ao processo descrito, inferir que a pesquisa de preços é parte da primeira etapa, a preparação do pedido.

Figura 2 - Processo de Processamento de Pedidos



Fonte: Ballou (2006, p. 122).

Como nos processos de aquisição de materiais realizados pelo CBMDF o tempo de ressuprimento é afetado em grande parte pela pesquisa de preços para estimativa do preço base para o processo licitatório, efeitos indesejados como o dito “efeito chicote” podem aparecer. Para Coelho, Manoel e Rodriguez (2009, p. 3):

O efeito chicote é o resultado de uma expectativa de demanda ou oferta que não se realiza, por diversos motivos, entre elas a incapacidade de prever a demanda dos clientes, e que se propaga por todas as empresas da cadeia, influenciando os níveis de estoques, os tamanhos dos pedidos e a produtividade.(Coelho, Manoel e Rodriguez, 2009, p. 3)

No âmbito da SULAC/POMED, o efeito chicote pode resultar em implicações sérias, pois quando a Corporação incorpora novos militares, não é possível prever quantos novos dependentes chegarão para o serviço de saúde, e o estoque planejado pode ser insuficiente. O que se pode fazer é estimar uma quantidade de novos dependentes baseada em dados da população de militares. Ainda assim, insere-se incerteza na previsão, possibilitando do aparecimento do “efeito chicote”.

Como esse fenômeno tem relação com o nível de incerteza em relação ao futuro, a ocorrência de fatos inesperados também podem causá-lo, mesmo que a previsão tenha sido feita corretamente. A pandemia de Sars-Cov-2, por exemplo, pressionou o estoque de Equipamentos de Proteção Individual como máscaras cirúrgicas e luvas de procedimento, a nível mundial.

2.3 A linguagem HTML

A linguagem HTML (Hypertext Markup Language), ou Linguagem de Marcação por Hipertexto, em uma tradução livre, é a linguagem de programação por onde todo o conteúdo da World Wide Web se expressa, e isso ocorre pelo chamado sistema de Hipertexto. Segundo Flatschart (2011), a linguagem de Hipertexto se caracteriza por conectar blocos de informações por meio de elementos de ligação, popularmente conhecidos como *links*.

Esse tipo de encadeamento de informações parece ser natural ao modo de processamento de informações pelo cérebro humano, caracterizado por não ser linear, mas sim operado por meio de grupos de dados. O experimento realizado pelo militar americano Vannevar Bush, denominado *Memory Extension*, ou MEMEX, foi considerado um marco fundamental na criação de modelos de hipertexto. Tratava-se de um sistema de indexação analógico de documentos físicos como folhas digitadas, mapas e fotografias (FLATSCHART, 2011).

O termo hipertexto é atribuído pelo autor citado ao filósofo americano Ted Nelson, que idealizou o ambicioso projeto Xanadu em 1960, ao construir uma maneira de interligar computadores por meio de protocolos simples de comunicação. O trabalho de Ted Nelson subsidiou, no final da década de 1980, o desenvolvimento chefiado por Tim Berners-Lee do protocolo HTTP (*Hypertext Transference Protocol*) (FLATSCHART, 2011), utilizado até os dias atuais para navegação na internet.

As chamadas Linguagens de Marcação, ou Hipertexto, já eram utilizadas extensivamente por revisores e editores de conteúdo para aprimorar a comunicação e trabalho em equipe. Com a consolidação da digitalização das informações, a partir da década de 1980, as linguagens de marcação permitiram “a interoperabilidade entre dispositivos, sistemas e plataformas distintas” (FLATSCHART, 2011, p. 408).

A HTML é a linguagem mais popular no desenvolvimento de aplicações web, e permite a exibição do conteúdo online vistos na atualidade, permitindo a exposição de imagens, vídeos, texto e tabelas. Ela é caracterizada pela inserção de dados em blocos chamados “*tags*”, fisicamente delimitados pelos caracteres “< >” (início do bloco) e </ > (fim do bloco), que permitem que se faça o endereçamento de todo o conteúdo exibido ao usuário.

O que este trabalho propõe é a obtenção de dados puros contidos nos códigos HTML dos sítios eletrônicos de pesquisa de preços do Governo Federal para transformá-los em informação útil, sem a necessidade de utilizar a

interface dos sítios por um usuário humano. Para tal, será utilizada uma técnica conhecida como *Webscraping*.

2.4 Webscraping

O objetivo geral deste trabalho demanda que se analise uma estratégia que reduza o tempo gasto na pesquisa de preço na execução dos processos licitatórios, como alternativa à utilização de recursos humanos operando uma interface gráfica. A estratégia que este trabalho se propõe a utilizar é a metodologia de *webscraping*.

Em uma tradução livre para o português, o verbo *to scrape* significa raspar, arranhar. Ou seja, a técnica visa “raspar” o conteúdo do arquivo HTML, e obter o que interessa ao pesquisador.

Segundo Haddaway (2015), *webscraping* descreve o uso de um programa ou algoritmo para extrair dados de arquivos HTML disponíveis na internet, tipicamente na forma de listas e tabelas.

Lawson (2015) afirma que se os dados obtidos na técnica de *webscraping* forem fatos públicos, podem ser republicados, o que não ocorre quando se trata de opiniões ou outro dado que possua propriedade intelectual.

Uma vez que a publicidade dos processos licitatórios é uma exigência legal, não se vislumbra impedimentos legais na obtenção destes dados. De fato, sua obtenção é determinada por decretos governamentais.

3. METODOLOGIA

3.1 Apresentação

Considerando que o objetivo deste estudo visa testar uma solução desconhecida e verificar sua aplicação em um caso específico, trata-se de uma pesquisa exploratória, que, de acordo com Gil (2002, p. 41), tem como objetivo “proporcionar maior familiaridade com o problema com vistas a torna-los mais explícitos ou a construir hipóteses”.

Dentre as técnicas e linguagens de programação gratuitas existentes, foi escolhida uma técnica de *webscraping* que utiliza a linguagem de programação *python* para interagir com o *drive* do navegador Google Chrome e acessar o portal de pesquisa de preço automaticamente.

Quanto ao método de pesquisa, utilizou-se o método dedutivo, uma vez que este estudo investiga soluções tecnológicas que resolvam problemas semelhantes de modo geral, acreditando-se que também possam servir ao problema de pesquisa. A técnica é utilizada recorrentemente para obtenção de dados não estruturados disponíveis em portais de órgãos públicos, como por exemplo, o trabalho de Azevedo (2019), que utilizou a técnica para obtenção de dados de empresas listadas na Bolsa de Valores de São Paulo fornecidos pela Comissão de Valores Mobiliários do Ministério da Fazenda.

No que se refere à natureza, trata-se de uma pesquisa aplicada, pois objetiva gerar conhecimentos para aplicações práticas dirigidas à solução de problemas específicos, neste caso, a redução do tempo gasto na pesquisa de preço para balizamento de licitação. De fato, a pesquisa pode ser caracterizada, no que diz respeito aos procedimentos técnicos executados, como um estudo de caso, pois será utilizada uma lista real de insumos demandada pela SULAC/POMED.

Esta pesquisa possui uma abordagem quantitativa ao determinar se a nova tecnologia diminui ou não o tempo do processo de pesquisa de preços. Dessa maneira, será comparada o tempo de execução da pesquisa de preço

no processo atual ao o tempo gasto com a pesquisa de preço por meio do algoritmo produzido. O resultado obtido em unidades de tempo permitirá medir a eficiência dos dois processos. Será aferido o tempo, em segundos, na pesquisa de itens mediante o processo atual, e comparado ao o tempo gasto, em segundo, aferido por meio do sistema durante a execução do *script* de *webscraping*.

3.2 Procedimentos

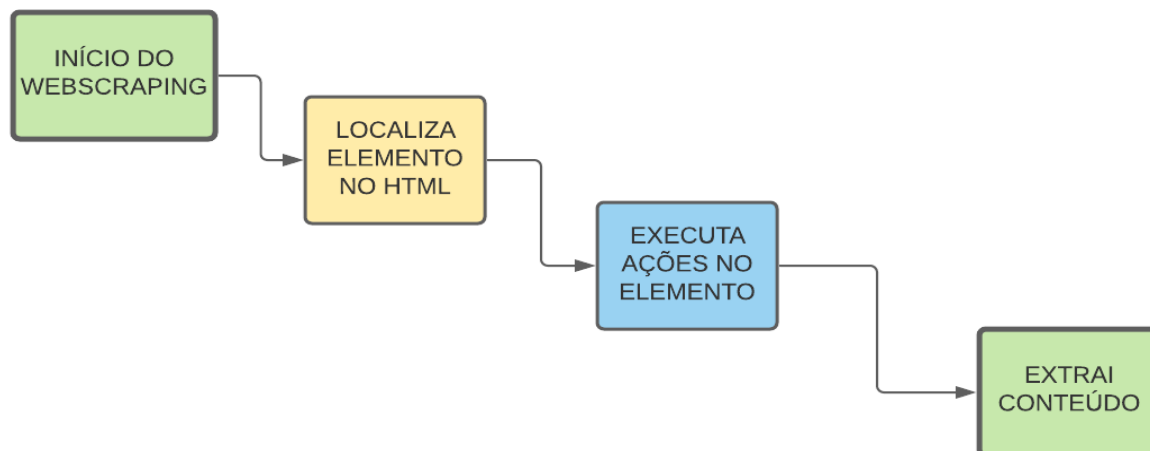
De acordo com os procedimentos utilizados nesta pesquisa, é possível classificá-la como pesquisa bibliográfica, experimental e estudo de caso. A pesquisa bibliográfica, presente no tópico de revisão de literatura, forneceu o arcabouço para o desenvolvimento dos outros dois aspectos da pesquisa.

Na parte experimental, mapeou-se o processo “as-is” e a mensuração do tempo foi realizada de duas formas. A primeira, medindo “in loco” a pesquisa realizada por militares do laboratório e a segunda, estimando os tempos gastos em cada tarefa do processo mapeado seja para execução de clicks, inserção de textos ou navegação.

O terceiro experimento foi a utilização do *webscrapping*. O algoritmo possui a funcionalidade que permite a aferição do tempo gasto na sua execução. Processos desse tipo, em geral, possuem uma sequência de execução em três etapas: localização do elemento do HTML, execução de ações no elemento e extração do conteúdo.

Os elementos do arquivo HTML são encontrados dentro da *tags*, elementos de código delimitados pelos caracteres ‘< >’ e ‘<\ >’. Cada *tag* possui uma classificação hierárquica que fornece uma espécie de endereço do elemento, chamado de ‘XPATH’.

Além de ocorrer por meio do ‘XPATH’, os elementos da *tag* podem ser acessados por atributos, que podem ser únicos de cada *tag* ou não. Esses atributos são denominados ‘name’, nome do elemento, ‘id’, identificação, ‘value’, valor, dentre outros que estão listados na documentação do módulo Selenium (SELENIUM, 2021). A Figura 3 apresenta um diagrama com o processo de execução do algoritmo de *webscraping*.

Figura 3 - Etapas do processo de *Webscraping*

Fonte: O autor.

Após localizar o elemento, o *driver* pode executar ações nesses elementos. O módulo utilizado para interagir com o *driver* navegador foi o Selenium, que permite ações como clicar, clicar e segurar, clique duplo, enviar valores a formulários, ou seja, permite emular completamente a atividade humana na navegação. As ações disponibilizadas por esse módulo do Selenium também podem ser acessadas na documentação do módulo (SELENIUM, 2021).

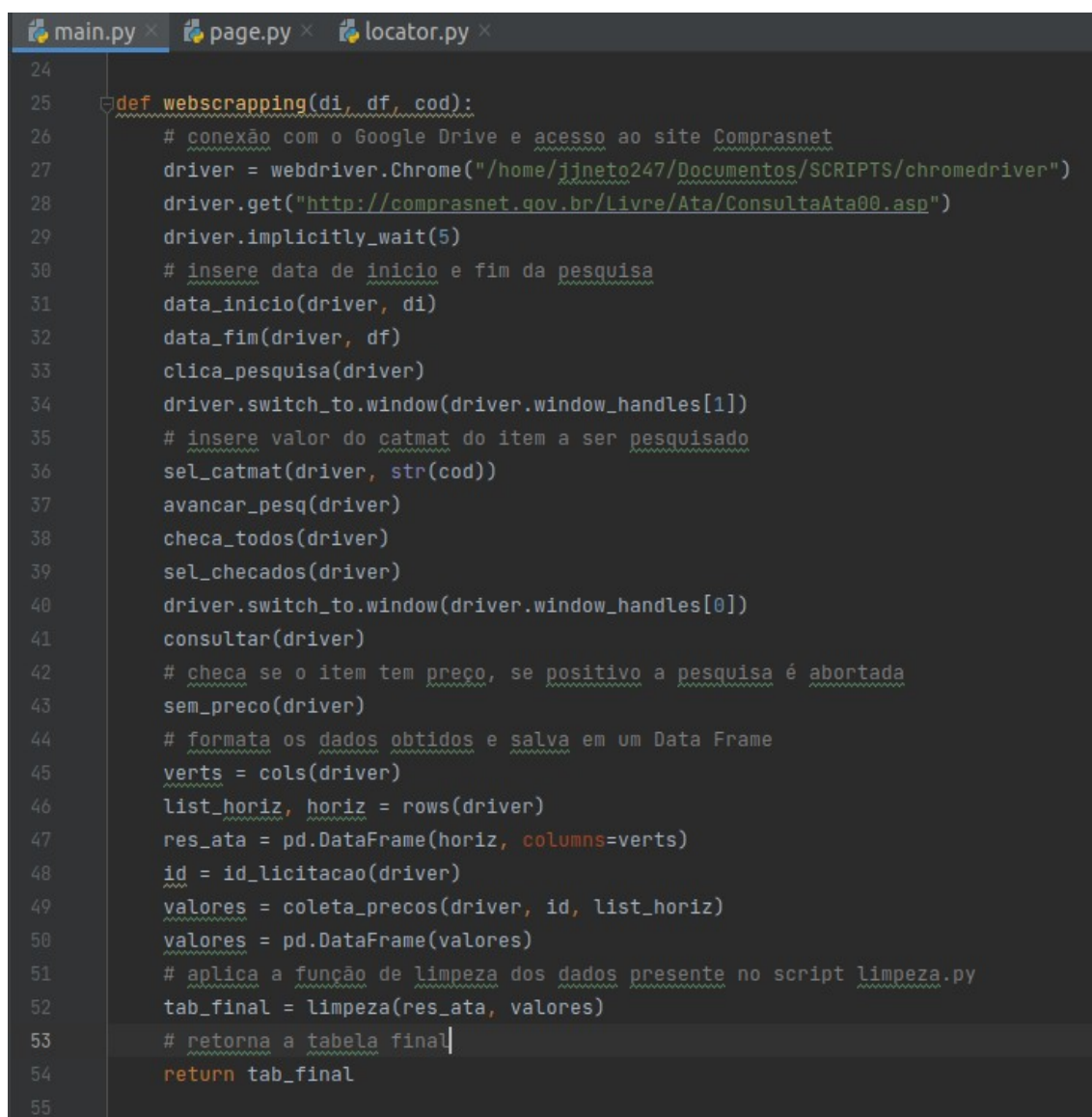
O algoritmo desenvolvido possui 5 scripts principais:

1. *main.py*: script principal e possui as seguintes funções: *webscraping* (recebe período de pesquisa e código catmat como entrada e realiza o procedimento de webscraping) e duas funções (*prep_df_pdf* e *imp_result*) de formatação.
2. *page.py*: realiza atividades como clicar, enviar valores e esperar utilizando o *drive*, tendo como entradas os localizadores do script *locator.py*.
3. *locator.py*: define todos os localizadores por informação disponível nas *tags* do arquivo HTML e os converte em objetos.

4. *limpeza.py*: limpa as informações, converte as variáveis em seu tipo correto (por exemplo, números obtidos como texto em formato numérico) e formata a tabela em condição de ser analisada.
5. *url_curta*: converte os *links* para os termos de homologação em endereços *url* curtos, facilitando a criação de tabelas menores.

O script *main.py* contém a função principal do processo de *webscrapping*, que simula a atuação de um ser humano na operação do portal Comprasnet, e pode ser visto na Figura 4.

Figura 4 - Script principal do algoritmo



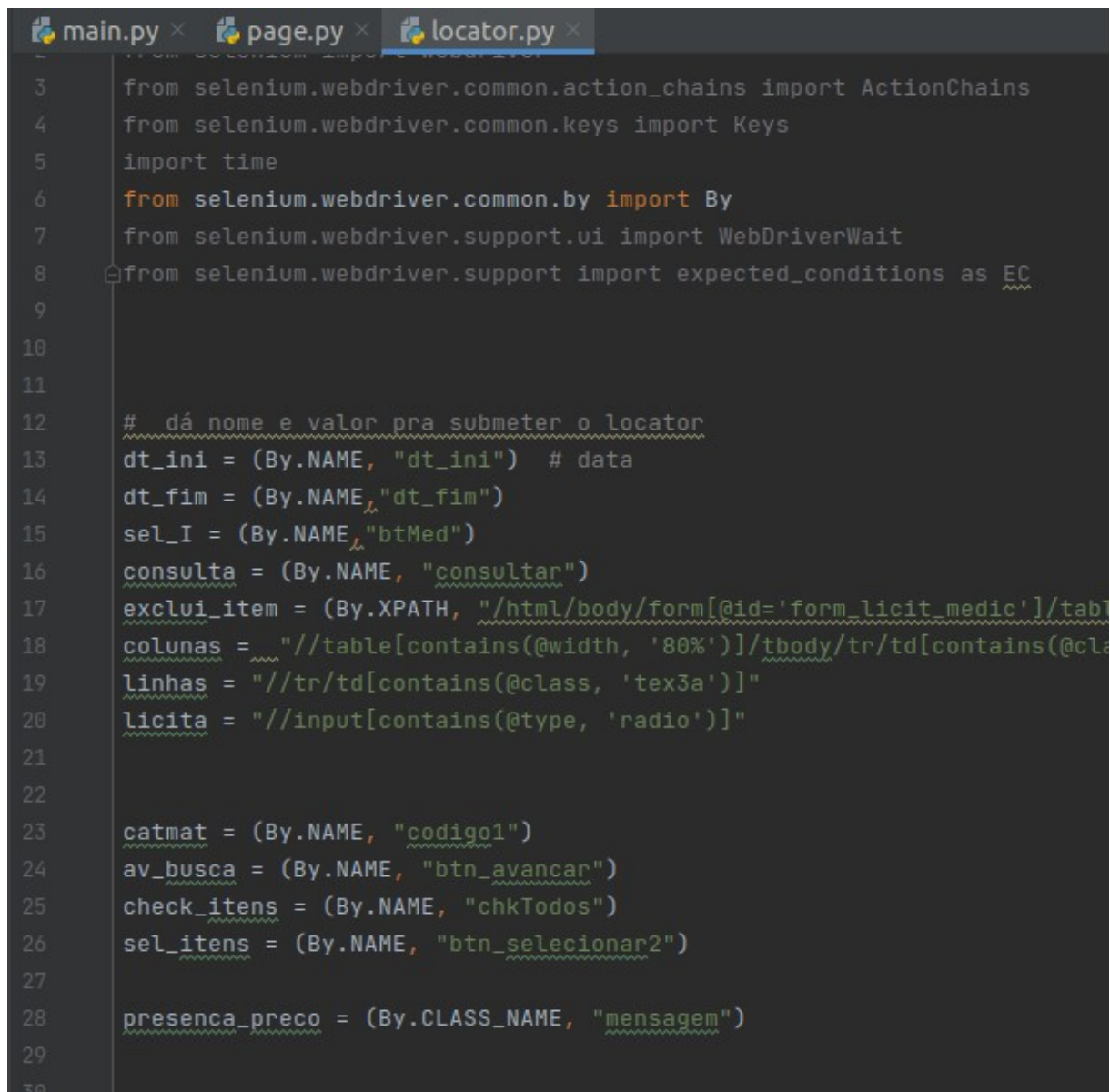
```

24
25 def webscrapping(di, df, cod):
26     # conexão com o Google Drive e acesso ao site Comprasnet
27     driver = webdriver.Chrome("/home/jjneto247/Documentos/SCRIPTS/chromedriver")
28     driver.get("http://comprasnet.gov.br/Livre/Ata/ConsultaAta00.asp")
29     driver.implicitly_wait(5)
30     # insere data de início e fim da pesquisa
31     data_inicio(driver, di)
32     data_fim(driver, df)
33     clica_pesquisa(driver)
34     driver.switch_to.window(driver.window_handles[1])
35     # insere valor do catmat do item a ser pesquisado
36     sel_catmat(driver, str(cod))
37     avancar_pesq(driver)
38     checa_todos(driver)
39     sel_checados(driver)
40     driver.switch_to.window(driver.window_handles[0])
41     consultar(driver)
42     # checa se o item tem preço, se positivo a pesquisa é abortada
43     sem_preco(driver)
44     # formata os dados obtidos e salva em um Data Frame
45     verts = cols(driver)
46     list_horiz, horiz = rows(driver)
47     res_ata = pd.DataFrame(horiz, columns=verts)
48     id = id_licitacao(driver)
49     valores = coleta_precos(driver, id, list_horiz)
50     valores = pd.DataFrame(valores)
51     # aplica a função de limpeza dos dados presente no script limpeza.py
52     tab_final = limpeza(res_ata, valores)
53     # retorna a tabela final
54     return tab_final
55

```

A função *webscrapping* importa funções presentes em outros *scripts*. A localização de todos os elementos da navegação do portal está presente no *script locator.py*, como mostra a Figura 5.

Figura 5 - Script contendo os localizadores dos elementos



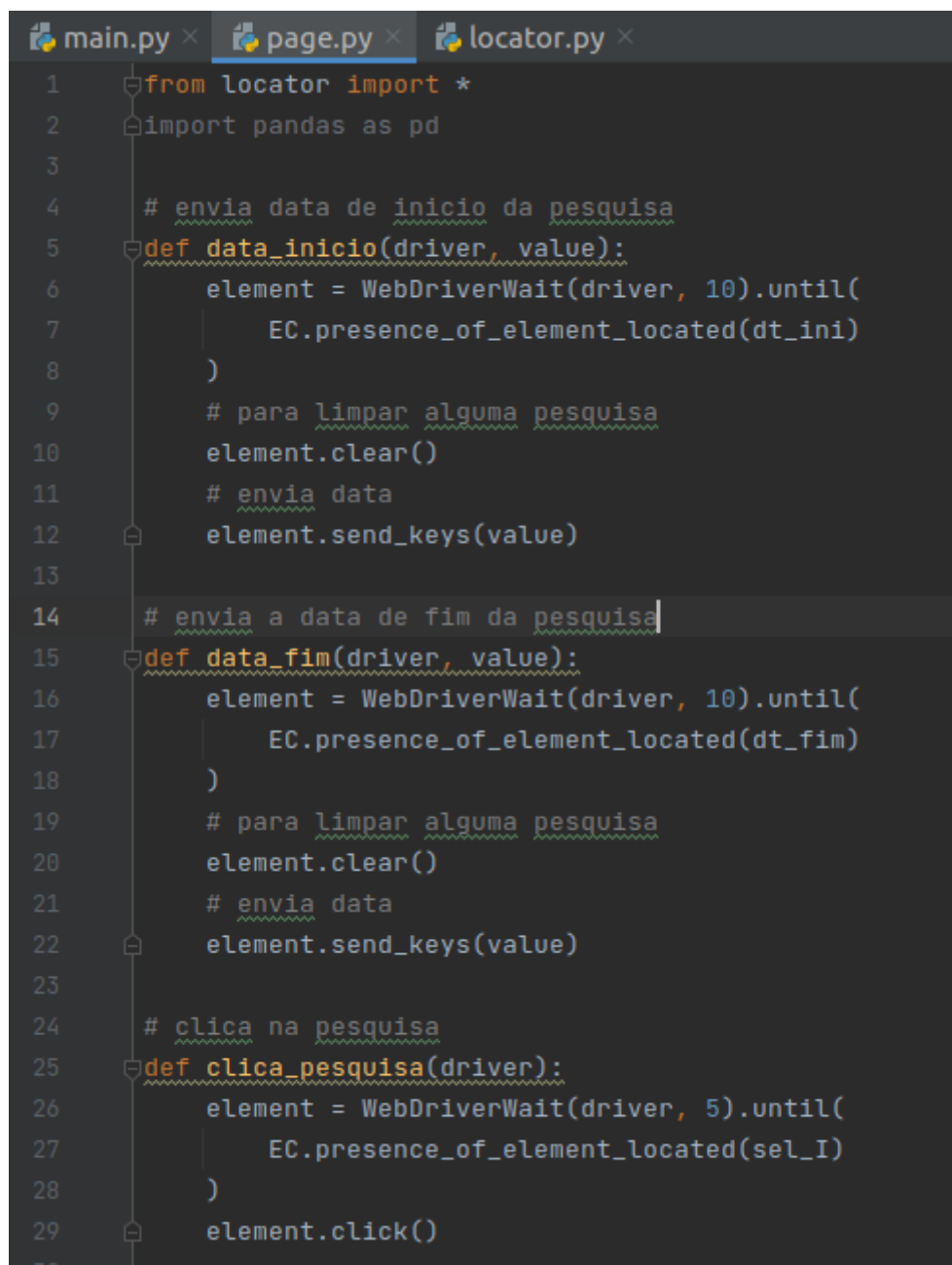
```

1  from selenium.webdriver.common.action_chains import ActionChains
2  from selenium.webdriver.common.keys import Keys
3  import time
4  from selenium.webdriver.common.by import By
5  from selenium.webdriver.support.ui import WebDriverWait
6  from selenium.webdriver.support import expected_conditions as EC
7
8  # dá nome e valor pra submeter o locator
9  dt_ini = (By.NAME, "dt_ini") # data
10 dt_fim = (By.NAME, "dt_fim")
11 sel_I = (By.NAME, "btMed")
12 consulta = (By.NAME, "consultar")
13 exclui_item = (By.XPATH, "/html/body/form[@id='form_licit_medic']/table")
14 colunas = "//table[contains(@width, '80%')]/tbody/tr/td[contains(@class, 'text-align: center;')]"
15 linhas = "//tr/td[contains(@class, 'text-align: center;')]"
16 licita = "//input[contains(@type, 'radio')]"
17
18 catmat = (By.NAME, "codigo1")
19 av_busca = (By.NAME, "btn_avancar")
20 check_itens = (By.NAME, "chkTodos")
21 sel_itens = (By.NAME, "btn_selecionar2")
22
23 presenca_preco = (By.CLASS_NAME, "mensagem")
  
```

Fonte: O autor

Uma vez listados todos os localizadores da navegação necessários para a execução da pesquisa de preço, foram elencadas todas ações a serem realizadas pelo *drive* no *script page.py*, como mostra a Figura 6.

Figura 6 - Script contendo as ações realizadas em cada localizador



```

1  from locator import *
2  import pandas as pd
3
4  # envia data de inicio da pesquisa
5  def data_inicio(driver, value):
6      element = WebDriverWait(driver, 10).until(
7          EC.presence_of_element_located(dt_ini)
8      )
9      # para limpar alguma pesquisa
10     element.clear()
11     # envia data
12     element.send_keys(value)
13
14     # envia a data de fim da pesquisa
15     def data_fim(driver, value):
16         element = WebDriverWait(driver, 10).until(
17             EC.presence_of_element_located(dt_fim)
18         )
19         # para limpar alguma pesquisa
20         element.clear()
21         # envia data
22         element.send_keys(value)
23
24     # clica na pesquisa
25     def clica_pesquisa(driver):
26         element = WebDriverWait(driver, 5).until(
27             EC.presence_of_element_located(sel_I)
28         )
29         element.click()
30

```

Fonte: O autor

Após a execução das ações, a função *webscraping* realiza a limpeza e formatação dos dados em um objeto do tipo Data Frame, uma estrutura de dados do tipo tabular ofertada pelo módulo *Pandas*, que oferece ferramentas de análise de dados para a linguagem de programação *python*. Ao final deste processo, obtém-se uma tabela com as seguintes variáveis:

- "TIPO_LICIT": Tipo de licitação. A Diretoria de Materiais preconiza a pesquisa de licitações do tipo Pregão.
- "NB_LICIT": Número do Pregão
- "QTD": Quantidade do insumo licitado na Licitação.
- "PRECO": Preço praticado na Licitação.
- "FIM_VIG": Fim da vigência da Ata de Registro de Preço. A Diretoria de Materiais preconiza a utilização de preços em uma janela inferior a um ano da publicação da Ata de Registro de Preço.
- "LINK": endereço do tipo *url* para acesso ao Termo de Homologação. A pesquisa de preço é comprovada com a impressão dos Termos de Homologação; com o fornecimento dos links, o conferente do processo pode acessar a validade da pesquisa sem precisar inserir dados em excesso no SEI (Sistema Eletrônico de Informações).
- "CATMAT": Código CATMAT do item pesquisado.

Por fim, a pesquisa realiza um estudo de caso frente a uma licitação em curso da SULAC/POMED, contendo 20 (vinte) itens para aquisição, e reporta o tempo total gasto para a execução da pesquisa de preço.

4. RESULTADOS E DISCUSSÃO

4.1 Desenvolvimento e Execução do Algoritmo

A verificação da possibilidade da execução de um algoritmo que realizasse a pesquisa de preços foi a primeira etapa deste trabalho. As fontes de preços *online* que a legislação vigente aceita e que a corporação adotou como rotina para este processo são: o Mapa de Preços de Notas Fiscais Eletrônicas do Governo do Distrito Federal e o Painel de Preços do Governo Federal.

A primeira fonte é ofertada sob uma aplicação na forma de *dashboard*, isto é, um painel de dados com uma interface gráfica aprimorada, voltada para a utilização do sistema por um usuário humano. Esse tipo de aplicação, embora seja familiar ao usuário de computador comum, é pouco flexível e requer muito tempo de pesquisa devido ao carregamento de inúmeros menus de pesquisa que demandam dados presentes no servidor.

Além disso, a pesquisa neste portal apresenta algumas dificuldades, pois aplicações com foco na interface gráfica possuem alguns problemas que decorrem justamente da tentativa de torná-las palatáveis ao usuário humano. Dessa forma, os *scripts* HTML são complexos e contém trechos dinâmicos de código, em geral na linguagem *Javascript*, que tornam o algoritmo de *web scraping* mais abstruso e, por ter mais tarefas, menos eficiente.

A aplicação do Governo Federal também segue a mesma linha de interface de usuário. No entanto, essa aplicação veio para substituir o antigo portal Comprasnet, que não foi descontinuado: ele é utilizado para outras tarefas do processo de licitação além da pesquisa de preços.

A forma como os dados das atas de registro de preço do sistema Comprasnet são disponibilizados é propícia para a automação via *web scraping*, por isso optou-se por utilizá-la. Além disso, o órgão disponibilizou uma *API* que facilitou a busca dos Termos de Homologação das licitações. Para tanto, lançou-se mão da linguagem de programação *python*, e do fato da mesma

possuir um módulo (um conjunto de scripts feitos por outros usuários) específico para essa tarefa, chamado Selenium.

O módulo Selenium é disponibilizado gratuitamente e pode ser importado para qualquer aplicação *python*, estando sua documentação disponível na internet. Junto com o módulo é disponibilizado um *drive* para que a aplicação opere um navegador, cujo escolhido foi o Google Chrome, que também é gratuito.

Como resultado, o algoritmo impetrou com sucesso uma planilha no formato *.xlsx* (Microsoft Excel) contendo todos os dados das licitações realizadas para o item de pesquisa e o período selecionado.

Para a obtenção da lista de reposição de estoque pelo Sistema de Gestão do Laboratório da POMED, consultou-se a empresa mantenedora que gerou um relatório no formato desejado. Assim, o laboratório utiliza a ferramenta do sistema para gerir o seu estoque, e no momento oportuno irá solicitar um relatório contendo quantidade de reposição e o código CATMAT dos itens a serem adquiridos: essa é a entrada inicial do algoritmo de busca.

O algoritmo foi utilizado a partir da obtenção da lista de reposição, contendo 20 itens padronizados pela SULAC/POMED para licitação (ANEXO A). Assim, o algoritmo demonstrou que pode utilizar a planilha gerada pelo sistema em uma situação real.

O algoritmo forneceu informações em condições de serem utilizadas no documento oficial de pesquisa de preços. O processo, que demandava abrir o documento no formato pdf contendo o Termo de Homologação, agora demanda apenas uma inserção entre duas tabelas Excel. Seria possível realizar um *script* que automatizasse esse processo, tendo como entrada um vetor de preços obtidos no Mapa de Preços de Notas Fiscais Eletrônicas do GDF e a planilha obtida por *webscraping*, para fins de comparação do tempo gasto não faria sentido para esta pesquisa.

Quanto ao desempenho, o algoritmo precisou de 26 (vinte e seis) minutos para realizar a pesquisa, conforme se observa na Tabela 1.

Tabela 1 – Tempo de desempenho do algoritmo

Número de itens	Tempo total	Tempo por item pesquisado
20	26 minutos	78 segundos

Fonte: O autor

4.2 Processo atual: estimativa através do mapeamento do processo

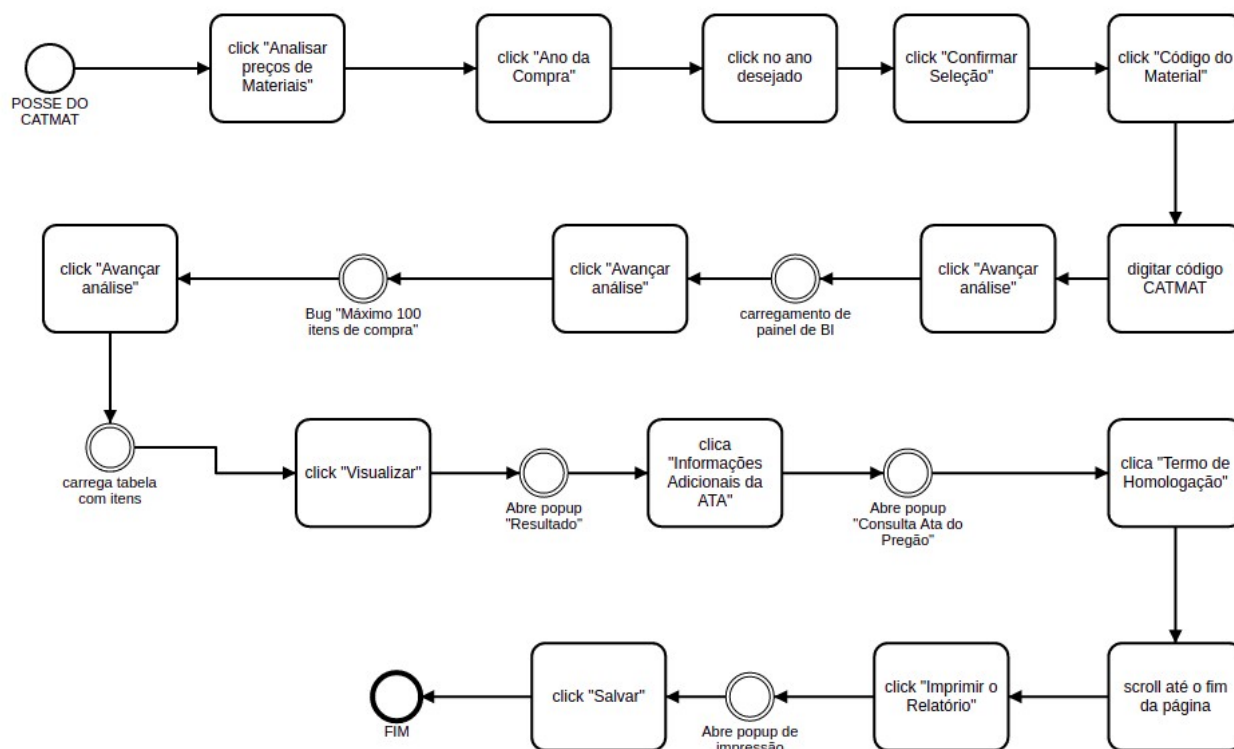
O processo que o algoritmo se propôs a automatizar é o processo que se executa após a obtenção da lista de ressuprimento contendo as quantidades e os códigos CATMAT da aquisição, e se opera manualmente o portal Painel de Preços do MPOG.

O processo necessita de 12 *clicks* de mouse que demandam, em média e condições normais de fluxo da internet da corporação, 3 segundos por *click*. Também requer 5 (cinco) carregamentos de *popup*, que nas mesmas condições demandam em torno de 5 (cinco) segundos para carregamento.

Além disso, a aplicação de pesquisa de preço possui um *bug* na etapa de acesso à planilha de pregões que requer a repetição do *click* “Avançar Análise”. O problema pode se repetir por várias vezes, mas considerando a ocorrência mínima de 1 (uma) vez, incrementa 10 segundos ao processo; em caso de persistência do erro, é necessário reiniciar o processo recarregando a página.

Ao imprimir o Termo de Homologação, é necessário fazer um *scroll* para chegar ao final da página, aumentando 3 segundos no processo. A Figura 7 apresenta um mapeamento deste processo.

Figura 7 - Processo atual de pesquisa de preços no Painel de Preços



Fonte: O autor

Dessa maneira, nas condições ideais de velocidade da rede, são gastos 36 (trinta e seis) segundos em *clicks*, 25 (vinte e cinco) segundos aguardando a abertura de *popups* e 10 (dez) segundos no *bug* do painel. Destarte, a pesquisa de um item de licitação demanda 71 (setenta e um) segundos para sua execução.

Caso o encarregado da pesquisa necessite obter um outro preço da consulta, ele tem que retornar ao passo “Visualizar”, o que acrescentaria 6 (seis) cliques e 3 (três) aberturas de *popup*. Para um novo item da licitação que se desejar obter o seu Termo de Homologação, deve-se acrescentar 33 (trinta e três) segundos ao processo.

Como a legislação exige três preços para o balizamento, e um obrigatoriamente obtido da Mapa de Preços da SEFAZ/DF, são necessários 2 (dois) Termos de Homologação. Assim, o tempo total necessário para se obter a pesquisa de dois preços no portal Painel de Preços, nas condições ótimas de operação, é de 104 (cento e quatro) segundos.

Aferiu-se o desempenho utilizando uma lista de ressuprimento contendo 20 itens, emitida pelo sistema de gestão da SULAC/POMED (APÊNDICE A). O desempenho estimado pelo processo mapeado está descrito na Tabela 2.

Tabela 2 – Experimento 1: Estimativa do tempo pelo processo mapeado

Item pesquisado	Tempo por item pesquisado	Tempo por item pesquisado
20	35 minutos	104 segundos

Fonte: O autor

No entanto, salienta-se que o algoritmo buscou todos os preços disponíveis para o item no período em questão que demonstra a saída de execução do programa, conforme se observa na Figura 8. É possível melhorar o desempenho do algoritmo produzindo um filtro para os valores de pesquisa, que pode ser realizado em outra etapa de desenvolvimento.

Figura 8 - Tabela final do algoritmo com aferição do tempo de execução

TIPO_LICIT	NB_LICIT	QTD	PREÇO	FIM_VIG	
0	Pregão	00012/2020	2.000	4,5600	16/04/2021
1	Pregão	00002/2020	500	16,9000	31/07/2021
2	Pregão	00006/2020	1.200	14,0000	10/07/2021
3	Pregão	00016/2020	1.000	12,8500	09/11/2021
			LINK	CATMAT	
0		https://tinyurl.com/y5nz4hyw		429299	
1		https://tinyurl.com/y4wmz987		429299	
2		https://tinyurl.com/yyu8jsqk		429299	
3		https://tinyurl.com/yxjoea3u		429299	
0:00:47.840185					

Fonte: O autor

Observou-se que o algoritmo demonstrou realizar a tarefa com mais rapidez, fornecendo uma quantidade maior de dados do que a pesquisa feita pelo ser humano. Assim, caso o usuário do portal quisesse obter todos esses preços, seria necessário retornar alguns passos e gastar mais tempo selecionando os demais itens.

4.3 Experimento 2: Execução real do processo atual

Há que se levar em conta que a tarefa realizada pelo ser humano está suscetível a distrações, cansaço e interrupções. Para observar esses efeitos, foi solicitado a 3 (três) militares da SULAC/POMED a realização da pesquisa dos mesmos 20 itens em tempo real, utilizando a interface de usuário do Portal de Compras. O resultado pode ser observado na Tabela 3.

Tabela 3 - Aferição de tempo gasto em pesquisa de preços de 20 itens no portal Painel de Preços

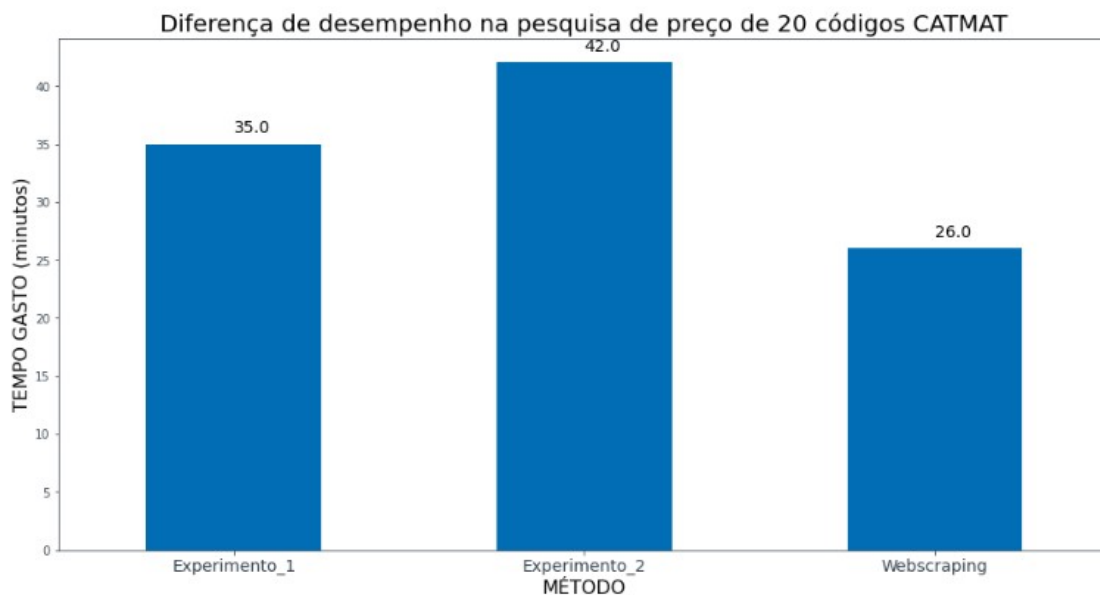
Quantidade de militares	Número de itens pesquisado	Tempo médio total gasto	Tempo médio gasto por item (segundos)	Necessidade de folga (em minutos, média)
3	20	37	131	26

Fonte: O autor

Conforme observado na tabela acima, os militares relataram a necessidade de interromper a atividade a cada 26 (vinte e seis) minutos, devido à exaustão causada pela tarefa repetitiva. Considerando uma interrupção mínima de 5 minutos. O processo atual demandou 42 minutos.

Nesse experimento, a utilização do algoritmo reduziu o tempo gasto na pesquisa de 20 (vinte) itens em 16 (dezesesseis) minutos, uma redução de 38%.

Assim, a pesquisa de preços dos 20 (vinte) itens foi realizada mediante três experimentos: tempo gasto estimado no portal Painel de Preços (2021) baseado no mapeamento do processo (Experimento 1), tempo gasto no portal Painel de Preços (2021) em situação real (Experimento 2) e tempo gasto com a pesquisa por *webscraping* (Experimento 3). O desempenho desses três experimentos são apresentados na Figura 9.

Figura 9 - Desempenho aferido na pesquisa de preço - Experimentos

Fonte: O autor

As condições ótimas aqui comparadas levam em conta um profissional absolutamente focado, com nível alto de energia e imune a interferências externas, o que nem sempre é a realidade do ambiente de trabalho.

No caso dos oficiais farmacêuticos da SULAC/POMED, a rotina envolve supervisão de coleta de sangue, assinatura de laudos, atendimento ao público e execução de exames. Essas atividades são essenciais e o ambiente de trabalho é agitado e barulhento, o que sem dúvida impactaria no tempo de obtenção do preço público.

4.4 Discussão dos Resultados

Além do fato do algoritmo ter apresentado performance superior, há que se salientar que a obtenção de preços pelo algoritmo e pelo usuário humano via interface de usuário não são tarefas concorrentes. É possível programar a execução do algoritmo diariamente, e abastecer um banco de dados de preços de materiais padronizados.

Para os materiais padronizados na SULAC/POMED, a pesquisa de um ano realizada pelo algoritmo gera um arquivo de tamanho aproximado de 56 *kilobytes*, não impactando a disponibilidade de memória do servidor. Dessa maneira, elimina-se 100% do tempo gasto com a extração dos dados no banco de dados nativo do sistema de compras governamentais, e os preços estarão disponíveis sob demanda para a realização de processos de aquisição.

Mesmo diante de resultados positivos, torna-se necessário alertar sobre possíveis problemas com a utilização do algoritmo de *webscraping*. O principal deles é a possibilidade do mantenedor do portal de pesquisa incluir empecilhos para a utilização da aplicação por algoritmos, na forma de janelas de confirmação do tipo *captcha*.

Ainda que as informações sejam públicas, esta é uma possibilidade que não se pode descartar. Ela acarretaria a necessidade de alteração do algoritmo, de forma que um ser humano assinale a ação demandada pelo *captcha* e permita a continuidade da execução.

No entanto, a oferta de dados em melhores condições poderia alavancar a automatização de tarefas e, por consequência, a produtividade dessas tarefas. A oferta da Secretaria de Fazenda do Distrito Federal de uma aplicação para transferência de dados das Notas Fiscais Eletrônicas permitiria automatizar essa tarefa, atualmente, considerada bastante difícil.

A iniciativa do portal “dados.gov.br” de criar uma *API* de consulta a licitações é um grande facilitador e um avanço na iniciativa de disponibilizar dados públicos na internet, sendo utilizada no desenvolvimento do algoritmo. No entanto, ela ainda não permite a descoberta dos itens, órgãos e números dos processos licitatórios, tarefas que foram feitas por *webscraping*, mas foi indispensável na etapa de captura dos Termos de Homologação. Não há dúvida que se trata de um avanço na iniciativa de disponibilizar dados públicos na internet.

5. CONSIDERAÇÕES FINAIS

Esse trabalho teve como foco uma característica contemporânea do mercado de trabalho: a substituição do trabalho humano por algoritmos de automatização de tarefas. Demonstrou-se que um algoritmo desse tipo pode realizar operações que demandam tempo e atenção de militares no âmbito do CBMDF.

O novo processo demonstrou ainda maior eficiência no tempo gasto com pesquisa. Assim, realizou o mesmo trabalho sem a necessidade de supervisão humana e em menos tempo do que levaria com o processo atual de pesquisa de preços.

O algoritmo tem potencial para ser melhorado e se tornar mais eficiente. Pode também ser adicionado a outras automatizações de processos relacionados ou absorver novas tarefas.

Na realidade do CBMDF, várias atividades estão sendo automatizadas graças a iniciativas tanto de origem externa, como o SEI, como internas, como o Inova (automatização de processos administrativos). Em comparação à situação de apenas dez anos atrás, a mudança é drástica.

Ademais, as ferramentas utilizadas neste trabalho estão disponíveis gratuitamente na internet, portanto sua implementação não acarreta custos de aquisição à instituição. No entanto, a manutenção preventiva, corretiva e adaptativa, esta última para inserção de melhorias no algoritmo, requer o emprego de recursos humanos.

Segundo a Brasscom (TIC2019, 2019), Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação, entre 2020 e 2024 o número de vagas de trabalho em Tecnologia da Informação deve chegar a 420 mil, enquanto surgem cerca de 42 mil novos profissionais a cada ano. Esse cenário impõe um desafio, não só ao CBMDF, mas a todos os órgãos públicos, que é a criação de quadros técnicos robustos para dar vazão à crescente necessidade de serviços de tecnologia de informação.

Por outro lado, grande parte do serviço administrativo hoje executado é potencialmente possível de ser automatizado. Dessa maneira, a automatização dos processos pode ser importante para aumentar a disponibilidade de mão de obra nos processos finalísticos da Corporação que são poucos propensos à automatização.

Sendo assim, a automatização da tarefa de pesquisa de preços públicos tem como benefícios a redução da necessidade de recursos humanos, a redistribuição de recursos humanos para atividades finalísticas e a otimização do tempo gasto em pesquisa de preços.

6. RECOMENDAÇÕES

Recomenda-se que a Corporação continue com o processo de automatização de tarefas. As atividades de extração e consulta de dados na internet podem ser implementadas por meio de *webscraping*.

Também seria proveitoso o entendimento junto à Secretaria de Fazenda do Distrito Federal no intuito de ofertar uma *API* de consulta aos dados das Notas Fiscais Eletrônicas. Isso permitiria a automatização desta outra etapa do processo de pesquisa de preço e reduziria ainda mais o tempo de processamento do pedido.

O aumento dos serviços ligados à tecnologia de informação demanda o ingresso de recursos humanos especializados para implantar, monitorar e manter os sistemas próprios da Corporação. Sugere-se que a Corporação priorize a entrada destes profissionais, seja por meio de concursos públicos, seja por terceirização.

Na área de Saúde, é fundamental que a Corporação desenvolva uma aplicação de prontuário eletrônico para sistematizar a obtenção de dados da população de usuários. Isso permitiria a previsão mais acertada de níveis de estoque de materiais e oferta de serviços.

REFERÊNCIAS

BRASIL. **Lei nº 8.255, de 20 de novembro de 1991.** Dispõe sobre a organização básica do Corpo de Bombeiros Militar do Distrito Federal e dá outras providências. Brasília: Presidência da República, 1991. Disponível em: http://www.planalto.gov.br/ccivil_03/Leis/L8255.htm. Acesso em: 04 dez. 2020.

BRASIL. **Lei nº 8666, de 21 de junho de 1993.** Regulamenta o art. 37, inciso XXI, da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências. Diário Oficial da União - Seção 1 - 22/6/1993, 8269p.

DISTRITO FEDERAL. **Decreto nº 39.453, de 14 de novembro de 2018.** Regulamenta a Lei distrital nº 5.525, de 26 de agosto de 2015, que estabelece que, em compras e contratações de bens e serviços, qualquer que seja a modalidade de licitação, o valor a ser pago não seja superior à média de preços do mercado, no âmbito do Distrito Federal. Brasília: Governo do Distrito Federal, 2018. Disponível em: http://www.buriti.df.gov.br/ftp/diariooficial/2018/11_Novembro/DODF%20218%2016-11-2018/DODF%20218%2016-11-2018%20INTEGRA.pdf. Acesso em: 17 fev. 2021.

DISTRITO FEDERAL. **Portaria Nº 514, de 16 de novembro de 2018.** Regulamenta os procedimentos administrativos básicos para realização de pesquisa de preços na aquisição de bens e contratação de serviços em geral na forma do Decreto Distrital nº 39.453, de 14 de novembro de 2018. Brasília: Governo Federal, 2018. Disponível em: http://www.sinj.df.gov.br/sinj/Norma/075b64fe392842daae14a8c7d79f119c/Portaria_514_16_11_2018.html. Acesso em: 17 fev. 2021.

CORPO DE BOMBEIROS MILITAR DO DISTRITO FEDERAL. **Manual para normatização de trabalhos acadêmicos.** Ed. rev. Brasília: CBMDF, 2019.

AZEVEDO, Lucas Lyon de. **Uma análise exploratória de dados aplicado ao mercado financeiro.** Trabalho de Conclusão de Curso. Universidade Federal do Rio Grande do Norte: Natal, 2019.

BALLOU, Ronald H. **Gerenciamento da Cadeia de Suprimentos.** Ed. Bookman: Porto Alegre, 2001.

BALLOU, Ronald H. **Gerenciamento da cadeia de suprimentos/logística empresarial.** Tradução Raul Rubenich. 5ª ed. Ed. Bookman, Porto Alegre, 2006.

BOWERSOX, Donald J.; CLOSS, David J. **Logistical Management.** McGraw-Hill, New York, 1996.

CHUI, Michael et al. **Where machines could replace humans-and where they can't (yet)**. 2016. Disponível em: <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/where-machines-could-replace-humans-and-where-they-cant-yet#>. Acesso em: 10 fev. 2021.

COELHO, Leandro Callegari; FOLLMANN; Neimar, RODRIGUEZ, Carlos Manuel Taboada. **O impacto do compartilhamento de informações na redução do efeito chicote na cadeia de abastecimento**. Gestão & Produção, 2009. Disponível em: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2009000400007#:~:text=Em%20outras%20palavras%2C%20o%20efeito,tamanhos%20dos%20pedidos%20e%20a. Acesso em: 10 fev. de 2021.

COMPRASNET. **Sistema de Compras Governamentais**: Consulta a Atas de Registro de Preço. Disponível em: <http://comprasnet.gov.br/Livre/Ata/ConsultaAta00.asp>. Acesso em: 17 fev. 2021.

DADOS.GOV.BR. **Portal Brasileiro de Dados Abertos**. Disponível em: <https://dados.gov.br/>. Acesso em: 17 fev. 2021.

FLATSCHART, Fábio. **HTML 5 - Embarque Imediato**. Ed. Brasport: Rio de Janeiro, 2011.

GARCIA, Eduardo Saggioro, et al. Cálculo do ponto de pedido baseado em previsões de uma política de gestão de estoques. **Pesquisa Operacional**, 2009. Disponível em: <https://www.scielo.br/pdf/pope/v29n3/a09v29n3.pdf>; Acesso em: 18 fev. 2021.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4ª Ed. Editora Atlas. São Paulo, 2002.

HADDAWAY, Neal R. **The use of web-scraping software in searching for grey literature**. Grey J, v. 11, n.3, p. 186-90, 2015.

LAU, Hon-Shiang. Toward a Stock Control System under Non-Normal Demand and Lead-time Uncertainty. **Journal of Business Logistics**, 10(1), 88-103. 1989.

LAWSON, Richard. **Web scraping with Python**. Packt Publishing Ltd, 2015.

MAPADEPRECOS. **Painel Mapa de Preços da SEFAZ/DF**. Disponível em: <http://qap.fazenda.df.gov.br/mapa-de-precos/>. Acesso em: 17 fev. 2021.

MICHAELIS. **Dicionário**. Disponível em: <http://michaelis.uol.com.br> Acesso em: 17 fev. 2021.

PEINADO, Jurandir; GRAEML, Alexandre Reis. Administração da produção. **Operações industriais e de serviços**. Unicenp: Curitiba, 2007.

PINTO, Luiz Felipe da Silva, et al. **Estratégias de integração e utilização de bancos de dados nacionais para avaliação de políticas de saúde no Brasil**. Tese de Doutorado. Fundação Oswaldo Cruz - Escola Nacional de Saúde Pública, departamento de Ciências Sociais: Rio de Janeiro, 2006.

PAINEL DE PREÇOS. **Painel de Preços do Ministério da Economia**. Disponível em: <https://paineldeprecos.planejamento.gov.br/>. Acesso em: 17 fev. 2021.

ROCHA, Arlindo Carvalho (2011). **Accountability na administração pública: modelos teóricos e abordagens**. Journal of Accounting, Management and Governance, p. 82-97. Disponível em: <https://www.revistacgg.org/contabil/article/view/314>. Acesso em 30 mar. 2021.

SELENIUM. **Selenium with Python**. Disponível em: <https://selenium-python.readthedocs.io/>. Acesso em: 17 fev. 2021.

TIC2019. **Relatório Setorial de TIC 2019**. Disponível em: <https://brasscom.org.br/relatorio-setorial-de-tic-2019/>. Acesso em: 17 fev. 2021.

ANEXOS

**ANEXO A - LISTA DE RESSUPRIMENTO FORNECIDA PELO SISTEMA
LABOR TI, DE PROPRIEDADE DA SULAC/POMED**

CATMAT	DESCRIÇÃO	QTD
413720	Hemograma completo	27300
382699	Tempo de Tromboplastina Parcial Ativada (TTPA)	2100
442070	Fibrinogênio Dosado 10X5 ML	300
442196	Tromboplastin, determinação quantitativa do Tempo de Protrombina (TP)	2300
442071	D-Dímero	300
352715	AFP (ALFAFETOPROTEINA)	100
389758	CA 125	500
392019	CA 15-3	400
382633	CA 19-9	300
392009	CEA	600
433774	CK MASSA	200
378030	PSA LIVRE	6400
378032	PSA TOTAL	1300
345602	TROPONINA	100
363438	NT-proBNP - PEPTÍDEO NATRIURÉTICO CEREBRAL	300
340717	ANTI TG (ANTIT)	2000
442236	ANTI TPO	2400
378044	BETA HCG	1300
378031	CORTISOL	800
378016	Estradiol	1600

APÊNDICES

APÊNDICE A - COMPARATIVO ENTRE PROCESSO AUTOMATIZADO E O PROCESSO ATUAL

Item	Tempo Decorrido Total	Cálculo da Variação entre Pesquisas	Tempo Gasto por Item	Tempo processo antigo	Ganho / Perda (segundos)
1	00:00:42.900276	0	00:00:42.900276	00:01:34	61
2	00:02:33.865982	00:00:42.900276	00:01:50.965706	00:01:34	-7
3	00:03:26.708367	00:02:33.865982	00:00:52.842385	00:01:34	51
4	00:05:04.099150	00:03:26.708367	00:01:37.390783	00:01:34	-3
5	00:06:03.294188	00:05:04.099150	00:00:59.195038	00:01:34	45
6	00:06:31.217228	00:06:03.294188	00:00:27.923040	00:01:34	76
7	00:09:37.332100	00:06:31.217228	00:03:06.114872	00:01:34	-82
8	00:09:59.410918	00:09:37.332100	00:00:22.078818	00:01:34	82
9	00:14:01.385325	00:09:59.410918	00:04:01.974407	00:01:34	-138
10	00:14:31.186386	00:14:01.385325	00:00:29.801061	00:01:34	74
11	00:15:18.814387	00:14:31.186386	00:00:47.628001	00:01:34	57
12	00:16:06.222689	00:15:18.814387	00:00:47.408302	00:01:34	57
13	00:17:11.933323	00:16:06.222689	00:01:05.710634	00:01:34	39
14	00:17:45.525882	00:17:11.933323	00:00:33.592559	00:01:34	70

15	00:18:06.414430	00:17:45.525882	00:00:20.888548	00:01:34	84
16	00:20:49.676521	00:18:06.414430	00:02:43.262091	00:01:34	-59
17	00:22:01.342310	00:20:49.676521	00:01:11.665789	00:01:34	32
18	00:23:21.985918	00:22:01.342310	00:01:20.643608	00:01:34	24
19	00:24:31.127133	00:23:21.985918	00:01:09.141215	00:01:34	35
20	00:26:07.532366	00:24:31.127133	00:01:36.405233	00:01:34	8

APENDICE B: ALGORITMO DESENVOLVIDO

Script main.py:

```

from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.keys import Keys
import datetime
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from page import *
import pandas as pd
#from limpeza import *
from limpeza_2 import *
from url_curta import *
import url_curta
#import plotly.graph_objects as go
#from bokeh.io import export_png, export_svgs
#from bokeh.models import ColumnDataSource, DataTable, TableColumn
'''
di = "01012020"
df = "01112020"
cod = "351675"
'''
def webscrapping(di, df, cod):
    # conexão com o Google Drive e acesso ao site Comprasnet
    driver = webdriver.Chrome("/home/jjneto247/Documentos/SCRIPTS/chromedriver")
    driver.get("http://comprasnet.gov.br/Livre/Ata/ConsultaAta00.asp")
    driver.implicitly_wait(5)
    # insere data de inicio e fim da pesquisa
    data_inicio(driver, di)
    data_fim(driver, df)
    clica_pesquisa(driver)
    driver.switch_to.window(driver.window_handles[1])
    # insere valor do catmat do item a ser pesquisado
    sel_catmat(driver, str(cod))
    avancar_pesq(driver)
    checa_todos(driver)

```

```

sel_checados(driver)
driver.switch_to.window(driver.window_handles[0])
consultar(driver)
# checa se o item tem preço, se positivo a pesquisa é abortada
sem_preco(driver)
# formata os dados obtidos e salva em um Data Frame
verts = cols(driver)
list_horiz, horiz = rows(driver)
res_ata = pd.DataFrame(horiz, columns=verts)
id = id_licitacao(driver)
valores = coleta_precos(driver, id, list_horiz)
valores = pd.DataFrame(valores)
# aplica a função de limpeza dos dados presente no script limpeza.py
tab_final = limpeza(res_ata, valores)
# retorna a tabela final
return tab_final
#tab_final.to_csv("tab_final.csv")
driver.close()

def prep_df_pdf(df):
    lista = df.loc[:, ["NB_UASG", "NB_LICIT"]]
    lista = lista.values.tolist()
    lista_limpa = []
    for row in lista:
        for col in range(len(row)):
            row[col] = row[col].strip()
        lista_limpa.append(row)
    lista = lista_limpa
    lista_links = []
    for i, j in lista:
        link_homo = "http://comprasnet.gov.br/livre/pregao/termohom.asp?
prgcod=817208&co_no_uasg={}&numprp={}&\"
        \"f_lstSrp=&f_Uf=&f_numPrp={}&f_coduasg={}&f_tpPregao
=E&\"
        \"f_lstICMS=&f_dtAberturaIni=&f_dtAberturaFim=\".format(i, j,
j, i)
        lista_links.append(link_homo)
    return lista_links
# na tabela resultado vou acrescentar a url curta e selecionar colunas
# "TIPO LICIT", "NB LICIT", "QTD", "PRECO", "FIM_VIG"

```

```

def imp_result(res, lista_links, cod):
    df = res[["TIPO_LICIT", "NB_LICIT", "QTD", "PRECO", "FIM_VIG"]]
    temp = []
    for i in lista_links:
        curta = make_shorten(i)
        temp.append(curta)

    #temp_series = pd.Series(temp, index = df.columns)
    #df = df.append(temp_series, ignore_index = True)
    df["LINK"] = temp
    df["CATMAT"] = cod

    #df.QTD = df.QTD.str.replace(".", "").astype(int)
    #df.PRECO.astype(float)
    #df["QTD"] = df["QTD"].astype(int)
    #df["CATMAT"] = pd.to_numeric(df["CATMAT"])
    #print(df.dtypes)
    return df

def convert(row):
    #print(row)
    return '{}'.format(row['LINK'])
    #href=" ">{}< , row.name

if __name__ == '__main__':
    begin_time = datetime.datetime.now()
    di = "15012020"
    df = "15012021"
    ""

    # para buscar um só preço
    cod = "429299"
    resultado = webscrapping(di, df, cod)
    #tabula = tabela(resultado, "tabela_text.png")
    lista_links = prep_df_pdf(resultado)
    final = imp_result(resultado, lista_links, cod)
    #print(resultado, lista_links)
    print(final)
    #limpo = final_filter(final)

    ""

    # tarefa para todos os catmats:
    catmat = pd.read_csv("/home/jjneto247/Documentos/SCRIPTS/catmat.csv")
    catmat.columns = ["CATMAT", "DESC", "QTD"]

```

```

codigos = catmat["CATMAT"]
dimat = []
for codigo in codigos:
    resultado = webscrapping(di, df, codigo)
    lista_links = prep_df_pdf(resultado)
    final = imp_result(resultado, lista_links, codigo)
    lista = final.to_numpy().tolist()
    for row in lista:
        dimat.append(row)
    print(datetime.datetime.now() - begin_time)

colunas = ["TIPO_LICIT", "NB_LICIT", "QTD", "PRECO", "FIM_VIG", "LINK",
"CATMAT"]
dimat = pd.DataFrame(dimat, columns= colunas)
#dimat.to_excel("pesquisa_abo.xlsx")
catmats = pd.read_csv('catmat_preco_medio_horm_2021.csv')
catmats.columns = ["GRUPO", "ITEM", "CATMAT", "ESPECIF", "QTD", "AVG"]
dimat['LINK'] = dimat.apply(convert, axis=1)
dimat['LINK'].head()
# filtro de preços
mergido = pd.merge(dimat, catmats, on='CATMAT')
mergido = mergido[mergido['PRECO'] <= 2 * mergido['AVG']]
mergido.to_excel("pesquisa_2001.xlsx")
print(datetime.datetime.now() - begin_time)

```

Script locator.py:

from selenium.webdriver.common.by import By

```

from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
# dá nome e valor pra submeter o locator
dt_ini = (By.NAME, "dt_ini") # data
dt_fim = (By.NAME, "dt_fim")
sel_l = (By.NAME, "btMed")
consulta = (By.NAME, "consultar")

```

```

exclui_item = (By.XPATH,
"/html/body/form[@id='form_licit_medic']/table[@id='TABLE1']/tbody/tr[4]/
td[@class='TdTable']/table[@id='TABLE1']/tbody/tr[3]/td[2]/table[@class='campo']/
tbody/tr/td[3]/input[@id='button']")
colunas = "//table[contains(@width, '80%')]/tbody/tr/td[contains(@class, 'tex3b')]"
linhas = "//tr/td[contains(@class, 'tex3a')]"
licita = "//input[contains(@type, 'radio')]"
catmat = (By.NAME, "codigo1")
av_busca = (By.NAME, "btn_avancar")
check_itens = (By.NAME, "chkTodos")
sel_itens = (By.NAME, "btn_selecionar2")
presenca_preco = (By.CLASS_NAME, "mensagem")

```

Script page.py:

from locator import *

```

import pandas as pd
# envia data de inicio da pesquisa
def data_inicio(driver, value):
    element = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located(dt_ini)
    )
    # para limpar alguma pesquisa
    element.clear()
    # envia data
    element.send_keys(value)
# envia a data de fim da pesquisa
def data_fim(driver, value):
    element = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located(dt_fim)
    )
    # para limpar alguma pesquisa
    element.clear()
    # envia data
    element.send_keys(value)
# clica na pesquisa
def clica_pesquisa(driver):
    element = WebDriverWait(driver, 5).until(
        EC.presence_of_element_located(sel_l)
    )

```

```

        element.click()
# consulta as licitações
def consultar(driver):
    element = WebDriverWait(driver, 5).until(
        EC.presence_of_element_located(consulta)
    )
    element.click()
# pode ser que não exista preços do ítem selecionado
def sem_preco(driver):
    l = driver.find_elements_by_class_name("mensagem")
    # get list size with len
    s = len(l)
    # check condition, if list size > 0, element exists
    if (s > 0):
        m = l.text
        print("Element does not exist -" + m)
        driver.close()
    else:
        print("Element does exist")
# pega o nome das colunas
def cols(driver):
    driver.implicitly_wait(5)
    columns = driver.find_elements_by_xpath(colunas)
    column_info = []
    for column in columns:
        column_info.append(str(column.text))
    column_info.pop(0)
    return column_info
def rows(driver):
    driver.implicitly_wait(5)
    lines = driver.find_elements_by_xpath(linhas)
    dados_linha = []
    for linha in lines:
        dados_linha.append(str(linha.text))
    dados_linha = [s for s in dados_linha if s != ""]
    lista_linhas = []
    lista_linhas = [dados_linha[i: i + 4] for i in range(0, len(dados_linha), 4)]
    return dados_linha, lista_linhas
# identificadores das licitações

```

```

def id_licitacao(driver):
    driver.implicitly_wait(5)
    element = driver.find_elements_by_xpath(licita)
    id_licita = []
    for el in element:
        id_licita.append(el.get_attribute("value"))
    return id_licita

# coleta dos dados
def coleta_precos(driver, id_licita, dado_linha):
    price = []
    xpath = []
    for i in id_licita:
        b = "//tr/td[contains(@class, 'tex3a')]/input[contains(@value , '' + str(i) +
        '')]"
        xpath.append(b)
    for j in xpath:
        driver.find_element_by_xpath(str(j)).click()
        driver.implicitly_wait(5)
        driver.find_element_by_name('ok').click() # vai pra pesquisa de preço
        driver.implicitly_wait(5)
        # tentativa de contornar pesquisas com dois preços
        try:
            driver.find_element_by_name('Nulitem').click()
            driver.find_element_by_name('ok').click()
            dados_price = driver.find_elements_by_class_name("tex3a")
            for campo in dados_price:
                price.append(campo.text)
            driver.implicitly_wait(5)
            driver.find_element_by_id('button1').click()
            driver.find_element_by_id('button1').click()
            driver.implicitly_wait(5)
        except:
            dados_price = driver.find_elements_by_class_name("tex3a")
            for campo in dados_price:
                price.append(campo.text)
            driver.implicitly_wait(5)
            driver.find_element_by_id('button1').click() #volta da pesquisa de
            preço
            driver.implicitly_wait(5)

```

```

price = [s for s in price if s != "1"]
price = [price[i: i + 5] for i in range(0, len(price), 5)]
for el_price, el_id_licita in zip(price, id_licita):
    el_price.append(el_id_licita)
return price

# define catmat
def sel_catmat(driver, value):
    element = WebDriverWait(driver, 5).until(
        EC.presence_of_element_located(catmat)
    )
    # para limpar alguma pesquisa
    element.clear()
    # envia data
    element.send_keys(value)

def avancar_pesq(driver):
    element = WebDriverWait(driver, 5).until(
        EC.presence_of_element_located(av_busca)
    )
    element.click()

def checa_todos(driver):
    element = WebDriverWait(driver, 5).until(
        EC.presence_of_element_located(check_itens)
    )
    element.click()

def sel_checados(driver):
    element = WebDriverWait(driver, 5).until(
        EC.presence_of_element_located(sel_itens)
    )
    element.click()

```

Script limpeza.py:

import pandas as pd

```

data = pd.read_csv("res_data.csv")
datatwo = pd.read_csv("valores.csv")
def limpeza(data, datatwo):
    #res_ata: dataframe 1 -> colocar nomes, coluna identificador, ['licitação'] split by \n
    por " ",
    #data é um DataFrame
    data.drop(data.columns[0], axis = 1, inplace = True)

```



```

novo = data.iloc[:, 0].str.split("-", n= 1, expand = True)
data['NB_ORGAO'] = novo.iloc[:, 0]
data['ORGAO'] = novo.iloc[:, 1]
data.drop(columns = data.columns[0], inplace = True)
novo2 = data.iloc[:, 0].str.split("-", n= 1, expand = True)
data['NB_UASG'] = novo2.iloc[:, 0]
data['UASG'] = novo2.iloc[:, 1]
data.drop(columns=data.columns[0], inplace=True)
novo3 = data.iloc[:, 0].str.split("\n", n= 1, expand = True)
data['TIPO_LICIT'] = novo3.iloc[:, 0]
data['NB_LICIT'] = novo3.iloc[:, 1]
data.drop(columns=data.columns[0], inplace=True)
novo4 = data.iloc[:, 0].str.split("\n", n= 1, expand = True)
data['VIGENCIA'] = novo4.iloc[:, 1]
data.drop(columns=data.columns[0], inplace=True)
# limpando o DF valores
datatwo.drop(datatwo.columns[0], axis = 1, inplace = True)
novo = datatwo.iloc[:, 0].str.split(" - ", n= 1, expand = True)
datatwo['CNPJ'] = novo.iloc[:, 0]
datatwo['FORNECEDOR'] = novo.iloc[:, 1]
datatwo.drop(columns = "0", inplace = True)
datatwo.columns = ['QTD', 'PRECO', 'VL_TOTAL', 'MARCA', 'ID', 'CNPJ',
'FORNECEDOR']
# concatenando os dois dfs
df = pd.concat([data, datatwo], axis = 1)
return df
tab_final = limpeza(data, datatwo)
print(tab_final.columns)
#tab_final.to_csv("tab_final.csv")

```

Script url_curta.py:

#Importing the required libraries

```

from __future__ import with_statement
import contextlib
try:
    from urllib.parse import urlencode
except ImportError:
    from urllib import urlencode
try:

```

```

        from urllib.request import urlopen
except ImportError:
        from urllib2 import urlopen
import sys
#Defining the function to shorten a URL
def make_shorten(url):
    request_url = ('http://tinyurl.com/api-create.php?' + urlencode({'url':url}))
    with contextlib.closing(urlopen(request_url)) as response:
        return response.read().decode('utf-8')
#The main function to receive user inputs
def main():
    for shortyurl in map(make_shorten, sys.argv[1:]):
        print(shortyurl)
if __name__ == '__main__':
    url_ex = 'http://comprasnet.gov.br/livre/pregao/termohom.asp?prgcod=817208&co_no_uasg=120625&numprp=00063/2019&f_lstSrp=&f_Uf=&f_numPrp=00063/2019&f_coduasg=120625&f_tpPregao=E&f_lstICMS=&f_dtAberturaIni=&f_dtAberturaFim='
    url_short = make_shorten(url_ex)
    print(url_short)

```