Desenvolvimento de ferramenta para busca avançada em acervos de arquivos PDF

Adriano Vergínio Sinigaglia[1], **Taís Cristine Appel Colvero**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Veranópolis – BR-470, Km 172, 6500, Bairro Sapopema
CEP 95330-000 – Veranópolis – RS – Brazil

{adrianosinigaglia07@gmail.com, tais.colvero@veranopolis.ifrs.edu.br}

Abstract. *Developed by Adobe in the 90's, the Portable Document Format (PDF) is, since 2008, standardized by ISO 32000-1. Due to its growing use, it is common to create document collections, which makes it difficult to carry out searches in their contents. For this reason, there is a need to use tools to facilitate the search process. In this way, using the Java programming language together with the Java Swing framework and the Apache Tika library, a new advanced search tool was developed in PDF document collections. The development was carried out through an evaluation of existing tools, analysis and software design, implementation and subsequent validation of the tool.*

Resumo. *Desenvolvido pela Adobe nos anos 90, o Portable Document Format (PDF) é, desde 2008, padronizado pela ISO 32000-1. Devido ao seu crescente uso, se torna comum a criação de acervos de documentos, o que acaba dificultando a realização de buscas em seus conteúdos. Por este motivo, surge a necessidade da utilização de ferramentas para facilitar o processo de busca. Desta maneira, utilizando a linguagem de programação Java em conjunto com o framework Java Swing e a biblioteca Apache Tika, desenvolveu-se uma ferramenta de busca avançada em acervos de documentos PDF. O desenvolvimento foi realizado após uma avaliação de ferramentas existentes, análise e projeto de software, implementação e posterior validação da ferramenta.*

1. Introdução

Atualmente, o PDF (Portable Document Format) é um dos formatos de arquivos mais usados para o compartilhamento de informações [Acosta-Vargas et al. 2020]. Neste contexto, é comum aos pesquisadores e acadêmicos, de uma maneira geral, a criação de acervos de publicação que se resumem, sumariamente, em pastas com arquivos PDF, por vezes, de forma hierarquizada. Como o número de arquivos trocados e publicados na internet está crescendo constantemente e a troca de documentos eletrônicos está se tornando cada vez mais popular entre os usuários da internet [Castiglione et al. 2010], surge então a necessidade de ferramentas que permitam ao usuário, de forma objetiva, buscar informações nestes diretórios. Diversas ferramentas auxiliam nesta tarefa, tais como os gerenciadores de arquivos dos sistemas (Windows Explorer, Finder, Nautilus), utilitários em ferramentas de manipulação ou visualização de PDFs (Adobe Acrobat Pro DC, PDF Studio, FoxIt PhantomPDF, SmallPDF) ou especificadamente para a busca em acervos de arquivos (UltraEdit UltraFinder). Apesar de serem ferramentas extremamente úteis, há

uma lacuna com respeito a busca contextual, ou seja, a busca em que o escopo é restringido a uma parcela do conteúdo do texto, e não do arquivo como um todo. Em vista desta problemática, foi proposto o desenvolvimento de uma ferramenta de busca avançada em acervos de documentos PDF, com recurso para busca no escopo de frases.

As próximas seções do artigo dispõem da seguinte estrutura: na Seção 2, são apresentados os principais referenciais teóricos, mostrando o que atualmente é abordado em relação ao tema proposto; na Seção 3, é feito um estudo e projeto do software, onde é realizada uma análise contextual e levantamento de requisitos para posteriormente realizar o desenvolvimento do software proposto; na Seção 4, os resultados do desenvolvimento são apresentados, mostrando a interface obtida, o método de uso, as bibliotecas testadas, como foi feita uma validação do software e exibindo um comparativo com os softwares similares existentes; na Seção 5, é levantada uma discussão sobre o trabalho desenvolvido; e por fim, na Seção 6, são apresentados os trabalhos futuros que podem ser desenvolvidos para aprimorar a ferramenta.

2. Fundamentação teórica

A especificação do formato PDF tornou-se gratuita pela Adobe Systems em 1993, mantendo-se um formato proprietário controlado pela empresa até 2008, quando foi estabelecido um padrão aberto pela ISO 32000-1:2008, publicado pela Organização Internacional de Normalização [ISO 2008]. Com essa mudança, o padrão passou a ser disponível para acesso e implementação, independentemente de royalties e sem restrições de uso [Simcoe 2006]. O formato PDF é hoje um padrão para arquivos de texto, usado tanto pelo público em geral quanto por organizações, e tendo seu uso intensificado cada vez mais [Acosta-Vargas et al. 2020, Mladenov et al. 2019, Castiglione et al. 2010].

De acordo com Castiglione et. al (2010), a principal vantagem do PDF é que o formato permite que os documentos criados a partir de diferentes softwares de editoração eletrônica possa ser visualizado da mesma maneira, independentemente do sistema e aplicação em que estiver sendo exibido.

Segundo [Wives and Sardi 1996], a busca de informações específicas, contidas em documentos textuais extensos é uma tarefa que demanda tempo. Para facilitar tal operação, é proposto em seu trabalho o desenvolvimento de uma ferramenta que armazene palavras-chave de documentos (formato RTF) em um dicionário de dados com o objetivo de gerenciar estes documentos, permitindo achar a informação desejada com rapidez, obtendo-se um aumento na produtividade. Ferramentas de buscas em cenários distintos estão sendo desenvolvidas e aprimoradas a fim de auxiliar em buscas textuais, [Gil et al. 2015] propôs uma ferramenta para buscas complexas em códigos fontes, a ferramenta permite a busca nas estruturas utilizando apenas o valor sintático da consulta ou agregando valor semântico e com isso melhorando seus resultados.

A pesquisa recente em torno do tema sobre o formato de arquivos PDF aborda diversos temas, tais como a incorporação de conteúdo tri-dimensional nos documentos [Azkue 2021, Chung et al. 2020], extração e validação de dados tabulados [Bhatia et al. 2021, Mikhailov et al. 2020], aprendizado de máquina a partir de documentos PDF [Kostrinsky-Thomas et al. 2021], acessibilidade [Zulfiqar et al. 2020, Shelton and Yu 2020, Acosta-Vargas et al. 2020] e segurança [Gopaldinne et al. 2021, Hossain and Ayub 2020, He et al. 2020, Singh et al. 2020, Mladenov et al. 2019]. Al-

guns destes temas condizem com o uso do PDF como um padrão no processo de submissão e revisão de literatura científica [Castiglione et al. 2010].

3. Métodos

3.1. Projeto

Na fase de modelagem do software, o projeto detalhado foi realizado com base na análise dos requisitos, incluindo um projeto da arquitetura do software. A fim de visualizar as lacunas dos programas disponíveis atualmente e desenvolver a ferramenta proposta, foi realizada uma avaliação das ferramentas existentes, elencando-se suas capacidades de busca, de modo a culminar numa análise e projeto de software para, posteriormente, realizar o desenvolvimento.

Visualizando a Tabela 1, é possível examinar a avaliação da capacidade de busca das ferramentas comparativas, observando como principais lacunas exploráveis nos softwares existentes a ausência de um escopo restrito a uma parte do texto, a ausência de busca por sinônimo e a incapacidade de busca por mais de uma expressão regular nos softwares que suportam esse tipo de busca.

O software proposto opera uma arquitetura em três camadas, como ilustrado na Figura 1, a seguir descritas:

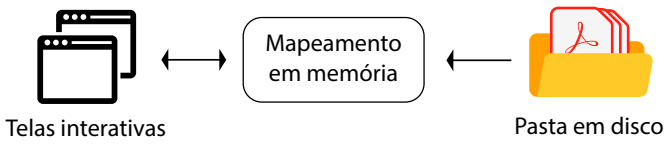

Figura 1. Arquitetura básica do software

1. Camada de apresentação: A camada de apresentação engloba o processo interativo, por exemplo: enquanto a biblioteca realiza a varredura, e o usuário a definição de parâmetros e consulta, cabe a camada de apresentação posteriormente, apresentar a exibição dos resultados. Aos resultados é incorporada uma discriminação de parcialidade para resultados positivos utilizando um índice de conformidade, permitindo ao usuário a classificação por este parâmetro.
2. Camada de mapeamento de dados: Esta camada é responsável pela instância dos dados oriundos da camada de dados, em memória, ocorre o processamento da lógica correspondente aos critérios selecionados pelo usuário, e pela geração de dados para apresentação correspondente às buscas realizadas. Finalmente, os resultados do processamento são refletidos para o usuário por meio da camada de apresentação. A abordagem de varredura prévia dos diretórios, diferentemente de uma leitura sob demanda, permite que se mantenha a indexação em memória, o que torna as buscas ágeis e permite visualizações rápidas de contexto, dando prioridade ao processo interativo sobre o custo de uso de memória.
3. Camada de dados: A camada de dados se trata essencialmente do acesso ao sistema de arquivos, onde realiza-se a varredura dos arquivos do formato PDF de uma pasta, opcionalmente de forma recursiva.

Para auxiliar no desenvolvimento do software, elaborou-se o diagrama de classes, ilustrado na Figura 2. Com o uso desse diagrama, é possível visualizar todas as classes presentes no projeto, bem como uma representação da estrutura do sistema proposto e as relações entre os objetos.

As principais classes presentes no diagrama da Figura 2 são:

- `TelaInicial`: nesta classe, desenvolveu-se a interface inicial do software, onde o usuário interage com a aplicação e a ele posteriormente são exibidos os resultados de suas ações.
- `PastaPDFParser`: nesta classe é feita a varredura dos arquivos do diretório selecionado, o texto de cada arquivo é enviado para a função `limpaTexto` onde é feito um processamento para deixá-lo menos poluído.
- `Sentencas`: esta classe é responsável por receber o texto do arquivo e segmentar ele em frases.
- `Presenca`: é responsável por verificar a presença das palavras buscadas no texto dos arquivos.
- `TelaDetalhes`: é responsável por apresentar a frase em que foi encontrado o resultado, além de mostrar a frase anterior e posterior, para contextualizar ao usuário aonde a frase se encontra no documento.
- `Sinonimos`: tem o papel de verificar a existência de sinônimos para cada palavra que o usuário deseja buscar.

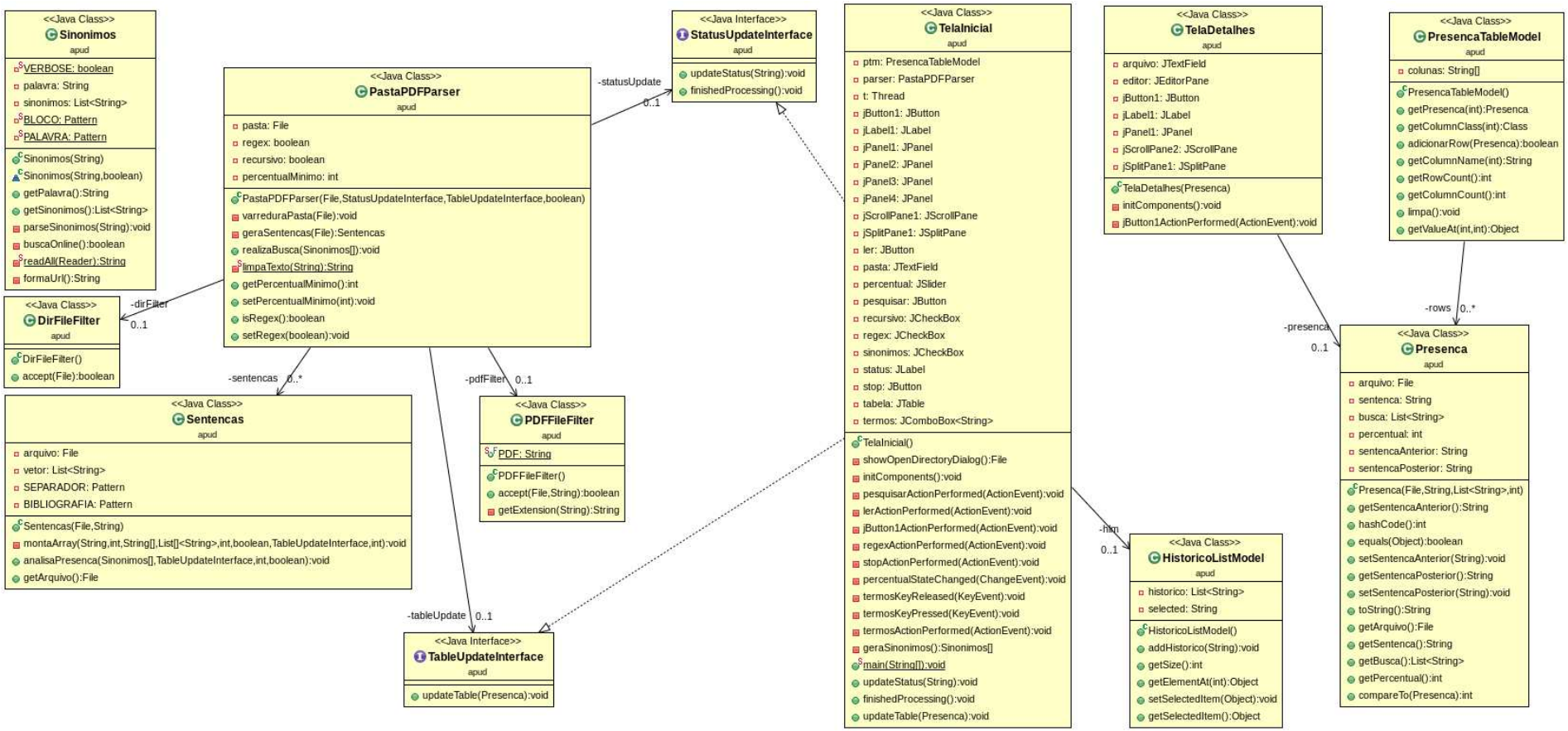

Figura 2. Diagrama de classes

3.2. Comparativo com softwares similares

Foram avaliados os softwares existentes citados na Seção 1 que atuam ou possuem recursos que os torna utilitários na busca em acervos de PDFs, de modo a caracterizar:

- Escopo: a unidade na qual a busca pode ser restringida, por exemplo, arquivos ou páginas;
- Capacidade de busca por termo simples: suporte do utilitário para a mais simples das buscas;
- Capacidade de busca por expressão simples: capacidade do utilitário suportar busca por uma expressão, ou seja, um conjunto de palavras separadas que devem ser tratadas como um único bloco;
- Capacidade de busca por múltiplos termos: disponibilidade para busca por múltiplos termos ou expressões dentro do escopo;
- Suporte a expressões regulares: possibilidade de uso de expressões regulares para a realização da busca;
- Suporte a recursão em subdiretórios: existência da possibilidade de busca em subdiretórios no caso de acervos de arquivos hierarquizados em pastas;
- Licença de uso: o tipo de licença de uso da ferramenta.

3.3. Implementação

Do início da modelagem, no decorrer do desenvolvimento do software, foi enfatizada a facilidade de uso por parte do usuário, para tal, foi desenvolvida uma interface simples e intuitiva, visando também, o pleno funcionamento e aumento na produtividade.

A fim de ilustrar as funcionalidades propostas e levantar requisitos funcionais, foi desenvolvido o diagrama de caso de uso ilustrado na Figura 3. Por meio do diagrama, demonstra-se como o software trabalha, e quais ações o usuário deve tomar para interagir com a ferramenta.

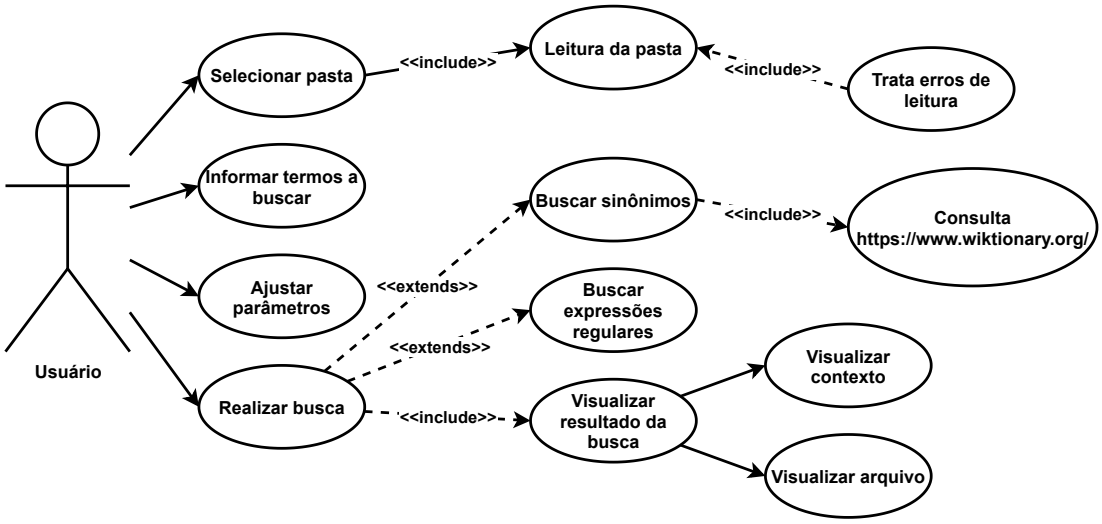

Figura 3. Interação no uso do software

No ciclo de vida completo da aplicação, após a análise de requisitos e concepção do projeto, a etapa seguinte foi a implementação do sistema. No estágio inicial destas etapas, o foco restringiu-se principalmente na lógica, função e tecnologia do sistema.

A fase de desenvolvimento do sistema foi empreender o trabalho concebido nas etapas anteriores transformando o projeto na implementação concreta correspondente. Portanto, a implementação do sistema foi o resultado final do projeto e da análise do sistema.

A linguagem utilizada para o desenvolvimento da aplicação foi Java, empregando-se Swing para as telas interativas, o que permite que o aplicativo seja compatível com múltiplos sistemas operacionais.

Pensando no amplo uso do sistema, foi implementado no projeto o Java resource bundles, um pacote de recursos configurável, que contém dados do idioma do software, oferecendo a possibilidade de internacionalizar aplicativos Java. Por padrão, o software adota a linguagem padrão do sistema em que está sendo executado, sendo que foram configuradas as expressões para o português do Brasil (`pt-br`) e inglês dos Estados Unidos (`en-us`).

Quando inicializado o software, o diálogo principal é apresentado ao usuário, onde ele deve apontar a pasta a ser realizada a leitura dos arquivos. Ao pressionar o botão de iniciar a leitura dos arquivos, um processo paralelo é iniciado, a fim de realizar a leitura sem interromper outras funcionalidades do software. Quando finalizado esse processo, a busca é habilitada. Nesse momento, cabe ao usuário dar entrada nos termos a serem buscados e determinar os parâmetros para a busca desejada. Após realizada a busca, a tabela é preenchida com os resultados, sendo estes passiveis de ordenação dos resultados por nome de arquivo, termo encontrado, frase, ou por relevância (porcentagem relativa a quantia de termos encontrados na sentença). Ao usuário, fica visível o *status* atualizado da busca em um campo de texto, para acompanhamento do processo e do andamento da busca.

Considerando que a ferramenta foi concebida especificamente para o fim de buscas, alguns recursos foram incorporados, tais como, busca por sinônimos e com expressões regulares. Para a implementação da busca por sinônimos, quando marcada a caixa de seleção, além de realizar a busca pelos termos determinados pelo usuário, é feita uma verificação da existência de sinônimo das palavras utilizando o Wiktionary [Wiktionary 2022], sendo este um projeto web concebido com a finalidade de criar um dicionário eletrônico de conteúdo livre.

Como um recurso adicional, foi implementado no campo de busca um histórico de pesquisas anteriores. Ao expandir o campo, é apresentado um histórico em ordem cronológica, das buscas realizadas anteriormente. A funcionalidade é válida para a sessão de uso atual, portanto, ao fechar o software e inicializá-lo novamente, as buscas realizadas na sessão anterior não terão persistência.

3.4. Estudo das bibliotecas de interação com PDF disponíveis

As principais bibliotecas atualmente disponíveis para leitura e manuseio dos arquivos PDFs foram elencadas, observando em que linguagens de programação elas são disponibilizadas, o seu funcionamento estrutural básico, a capacidade e restrições de trabalho e o licenciamento de uso. As seguintes bibliotecas para a interação com arquivos do tipo PDF foram analisadas:

- Content ExtRactor and MINEr (CERMINE) [Tkaczyk et al. 2015]: é uma biblioteca Java e um serviço web (cermine.ceon.pl) para extração de metadados e

conteúdo de arquivos PDF contendo publicações acadêmicas. O CERMINE é escrito em Java e desenvolvido no Centro de Ciência Aberta do Centro Interdisciplinar de Modelagem Matemática e Computacional da Universidade de Varsóvia. Embora a biblioteca conte com múltiplos recursos para interpretação de diferentes tipos de conteúdos, nos testes realizados, a performance deixa a desejar, talvez por contar com muitos recursos que não são aproveitados, uma vez que somente são interpretados e indexados blocos de texto.

- Apache PDFBox [PDFBox 2022]: a biblioteca Apache PDFBox é uma ferramenta Java de código aberto para trabalhar com documentos PDF. A biblioteca permite a criação de novos documentos PDF, a manipulação de documentos existentes e a capacidade de extrair conteúdo de documentos. Apesar de disponibilizar uma API completa, nos testes, a biblioteca demonstrou certa limitação na varredura de arquivos, gerando grande número de exceções e, eventualmente entrando em laços infinitos.
- Apache Tika [Tika 2022]: o conjunto de ferramentas do Apache Tika detecta e extrai metadados e textos de mais de mil tipos diferentes de arquivos (como PPT, XLS e PDF). Todos estes tipos de arquivo podem ser analisados através de uma única interface, tornando o Tika útil para indexação em mecanismos de busca, análise de conteúdo, tradução e muito mais.

Considerando testes de performance com as diferentes APIs, e a própria análise de contexto da concepção das bibliotecas, a Apache Tika foi selecionada para uso no software.

3.5. Teste de validação

Para realizar a validação da ferramenta proposta, foi desenvolvido um método próprio de teste a ser realizado com o objetivo de verificar e comparar a eficiência dos softwares testados em realizar buscas em um documento simples. Para tal, foi criado um documento no formato PDF contendo dois parágrafos, com duas frases cada, para a condução de testes com as ferramentas Windows Explorer (Windows 10), Finder (MacOS 10.15.7 Catalina) e Nautilus (Ubuntu 20.04), por estarem presentes nos sistemas operacionais atualmente mais populares [Moraes et al.], e também com as ferramentas Adobe Acrobat Pro DC, PDF Studio (Qoopa Software), FoxIt PhantomPDF, SmallPDF, UltraEdit UltraFinder e por fim, o software proposto, de modo a apresentar a acuracidade dos resultados apresentados ao usuário mediante diferentes formatos de busca.

Apesar de também terem sido realizado testes com grandes volumes de arquivos, para tornar o processo simples e funcional, o seu desenvolvimento consistiu na realização de nove buscas em um documento PDF simples (Figura 4), visando avaliar a capacidade de cada ferramenta em determinar o arquivo como positivo, conforme os testes enumerados a seguir:

1. Busca por termo simples: `document`
2. Busca por termo composto: `asymmetric_cryptography`
3. Busca por dois termos na mesma sentença: `signature approved`
4. Busca por dois termos em sentenças distintas: `signer approved`
5. Busca por expressão regular simples: `since\s\d{4}`
6. Busca por duas expressões regulares na mesma sentença: `since\s\d{4} alter(?:ed)?`

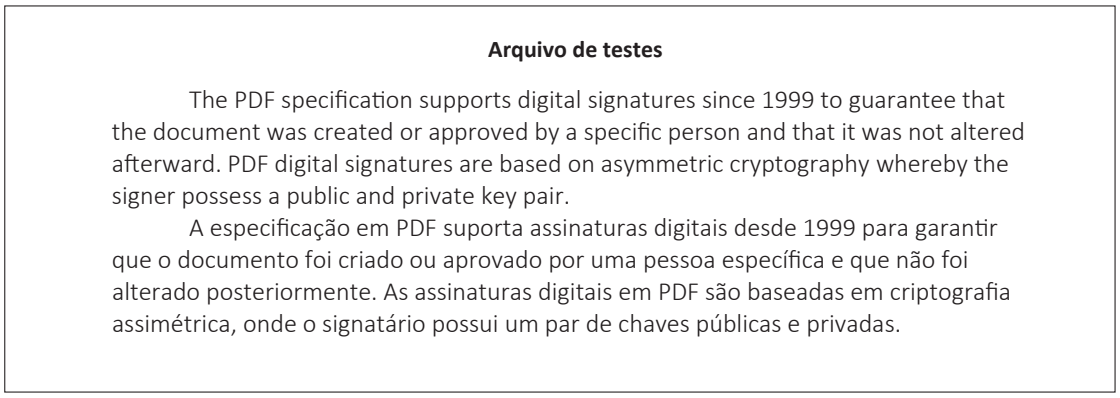

Figura 4. Arquivo utilizado para testes

7. Busca por duas expressões regulares em sentenças distintas: `since\s\d{4} asym(?:metric|metry)`
8. Busca com suporte a sinônimos: `signataire`
9. Busca com dois termos usando suporte a sinônimos: `signataire couple`

A partir destes testes, foi observada a quantidade de resultados positivos para cada um dos softwares. Apesar de terem sido realizados testes com o suporte a sinônimos, por ser um recurso único do software desenvolvido, estes foram desconsiderados para a elaboração gráfica, ficando, portanto, sete possíveis resultados positivos a serem considerados no teste das ferramentas.

4. Resultados

4.1. Interface e funcionamento

Utilizando o Java Swing para a construção da interface, foi obtida a tela inicial mostrada na Figura 5. Uma tela bastante simples e intuitiva, podendo ser facilmente entendida mesmo por usuários com pouca afinidade com informática, sendo que a interface foi testada com alguns usuários que relataram essa experiência de uso.

Para a utilização do software, o primeiro passo a ser seguido é apontar o diretório da pasta em que o acervo de PDFs se encontra, no campo ”Pasta de artigos”. Para tanto é preciso digitar o diretório, ou de forma mais simples, é possível utilizar o ícone que ilustra uma pequena pasta para navegar até o diretório dos arquivos. Após indicada a pasta em que os arquivos estão, opcionalmente pode ser selecionada a funcionalidade de ”Recursão nas pastas” para realizar a leitura de arquivos em subpastas. Posteriormente, basta clicar no botão ”Ler PDFs”, para que a ferramenta, fazendo uso da biblioteca Apache Tika, inicie a varredura dos dados presentes nos arquivos.

Após finalizada a leitura, é possível iniciar a busca textual nos arquivos. No lado direito da tela, é possível selecionar as opções de busca por sinônimos, busca por expressões regulares e, caso desejado, ajustar a opção ”Combinação mínima”, que mostra o nível de precisão da busca em relação aos termos inseridos (percentual de termos encontrados em relação ao número total de termos declarados pelo usuário). Após a determinação dos ajustes adequados à busca, deve ser acionado o botão ”Pesquisar” e os

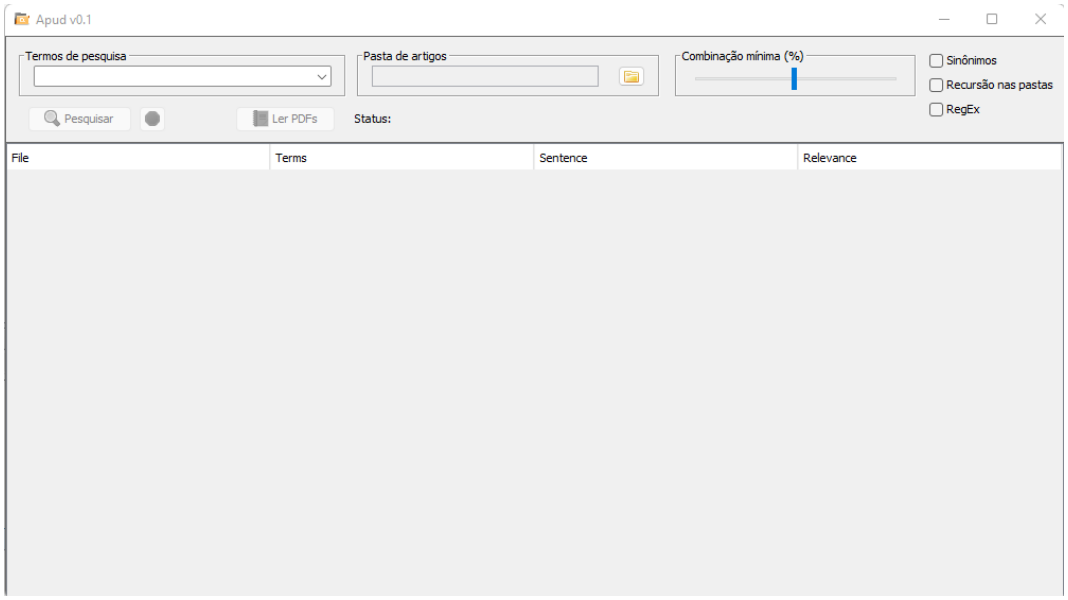

Figura 5. Interface da tela inicial do software

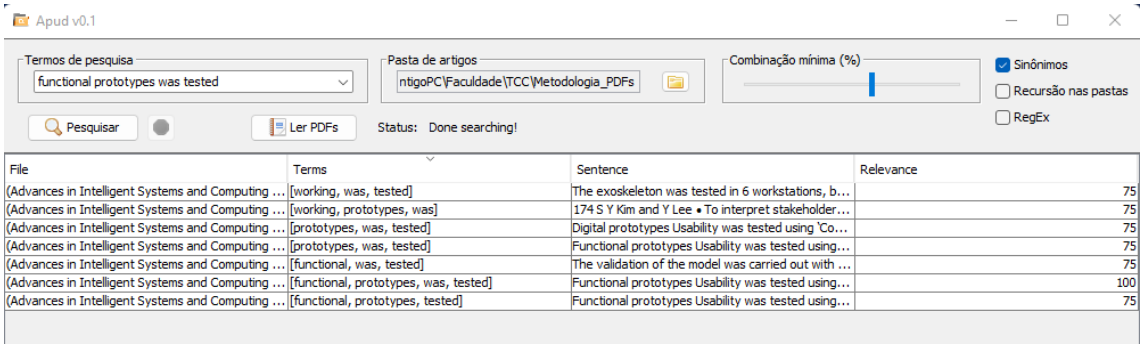

Figura 6. Resultado após busca de termos

resultados são apresentados na tabela que ocupa a parte inferior da tela, como mostrado na Figura 6.

A partir da visualização das buscas, é possível identificar em quais arquivos a busca teve êxito, quais os termos encontrados, a frase onde estão localizados e a relevância, que é baseada na quantidade de termos encontrados. Para facilitar a interpretação do contexto em que a frase se encontra, é possível acionar um gatilho com um clique duplo na linha da tabela, sendo assim exibido um diálogo de detalhes (Figura 7), no qual é possível visualizar as frases anterior e posterior da referenciada, também dando possibilidade de fazer uma chamada de abertura do arquivo no qual a frase foi encontrada, utilizando o visualizador de PDFs padrão do sistema.

No decorrer da realização das buscas, visualiza-se a geração do histórico de buscas apresentado no menu ”Termos de pesquisa”, conforme demonstrado na Figura 8.

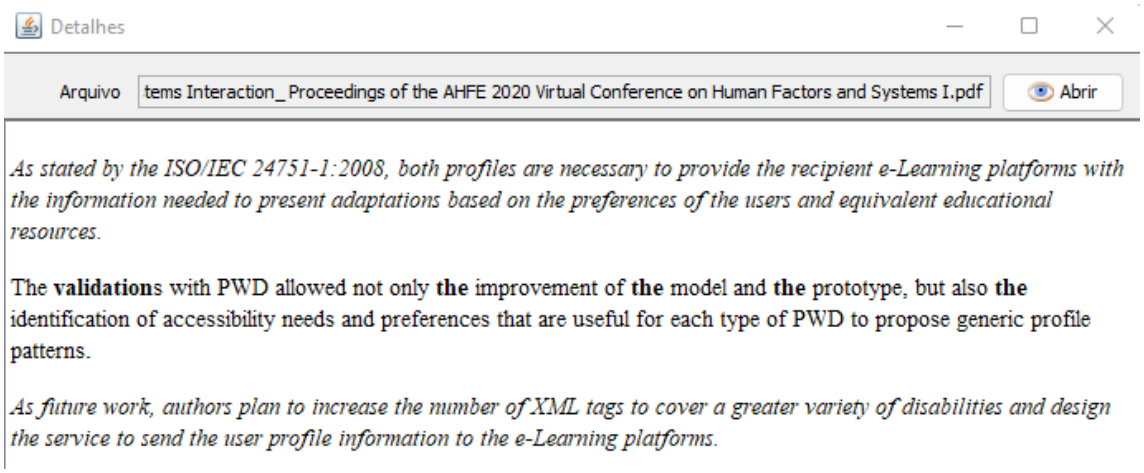

Figura 7. Tela com maiores detalhes

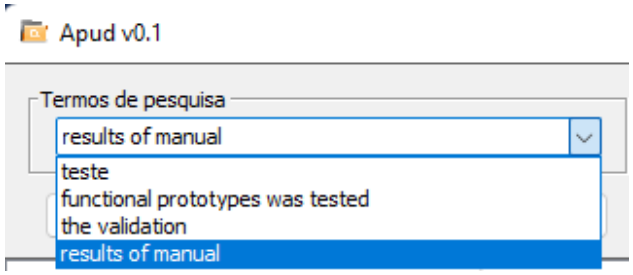

Figura 8. Histórico de buscas

4.2. Validação

O resultado do teste de validação, realizado conforme descrito na metodologia da Seção 3.2, está representado na Figura 9. É perceptível que a maior parte das ferramentas apresentou bons resultados nas buscas textuais, algumas suportando buscas por termos compostos, ou seja, palavras ou expressões distribuídas separadamente no corpo do texto, sendo na mesma frase ou não, como foi o caso do Windows Explorer, Finder do MacOS e FoxIt PhantomPDF.

O uso de expressões regulares é um recurso das ferramentas PDF Studio e UltraEdit UltraFinder, mas suportando apenas uma expressão regular por busca. Devido a grande flexibilidade das expressões regulares, este não chega a ser um fator que pode ser considerado limitante. Outro ponto que pode justificar a falta de suporte às expressões regulares, pode ser a necessidade de conhecimento técnico por parte do usuário para fazer bom uso deste recurso.

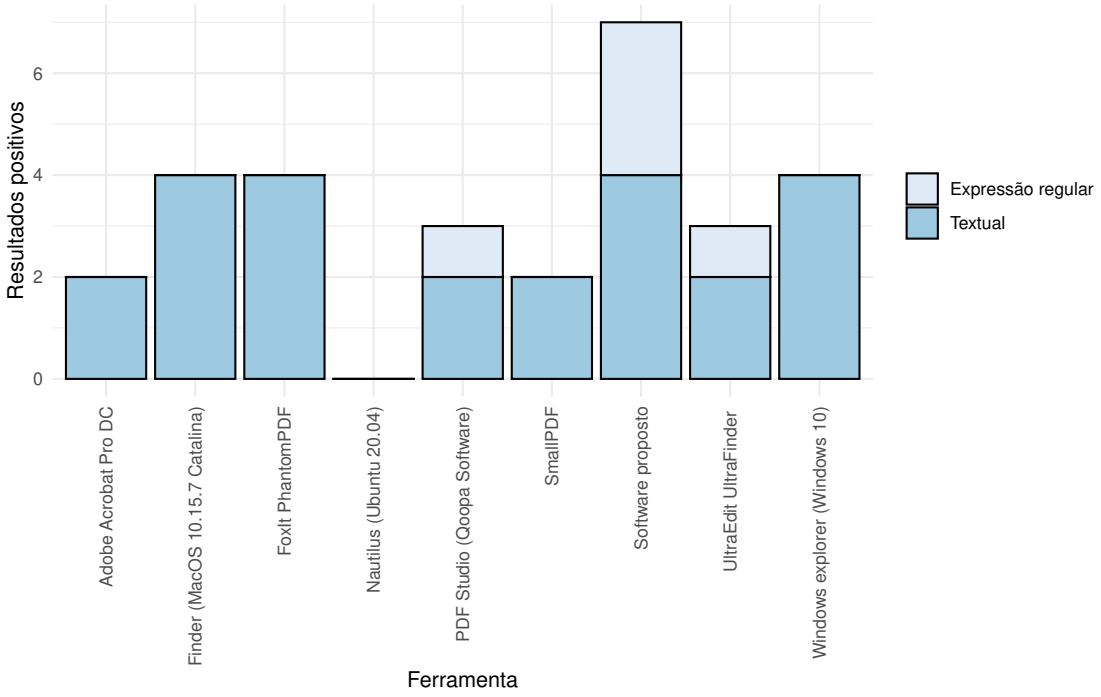

Figura 9. Resultado dos testes de buscas

4.3. Comparativo com softwares similares

Observando o paralelo de recursos dispostos na Tabela 1, é possível notar que o software proposto é o único que tem como escopo a frase para a busca nos documentos, o que se torna uma grande vantagem em relação aos demais softwares apresentados quando tratamos de busca por referências para afirmações específicas, prática usual na escrita acadêmica. Alguns dos softwares apresentados possuem como escopo o arquivo como um todo, o que possibilita buscas em acervos de arquivos, mas de forma limitada se comparado com o software proposto, e ainda, alguns softwares tem como escopo apenas nome de arquivos, ou arquivos abertos unitariamente, fato que impossibilita a realização de buscas avançadas. Também é perceptível que o software desenvolvido aporta as principais características e recursos dos demais softwares, incluindo o suporte a expressões regulares, além de trazer recursos exclusivos, como o suporte pela busca por sinônimos.

Outro fator de grande importância a ser destacado é o tipo de licença dos softwares comparativos elencados. Com exceção do Nautilus (gerenciador de arquivos do Ubuntu 20.04), que tem um escopo de busca bastante restrito (somente nome de arquivo), os demais são ferramentas de licença comercial ou embarcados em sistemas operacionais de licença comercial. Tal fato contrasta com o software desenvolvido, que possui licença gratuita, denominadamente *freeware*, a qual dispensa licenciamento, o que evidencia em um importante diferencial.

Tabela 1. Comparativo entre Softwares

Ferramenta	Escopo	Termo	Expressão simples	Múltiplos termos	RegExp[a]	Recursão em subdiretórios	Licença de uso
Windows explorer (Windows 10)	Arquivos	Sim	Sim, empregando aspas	Sim, operador padrão AND, suporta OR	Não	Sim, por padrão	Incorporado no sistema de licença Electronic Software Delivery
Finder (MacOS 10.15.7 Catalina)	Arquivos	Sim	Sim, empregando aspas	Sim, operador padrão OR	Não	Sim, por padrão	Incorporado no sistema de licença comercial
Nautilus (Ubuntu 20.04)	Apenas nome de arquivo	Sim[b]	Sim (somente nome de arquivo)	Sim (somente nome de arquivo)	Não	Sim	Software livre
Adobe Acrobat Pro DC	Arquivos	Sim	Sim, por padrão	Não	Não	Sim	Comercial (Pro) com versão gratuita (Reader)
PDF Studio (Qoopa Software)	Arquivos	Sim	Sim, por padrão	Não	Sim	Sim, como opção	Duas licenças comerciais (Pro e Standart)
FoxIt PhantomPDF	Arquivos	Sim	Sim, com aspas ou seleção de propriedade	Sim, operador padrão OR, com opção para AND	Sim[c]	Sim	Comercial
SmallPDF	Arquivo aberto	Sim	Sim, por padrão	Não	Não	Não	Comercial (formato de assinatura)
UltraEdit UltraFinder	Arquivos	Sim	Sim, por padrão	Não	Sim	Sim, como opção	Comercial
Evermap AutoDoc-Search plug-in	Arquivos / Páginas	Sim	Sim, por padrão	Sim (sistema de listas)	Sim	Não, mas com suporte a múltiplos diretórios	Comercial (Plug-in para o Acrobat)
Software proposto	Frase	Sim	Sim, utilizando underline para espaços explícitos	Sim, operador padrão OR, suporte a % de adequação	Sim	Sim, como opção	Software gratuito (Freeware)

[a]Expressão regular
[b]Somente nome de arquivo
[c]Suporta formatos pré-definidos de tipos de dados

5. Discussão

Obteve-se exito no desenvolvimento de uma ferramenta para buscas avançadas em acervos de PDFs de forma exitosa, que diferencia-se dos demais softwares com recurso similar por aportar um escopo restrito a frase, busca por sinônimos e capacidade de realizar busca por mais de uma expressão regular, recursos estes, que podem ser de grande valia ao usuário, especialmente em um cenário de busca de referências para a escrita acadêmica. Além disto, diferencia-se pelo fato de tratar-se de um *freeware*, ou seja, um software livre de licenciamento, destinado a qualquer público que tenha interesse em fazer sua utilização.

A partir da validação realizada na Seção 4.2, verificou-se que algumas ferramentas possuem limitações quando apenas o suporte a buscas avançadas é analisado, obtendo um resultado igual ou inferior a ferramentas oferecidas por sistemas operacionais.

As expressões regulares foram suportadas por alguns softwares, mas é entendível que não seja um recurso muito popular, visto que demanda do usuário um grau de conhecimento mais elevado.

Dentre as ferramentas avaliadas para as buscas, com base na validação realizada no trabalho, o software proposto foi o único capaz de realizar os testes propostos, o que é coerente uma vez que é uma ferramenta totalmente voltada para este fim. As demais ferramentas, por sua vez, embora tragam recursos para a realização de buscas, têm ênfase em outras funcionalidades, relacionadas com a manutenção de arquivos de PDF ou ainda, trazendo aos usuários outros recursos.

6. Trabalhos futuros

Como trabalhos futuros, analisa-se uma forma de disponibilização da ferramenta desenvolvida. Ainda, considera-se a possibilidade de implementar uma integração do software desenvolvido com outras ferramentas de uso acadêmico, como por exemplo o Mendeley [Reiswig 2010] e LaTeX, por meio da exportação de acervos em formato BibTeX [Fenn 2006]. Além disso, realizar a tradução do software para mais línguas pode ser de grande utilidade, a fim de facilitar seu uso por um público mais diverso. Por fim, alterações e ajustes podem ser realizados em função da demanda apresentada pelos usuários do software, buscando melhorar sua interface a usabilidade.

Referências

Acosta-Vargas, P., Gonzalez, M., Zambrano, M., Medina, A., Zweig, N., and Salvador-Ullauri, L. (2020). The portable document format: An analysis of pdf accessibility. *Advances in Intelligent Systems and Computing*, 1207 AISC:206–214.

Azkue, J. (2021). Embedding interactive, three-dimensional content in portable document format to deliver gross anatomy information and knowledge. *Clinical Anatomy*.

Bhatia, G., Tewari, A., Gurbani, G., Gokhale, S., Varyomalani, N., Kirtikar, R., Bhatia, Y., and Athavale, S. (2021). Extraction of tabular data from pdf to csv files. *Advances in Intelligent Systems and Computing*, 1174:183–193.

Castiglione, A., De Santis, A., and Soriente, C. (2010). Security and privacy issues in the portable document format. *Journal of Systems and Software*, 83(10):1813–1822.

Chung, B., Chung, M., and Park, J. (2020). Portable document format file containing the schematics and operable surface models of the head structures. *Journal of Korean Medical Science*, 35(27).

Fenn, J. (2006). Managing citations and your bibliography with bibtex. *The PracTEX Journal,(4)*.

Gil, R., Piveta, E., Saccol, D., and Faveri, C. (2015). Uma ferramenta para busca não estruturada em código aspectj. In *Anais do XI Simpósio Brasileiro de Sistemas de Informação*, pages 39–46. SBC.

Gopaldinne, S., Kaur, H., Kaur, P., Kaur, G., and Madhuri (2021). Overview of pdf malware classifiers. pages 337–341.

He, K., Zhu, Y., He, Y., Liu, L., Lu, B., and Lin, W. (2020). Detection of malicious pdf files using a two-stage machine learning algorithm. *Chinese Journal of Electronics*, 29(6):1165–1177.

Hossain, S. and Ayub, M. (2020). Parameter optimization of classification techniques for pdf based malware detection.

ISO, I. (2008). 32000-1: 2008, document management–portable document format–part 1: Pdf 1.7. *International Organization for Standardization, Geneva, Switzerland*.

Kostrinsky-Thomas, A., Hisama, F., and Payne, T. (2021). Searching the pdf haystack: Automated knowledge discovery in scanned ehr documents. *Applied Clinical Informatics*, 12(2):245–250.

Mikhailov, A., Shigarov, A., Rozhkov, E., and Cherepanov, I. (2020). On graph-based verification for pdf table detection. pages 91–95.

Mladenov, V., Mainka, C., Meyer zu Selhausen, K., Grothe, M., and Schwenk, J. (2019). 1 trillion dollar refund: How to spoof pdf signatures. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1–14.

Moraes, C., Dornelles, R., and da Rosa, E. Fotogrametria 3d-desempenho nos sistemas operacionais: windows, linux e mac os x. *URL: https://www. researchgate. net/publication/347513014_Fotogrametria_3D_-_Desempenho_nos_Sistemas_Operacionais_Windows_Linux_e_Mac_ OS_X, doi*, 10:m9.

PDFBox, A. (2022). Apache pdfbox. Disponível em: `https://pdfbox.apache.org/`. Acesso em: 04 Fevereiro 2022.

Reiswig, J. (2010). Mendeley. *Journal of the Medical Library Association: JMLA*, 98(2):193.

Shelton, Z. and Yu, C.-H. (2020). Pdf readability enhancement on mobile devices. In *Proceedings of the 17th International Web for All Conference*, pages 1–4.

Simcoe, T. (2006). Open standards and intellectual property rights. *Open innovation: Researching a new paradigm*, 161:183.

Singh, P., Tapaswi, S., and Gupta, S. (2020). Malware detection in pdf and office documents: A survey. *Information Security Journal*, 29(3):134–153.

Tika, A. (2022). Apache tika. Disponível em: https://tika.apache.org/. Acesso em: 04 Fevereiro 2022.

Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., and Bolikowski, Ł. (2015). Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(4):317–335.

Wiktionary (2022). Wiktionary. Disponível em: https://www.wiktionary.org/. Acesso em: 04 Fevereiro 2022.

Wives, L. K. and Sardi, F. L. (1996). Uma ferramenta para simplificar a busca de informações em documentos textuais. *Salão de Iniciação Científica (8.: 1996: Porto Alegre, RS). Livro de resumos. Porto Alegre: UFRGS/PROPESQ, 1996.*

Zulfiqar, S., Arooj, S., Hayat, U., Shahid, S., and Karim, A. (2020). Automated generation of accessible pdf.