Análise e Benchmarking de ferramentas de extração de documentos

Resumo

Este documento tem como objetivo documentar toda a pesquisa e avaliação efetuada para a fase de extração de documentos.

Mais especificamente, esta fase consistiu em **encontrar uma ferramenta de extração de PDF** que sirva os objetivos principais deste projeto.

Esta pesquisa é especialmente relevante dado que esta fase servirá como base para o resto da pipeline do projeto.

17/04/2026

Funds & Investments – Axians DC

axians

EXÉRCITO

VINCI ENERGIES

Índice

1 Introdução 4
1.1 Contexto e limitações 4
1.2 Natureza dos documentos 5
2 Estado de arte 6
2.1 Ferramentas consideradas 6
2.2 Análise detalhada 7
2.2.1 Metadados e Elementos 7
2.2.2 Preservação de estrutura e hierarquia 9
3 Procedimento 10
3.1 Documentos para Avaliação 10
3.2 Configurações das ferramentas usadas 10
3.3 Protocolo de Avaliação 10
4 Resultados e Discussão 11
4.1 Resultados para o Documento *ME 3-31-18* 11
4.1.1 Docling 11
4.1.2 Marker 12
4.1.3 MinerU 13
4.1.4 PyMuPDF4LLM 13
4.1.5 Unstructured 14
4.2 Resultados para o Documento *NAT 02.00* 15
4.2.1 Docling 15
4.2.2 Marker 16
4.2.3 MinerU 17
4.2.4 PyMuPDF4LLM 17
4.2.5 Unstructured 18
4.3 Avaliação dos resultados 18
4.3.1 Tabelas de avaliação 19
4.3.2 Discussão 20
5 Conclusão 22
5.1 Considerações Futuras 22

1 Introdução

O assistente conversacional irá responder a questões com base no conteúdo dos documentos. Assim, os documentos têm de ser extraídos com qualidade para o bom funcionamento do projeto, ou seja, esta primeira fase de extração é crucial e é a base para toda a pipeline.

O conteúdo, para além de ser extraído com qualidade e precisão, precisa também de ser estruturalmente rico e com uma hierarquia bem identificada. Isto engloba secções, cabeçalhos, hierarquia do documento, diversos elementos, páginas, rodapés, etc.

Estes dados dos documentos são especialmente importantes para a próxima fase do projeto: chunking. Quanto mais rico e estruturado for o output na fase de extração, mais opções permanecem abertas para o design de chunking, indexação e recuperação.

Uma extração que preserve a estrutura e ordem de leitura do documento, tipifica cada elemento, liga legendas às respetivas figuras e separa o corpo de cabeçalhos/rodapés, acaba por manter várias opções em aberto e dá à pipeline um maior controlo no que toca ao uso dos dados dos documentos.

Estas duas dimensões, **preservação de hierarquia** e **precisão da extração**, são os principais critérios de avaliação ao longo deste relatório.

1.1 Contexto e limitações

De forma a possibilitar a realização de testes nos computadores disponíveis, e também para englobarmos uma maior versatilidade, procuramos ferramentas que não sejam muito exigentes a nível computacional e com capacidade de operar apenas em CPU. Assim, ferramentas que operam exclusivamente em GPU são excluídas.

Dá-se então a exclusão de diversas ferramentas, permitindo uma pesquisa mais afunilada e seletiva. Isto porque, grande maioria das ferramentas de extração dependem de modelos neuronais para deteção de layout e OCR, acabando por serem substancialmente mais lentas em CPU. Mesmo aquelas que não são muito computacionalmente exigentes, chegaram a causar erros de memória devido à elevada dimensão dos documentos presentes no dataset do projeto, sendo necessário um processamento em separado por intervalos de páginas.

Adicionalmente, dada a natureza do projeto, todo o processamento tem de ocorrer localmente. Isto exclui diversas APIs cloud e serviços de extração consolidados no mercado.

O nosso espaço de pesquisa relativamente a ferramentas de extração passa então a ser a **pipelines de extração locais, open-source e capazes de funcionar em CPU e/ou GPU.**

1.2 Natureza dos documentos

O dataset do projeto é extenso e contempla diversos tipos de documentos com características distintas. Estes incluem: Despachos; Manuais; Normas/Instruções; Planos de Formação; etc.

Relativamente às propriedades e conteúdo, alguns dos documentos são nativos, outros digitalizados, existindo também documentos mistos. Muitos incluem tabelas, diagramas, cabeçalhos e rodapés (bastante comuns), e parágrafos (listas) com até seis níveis de indentação. Também é importante referir que a língua predominante é a língua portuguesa, com certas designações em inglês.

Através desta análise mais superficial, isto leva-nos a concluir que uma excelente qualidade de OCR com ênfase na língua portuguesa, e uma preservação da estrutura com múltiplos níveis de hierarquia são critérios imprescindíveis na avaliação das ferramentas.

2 Estado de arte

Dadas as limitações previamente referidas, a pesquisa passa por testar e procurar ferramentas de extração locais, open-source, capazes de funcionar em CPU e/ou GPU e com pipelines robustas e completas.

Dá-se então a exclusão a ferramentas exclusivamente cloud como LlamaParse, Mistral OCR, Azure Document Intelligence, Mathpix, entre outras. Outras ferramentas mais exigentes a nível computacional como Nougat, GOT-OCR 2.0, Zerox em modo local acabaram também por ser excluídas.

Existem também ferramentas mais específicas e modulares. Estas incluem GROBID para artigos académicos, Camelot para extração de tabelas, Surya como motor de OCR, LayoutParser que fornece deteção de regiões sem extração de texto e o pdfplumber que permite a extração das coordenadas do conteúdo com enfâse em páginas nativas. Estas são candidatas para uma futura pipeline customizada (Secção 5.1), mas não são pipelines completas sendo, portanto, excluídas do processo de avaliação deste relatório.

2.1 Ferramentas consideradas

Foram então escolhidas cinco ferramentas diferentes para avaliação que correspondem aos critérios previamente definidos, capazes de realizar uma extração com uma pipeline completa com outputs desejáveis:

- **Docling (IBM Research, MIT):** É a ferramenta arquiteturalmente mais completa. Oferece uma pipeline robusta e bem documentada, com um bom equilíbrio entre qualidade de extração e flexibilidade de configuração.
- **Marker (Datalab, GPL):** Reconhecida pela elevada qualidade de extração em documentos digitalizados. Produz um output estruturado e detalhado.
- **MinerU (OpenDataLab, Apache 2.0):** Orientado para documentos técnicos. Distingue-se pela sua qualidade de OCR.
- **PyMuPDF4LLM (Artifex, AGPL):** Extremamente rápida e leve. Ideal para documentos nativos, com a capacidade de extrair a informação da estrutura interna do PDF.
- **Unstructured (OSS) (Unstructured.io, Apache 2.0):** É também muito completa bem documentada com uma grande comunidade. É frequentemente citada em projetos.

2.2 Análise detalhada

A primeira análise foi então efetuada com base no enquadramento dos critérios mais essenciais do projeto, nomeadamente, tempo computacional, utilização de GPU e ferramentas disponíveis de OCR. Esta análise está representada na seguinte tabela:

Ferramenta	Tempo Computacional (CPU)	GPU	OCR
Docling	**Média**	**Opcional** Benéfica para aceleração de layout e OCR.	**EasyOCR/ Tesseract/ RapidOCR/ macOS Vision** Os motores mais eficientes sem GPU são Tesseract e/ou RapidOCR.
Marker	**Elevada**	**Recomendada** Surya em CPU é significativamente mais lento.	**Surya OCR/ ocrmypdf/ Tesseract** Surya é o default, e é transformer-based (mais exigente).
Unstructured	**Média**	**Opcional** GPU é usada no modo hi_res com Detectron2	**Tesseract/ PaddleOCR/ Google Cloud Vision** Tesseract é o default e usa Detectron2 para detetar o layout.
PyMuPDF4LLM	**Baixa**	**Não necessária**	**Tesseract/ RapidOCR** Heurística simples de OCR
MinerU	**Média**	**Opcional** Obrigatória para VLM	**PaddleOCR/ VLM backend** VLM usa o modelo MinerU2.5

Tabela 1. Análise de propriedades gerais das ferramentas.

2.2.1 Metadados e Elementos

Relativamente aos metadados presentes no output, as ferramentas diferem tanto na informação que fornecem sobre o documento como na variedade de elementos que conseguem distinguir individualmente.

A maioria das ferramentas é, em teoria, capaz de extrair os elementos necessários provenientes do conteúdo dos documentos presentes do dataset.

Categoria	Docling	Marker	Unstructured	MinerU	PyMuPDF4LLM
Títulos / Secções	✓	✓	✓	~	✓
Parágrafos / Texto	✓	✓	✓	~	✓
Listas	✓	✓	~	~	~
Tabelas	✓	✓	✓	✓	✓
Imagens / Figuras	✓	✓	✓	✓	✓
Fórmulas	✓	✓	✓	✓	✓
Blocos de Código	✓	✓	✓	✓	✓
Legendas	✓	✓	~	✗	✓
Notas de Rodapé	✓	✓	✗	✓	✓
Cabeçalhos / Rodapés	✓	✓	✓	✓	✓
Gráficos	✓	✗	✗	✓	✗

Tabela 2. Tipos de elementos gerados por cada ferramenta

Ao nível do documento, o **PyMuPDF4LLM** é o mais completo: lê diretamente os metadados internos do PDF e disponibiliza campos como autor, título e datas de criação e modificação. O **Docling** também fornece informação útil relativamente ao ficheiro em si e ao número de páginas, enquanto que o **Unstructured**, o **MinerU** e o **Marker** são mais limitados neste aspeto.

No que respeita à variedade de elementos distintos, o **Docling** apresenta o vocabulário mais robusto, incluindo tipos específicos como caixas de seleção, pares chave-valor ou índices do documento. O **MinerU** destaca-se por identificar elementos particularmente relevantes num contexto administrativo, como selos, carimbos e notas de margem, classificando ainda gráficos que se distinguem de imagens. O **Marker** é a única ferramenta a detetar elementos de escrita manuscrita, enquanto o **Unstructured** identifica automaticamente elementos como moradas e endereços de email. O **PyMuPDF4LLM** é o mais limitado, este não possui, por exemplo, legendas ou notas de rodapé como tipologias.

2.2.2 *Preservação de estrutura e hierarquia*

Relativamente à estrutura e preservação de hierarquia, as ferramentas apresentam outputs distintos, sendo que algumas apresentam mais limitações que outras.

O **Docling** possui a própria estrutura do documento como output que inclui referências aos elementos detetados. Adicionalmente, cada elemento possui referências a outros elementos associados de forma hierárquica.

Já o **Marker** pode ser considerado uma das ferramentas mais ricas neste aspeto. À semelhança do Docling, esta possui a própria estrutura do documento como output que inclui referências aos elementos detetados. O mesmo também possui um dicionário com o contexto hierárquico para cada elemento de forma individual, que é uma característica única desta ferramenta. Adicionalmente o mesmo cria o seu próprio índice através da estrutura do documento.

O **Unstructured** expressa a sua hierarquia para cada elemento também de forma muito rica. Cada elemento apresenta referências a outros elementos associados de forma hierárquica. Esta ferramenta mede também o nível de profundidade hierárquica para certos elementos de forma relativa.

Já o MinerU e o PyMuPDF4LLM são mais limitados neste aspeto. Nenhum possui um output relativo diretamente à hierarquia. A estrutura do documento teria de ser feita de forma ainda mais manual. O **PyMuPDF4LLm** possui o seu próprio índice, sendo possível extrair a estrutura a partir daí juntamente com headings provenientes do ficheiro markdown (classificados através do tamanho da fonte).

No caso do **MinerU**, o mesmo possui um nível atribuído para cada título, sendo a única propriedade de auxílio para a construção da estrutura do documento.

3 Procedimento

3.1 Documentos para Avaliação

Foram selecionados dois documentos do dataset do projeto para avaliação dadas as suas naturezas complementares, sendo um nativo e outro digitalizado:

- **ME 3-31-18 Manual Escolar: Emprego da Secção Mini UAV.** PDF nativo. Foram avaliadas as páginas de 1 a 15. O documento inclui: 2 páginas de capa (fundo verde e branco), uma tabela administrativa, um índice de 3 páginas, parágrafos densos com listas, 9 figuras (diagramas hierárquicos), uma tabela de classificação UAV, rodapés e cabeçalhos.

- **NAT 02.00 - Serviços de Diretório e Serviços Básicos de Rede da RDE.** PDF digitalizado. Foram avaliadas todas as 14 páginas de conteúdo. O documento inclui: uma tabela de cabeçalho administrativo, uma hierarquia de listas com até 6 níveis de profundidade, caixas de exemplo com bordas com conteúdo colorido, uma tabela de cabeçalho de 3 células em todas as páginas e um bloco de assinatura.

3.2 Configurações das ferramentas usadas

Cada ferramenta foi configurada para máxima fidelidade de extração dentro dos constrangimentos a nível de hardware:

- **Docling:** Modelo de layout por defeito com backend OCR Tesseract para língua portuguesa.
- **Marker:** Modo de output JSON (para acesso à estrutura do documento) com língua portuguesa.
- **MinerU:** Modo pipeline por defeito; operação CPU (modo VLM excluído). Opção de língua latim.
- **PyMuPDF4LLM:** Modelo de layout ativo com língua portuguesa.
- **Unstructured:** Estratégia de alta resolução com Tesseract OCR com língua portuguesa.

3.3 Protocolo de Avaliação

A avaliação foi realizada de duas formas:

1. **Análise de output bruto:** O output de markdown/texto foi comparado manualmente e forma empírica com as páginas do PDF de origem, isto engloba: conteúdo a nível de texto, ordem de leitura, estrutura das listas, precisão de tabelas, tratamento de figuras e legendas, deteção de notas em rodapé e a separação do cabeçalho/rodapé.

2. **Análise da estrutura do documento:** Através dos dados disponíveis de cada ferramenta, foi extraída manualmente a estrutura dos documentos. Esta foi inspecionada para contar e classificar os vários elementos presentes, dando enfâse à sua referência hierárquica.

4 Resultados e Discussão

4.1 Resultados para o Documento *ME 3-31-18*

Por ser um documento nativo, o desafio principal neste caso, não é propriamente o OCR, mas sim a fidelidade estrutural: a capacidade de cada ferramenta preservar hierarquia, identificar tipos de elementos e separar conteúdo principal de notas e cabeçalhos/rodapés.

Para este documento, os resultados para cada ferramenta a nível de elementos estão presentes na seguinte tabela:

Métrica	Docling	PyMuPDF4LLM	Unstructured	MinerU	Marker
Total de elementos	570	139	204	143	450
Tipos de elementos distintos	10	10	8	7	16
Texto/corpo	453	33	89	81	36
Cabeçalhos	18	21	16	19	15
Listas	30	18	19	-	23
Imagens/figuras	10	3	40	10	16
Tabelas	2	3	6	2	293 (incl. células)
Legendas	9	9	8	-	9
Títulos/subtítulos	20	20	26	-	26
Rodapés/Notas de Rodapé	28	32	-	31	29

Tabela 3. Resultados de cada ferramenta para o documento ME 3-31-18 Manual Escolar: Emprego da Secção Mini UAV

4.1.1 Docling

O Docling produziu cerca de 570 elementos, o maior volume bruto nesta avaliação, o que reflete uma segmentação detalhada do conteúdo, mas também uma tendência para fragmentar em excesso. Este possui um vocabulário tipológico rico e uma separação entre o corpo e cabeçalhos/rodapés que está arquiteturalmente presente na estrutura de output. As legendas das figuras estão também presentes e foram corretamente associadas.

No entanto, esta apresentou problemas significativos que afetam a escolha da ferramenta:

- Existem perturbações ocasionais na ordem de leitura, nomeadamente entre títulos de secções e texto, o que é relevante para o chunking.
- Uma parte do índice foi extraído de forma fragmentada, com os números de artigo separados dos seus conteúdos.

- Houve uma tabela que foi extraída com alguma degradação relevante.
- Existem certos elementos do tipo lista que ficaram sem página associada, apenas relevante caso o chunking use o número de página.

```
[parágrafo sobre a BAO...]          ← linha 227
[parágrafo sobre os 4 módulos ISR]  ← linha 233
Figura 1-1 ...                      ← linha 235
<!-- image -->

## CAPÍTULO 1 INTRODUÇÃO            ← linha 241  ⚠ título após o conteúdo
## SECÇÃO I SECÇÃO MINI-UAV         ← linha 243  ⚠

Introdução                          ← linha 245
[parágrafo sobre qualidade das notícias...]
```

Exemplo 1. Perturbações na ordem de leitura, Docling.

4.1.2 Marker

O Marker foi a ferramenta com melhor desempenho geral no documento nativo. Produziu 450 elementos com 16 tipos únicos, o vocabulário mais rico desta avaliação. A ordem de leitura está correta, assim como a extração de tabelas. O Marker foi também a única ferramenta a:

- Produzir o seu próprio índice (TOC) como um elemento distinto no seu output.
- Agrupar automaticamente as figuras com as respetivas legendas.
- Suprimir automaticamente e corretamente os cabeçalhos e rodapés identificados do conteúdo principal do documento (de forma geral).

Existem também alguns problemas identificados que embora não sejam sistemáticos, continuam a ser relevantes:

- Alguns conteúdos de células de tabelas estão corrompidos.
- Algumas notas de rodapé apresentam algum ruído, pois foram emitidas em HTML.
- Existem alguns cabeçalhos e rodapés que não foram corretamente suprimidos, estes infiltraram-se no conteúdo do documento a meio de blocos de texto.

```
PDF original:  ...integradas no BISTAR¹ vocacionada para garantir...

Marker:        ...integradas no BISTAR&lt;sup&gt;1&lt;/sup&gt; vocacionada...
               ← ⚠ ruído no rodapé

No cabeçalho:  ## 102. Classificação dos UAV&lt;sup&gt;2&lt;/sup&gt;
               ← ⚠ ruído no cabeçalho
```

Exemplo 2. Presença de ruído em cabeçalhos e rodapés, Marker.

4.1.3 MinerU

O MinerU produziu o menor volume de elementos (143) com o vocabulário tipológico mais restrito da avaliação: não existe uma distinção entre tipos de elementos relevantes como legendas, títulos, subtítulos, listas e cabeçalhos, todos estes foram colapsados em texto com títulos genéricos. O output markdown é globalmente legível e a ordem de leitura está correta. A tabelas também foram extraídas com sucesso e qualidade. Existe também uma associação implícita, mas não explícita, entre figuras e legendas.

Dada a pobre identificação de elementos, o MinerU apresentou os seguintes problemas:

- Alguns cabeçalhos foram identificados como títulos genéricos, fornecendo, portanto, uma hierarquia fundamentalmente errada.
- Algumas listas foram classificadas como parágrafos de texto e outros como títulos genéricos, perdendo assim a hierarquia e a parte semântica do documento.

```
── PAGE 12 ──
└─ [header] p.12  "NÃO CLASSIFICADO"                          ← ⚠  rodapé
└─ [header] p.12  "Introdução"                                ← ⚠  cabeçalho
└─ [header] p.12  "104. Orgânica das Secções MINI UAV (BrigRR)"



── PAGE 14 ──
└─ [header] p.14  "105. Orgânica das Secções MINI UAV (ZMA)5"
```

Exemplo 3. Cabeçalhos/rodapés classificados como headers, MinerU.

4.1.4 PyMuPDF4LLM

O PyMuPDF4LLM acabou por ser a ferramentas mais rápida. Foi a única, juntamente com o Unstructured, que tentou extrair o conteúdo dos diagramas presentes no documento, embora de forma incompleta e com pobre qualidade. Este produziu 139 elementos com 10 tipos únicos, o segundo menor volume da avaliação, mas com um perfil de problemas distinto e mais sistemático:

- índice foi extraído de forma anómala, este acabou por ficar fragmentados em tipos de elementos diferentes.
- Os cabeçalhos e rodapés não foram suprimidos. Estes acabam por interferir no fluxo da estrutura e conteúdo do documento de forma sistemática, degradando assim a parte semântica do documento.
- Alguns rodapés apresentam o conteúdo duplicado, pois está a ser extraído como dois elementos distintos.

```
── PAGE 5 ──
└─ [section-header]  "ÍNDICE"
└─ [table] p.5       ← ⚠ Índice p.5 como table

── PAGE 6 ──
└─ [text]   "Índice"
└─ [table] p.6          ← ⚠ Índice p.6 como table

── PAGE 7 ──
└─ [text]   "Índice 511. Segurança de itinerários..."
```

Exemplo 4. Índice fragmentado em três tipos de elementos diferentes, PyMuPDF4LLM.

4.1.5 Unstructured

O Unstructured produziu 204 elementos com 8 tipos únicos, o segundo menor volume. Como este não possui um output em markdown, os resultados focaram-se nas propriedades dos elementos em si, e na sua ordem de leitura. Foram observados inúmeros erros e problemas estruturais, nomeadamente:

- A hierarquia apresenta diversos problemas sistemáticos com associações erradas entre cabeçalhos, títulos, listas e texto, perdendo o seu contexto semântico. Parte deste problema provém do facto de todos os rodapés serem classificados como título.
- Certas tabelas foram duplicadas ao serem extraídas pelo facto de terem sido detetadas como tipos de elementos diferentes.
- Embora extraídos, os onogramas estão fragmentados de forma irregular com alguma degradação em OCR.
- O Índice, embora siga a ordem de leitura, ficou fragmentado em tipos de elementos diferentes.

```
── PAGE 3 ──
└─ [Header]  "NÃO CLASSIFICADO"
  └─ [Title] h1  "DESPACHO"
    └─ [ListItem]          "1. Aprovo, para utilização nos cursos..."
    └─ [ListItem]          "2. O ME 3-31-18 é uma publicação..."
    └─ [ListItem]          "3. Podem ser feitos extratos..."
    └─ [ListItem]          "4. O ME 3-31-18 entra em vigor..."
    └─ [UncategorizedText] "Mafra, ____ de _____________ de 2013"
    └─ [NarrativeText]     "O COMANDANTE"
    └─ [NarrativeText]     "JOÃO PEDRO RATO BOGA DE OLIVEIRA RIBEIRO..."
    └─ [UncategorizedText] "III"   ← ⚠ número de página dentro do Despacho
  └─ [Title] h1  "NÃO CLASSIFICADO"  ← ⚠ rodapé como título
```

Exemplo 5. Hierarquia definida indevidamente, com cabeçalhos como títulos e conteúdo não suprimido, Unstructured.

4.2 Resultados para o Documento *NAT 02.00*

Por ser um documento digitalizado, este possui propriedades complementares ao documento ME 3-31-18 permitindo avaliar com maior enfâse outros aspetos das ferramentas, nomeadamente: a qualidade de OCR, extração de tabelas com bordas finas, e a extração de conteúdo com texto colorido. Esta avaliação revelou diferenças de desempenho significativas entre ferramentas para este tipo de documento, o que também é possível verificar a nível de elementos na seguinte tabela:

Métrica	Docling	PyMuPDF4LLM	Unstructured	MinerU	Marker
Total de elementos	293	268	246	211	320
Tipos de elementos distintos	12	7	8	7	11
Texto/corpo	90	81	75	174	83
Cabeçalhos	19	50	34	12	13
Listas	105	78	106	-	65
Imagens/figuras	4	4	4	1	2
Tabelas	10	-	1	15	96 (incl. células)
Legendas	1	-	-	-	-
Títulos/subtítulos	37	40	26	-	46
Rodapés/Notas de Rodapé	16	15	-	8	15

Tabela 4. Resultados de cada ferramenta para o documento NAT 02.00 - Serviços de Diretório e Serviços Básicos de Rede da RDE

4.2.1 Docling

O Docling produziu cerca de 293 elementos o que, mais uma vez, revela um grande vocabulário tipológico para este tipo de documento, incluindo certos tipos de elementos distintos relativamente à avaliação do documento anterior. Estes permitiram a extração de caixas com bordas finas. A ordem de leitura também está globalmente correta.

Relativamente aos pontos negativos, o mesmo apresentou os seguintes problemas:

- Vários elementos do tipo lista ficaram sem página associada, sendo este um erro sistemático.
- Certas expressões de texto coloridas não foram extraídas em conteúdo
- As caixas que possuem texto normal e colorido foram classificadas de forma inconsistente no que toca ao seu tipo de elemento.
- Alguns elementos do tipo lista sofreram alguma degradação a nível de OCR onde certos caracteres passam a ser ilegíveis
- As tabelas de rodapé foram extraídas como tabelas normais que interrompem o conteúdo e fluxo do documento.

```
── PAGE 1 ──
└─ [section_header] h1  "3. SITUAÇÃO"

── PAGE None ──          ← ⚠ sem página
└─ [list]
   └─ [list_item] p.1  "a. A DCSI tem como missão, estudar, planear..."
   └─ [list_item] p.1  "b. O serviço de diretório é um repositório..."
   └─ [list_item] p.1  "c. A inexistência de um normativo..."

── PAGE 1 ──
└─ [section_header] h1  "4. EXECUÇÃO"

── PAGE None ──          ← ⚠ sem página
└─ [list]
   └─ [list_item] p.1  "Active Directory"
```

Exemplo 6. Elementos sem página associada, Docling.

4.2.2 Marker

O Marker produziu 320 elementos com 11 tipos diferentes, o maior volume da avaliação para este documento. A ordem de leitura está globalmente correta com uma boa hierarquia. As listas principais estão funcionais de forma geral, e todos os cabeçalhos/rodapés foram classificados corretamente mesmo de acordo com a sua tipologia.

Os problemas do Marker englobam:

- Os níveis de títulos/headers são atribuídos de forma inconsistente, embora tenham sido identificados com a hierarquia correta.
- As tabelas de rodapé foram extraídas com conteúdo corrompido por OCR
- Certas tags de HTML estão indevidamente presentes em rodapés e sublinhados
- Algumas sublistas que deviam estar separadas foram colapsados numa só lista.

```
PDF original:  DCSI | NAT 02.00 | Pág. 10 de 13 Pág.
Marker p.10:   nesi | NAT 02 00 | ray, it ue is ray.

PDF original:  DCSI | NAT 02.00 | Pág. 12 de 13 Pág.
Marker p.12:   ווקפ | NAT 02.00 | ray, iz ue io ray,
               ↑ caracteres hebraicos em vez de "DCSI"

PDF original:  DCSI | NAT 02.00 | Pág. 13 de 13 Pág.
Marker p.13:   DOOL | N. T. C. C. C. C. | (vazio)
               DCSI | | NA C 12:00      | l Dág 12 do 12 Dág
               DC01 | NAT 02.00         | Pág. 13 de 13 Pág.
```

Exemplo 7. Degradação da qualidade OCR nas tabelas de rodapé, Marker.

4.2.3 MinerU

O MinerU produziu cerca de 211 elementos com 7 tipos distintos, o vocabulário tipológico mais pobre desta avaliação. No entanto esta ferramenta apresentou uma excelente qualidade a nível de OCR, nomeadamente na extração de tabelas. Esta também extraiu a ordem de leitura correta e foi capaz de classificar as notas de rodapé.

Os pontos negativos englobam:

- A hierarquia deste documento que percorre até 6 níveis não foi de todo preservada, sendo que esta foi colapsada em texto.
- As caixas de exemplo com texto colorido foram classificadas de forma inconsistente
- Algumas dificuldades a nível de OCR para certas imagens e listas.
- Algumas tabelas de rodapés não foram detetadas como tabelas

```
└─ [text] h2 p.1  "1. FINALIDADE"                ← nível 1 → h2
└─ [text] h2 p.2  "(1) Estrutura lógica"          ← nível 2 → h2
└─ [text] h2 p.2  "(a) Objetos"                   ← nível 3 → h2
└─ [text] h2 p.2  "1. Computadores"               ← nível 4 → h2
└─ [text] h2 p.2  "b. Convenção de nomenclatura" ← nível 5 → h2
└─ [text] h2 p.2  "ii.Identificar a U/E/O..."     ← nível 6 → h2
```

Exemplo 8. Má preservação da hierarquia, com os níveis colapsados em h2, MinerU.

4.2.4 PyMuPDF4LLM

PyMuPDF4LLM produziu 268 elementos com 7 tipos diferentes, este é o segundo maior volume com 50 elementos de cabeçalhos detetados, o valor mais alto de todas as ferramentas e mais do dobro do esperado. Esta inflação provém do facto de detetar indevidamente as próprias bordas das tabelas e caixas com elementos de cabeçalho distintos.

Existem também outros problemas nomeadamente:

- Os cabeçalhos não foram suprimidos do conteúdo do texto
- Certas tabelas foram extraídas como figuras com uma baixa qualidade de OCR
- Pobre qualidade de OCR para caixas de exemplo e listas

```
PDF original:      DCSI | NEP/NAT 02.00 | Exemplar n.º / Versão 02 / 17AGO16


PyMuPDF4LLM:
  ----- Start of picture text -----
  DCSI p
  dam N = Pp Pag 1 de 13 Pag
  ENE PN NAT 02.00 Versão 02
  a Es [rs Ê
  ease 17AG016
  ----- End of picture text -----
  ← campos "DCSI", "NEP", "NAT" completamente ilegíveis
```

Exemplo 9. Tabela tratada como figura com uma baixa qualidade de OCR, PyMuPDF4LLM.

4.2.5 *Unstructured*

Unstructured produziu 246 elementos com 8 tipos únicos, o segundo menor volume. Esta replicou todos os problemas estruturais já presentes no documento anterior, sendo que alguns pioraram especificamente para este documento:

- As associações hierárquicas estão na sua grande maioria erradas. Mais uma vez, parte deste problema provém do facto de todos os rodapés serem classificados como título.
- Os cabeçalhos/rodapés estão sistematicamente presentes no conteúdo do documento
- Caixas de exemplo distintas foram colapsadas num só elemento com o conteúdo invertido e com baixa qualidade de OCR
- Baixa qualidade de OCR para marcadores de listas

```
── PAGE 1 ──
└─ [Header] p.1  "RESERVADO" ← ⚠ cabeçalho como título
  └─ [Table]  "Exemplar n.º NEP NAT 02.00..."
  └─ [Title] h1  "1. FINALIDADE"
    └─ [NarrativeText]  "Normalizar os serviços básicos..."
  └─ [Title] h1  "2. ÂMBITO"
    └─ [UncategorizedText]  "A"
    └─ [NarrativeText]  "presente norma aplica-se..."
  └─ [Title] h1  "3. SITUAÇÃO"
    └─ [ListItem]  "a. A DCSI tem como missão..."
    └─ [ListItem]  "b. O serviço de diretório..."
    └─ [ListItem]  "c. A inexistência de um normativo..."
  └─ [Title] h1  "4. EXECUÇÃO"
    └─ [ListItem]  "a. Active Directory"
  └─ [Title] h1  "RESERVADO"  ← ⚠ rodapé como título
```

Exemplo 10. Associações hierárquicas erradas, ao incluir cabeçalhos/rodapés, Unstructured.

4.3 Avaliação dos resultados

Como referido, existem duas dimensões que devemos considerar ao selecionar uma ferramenta de extração. A precisão da extração em si, assim como a riqueza dos dados do output nomeadamente a hierarquia e estrutura para as próximas fases.

Dados os resultados da extração para os dois documentos, definimos uma série de critérios para cada documento de forma a podermos avaliar a extração para cada ferramenta.

4.3.1 *Tabelas de avaliação*

A avaliação foi realizada de forma empírica e está enquadrada nas seguintes tabelas, com os respetivos critérios:

- ✓ **Bom,** indica que o critério foi satisfeito sem problemas relevantes;
- **~ Aceitável,** indica problemas presentes, mas limitados em termos de impacto ou frequência;
- ✗ **Deficiente,** indica uma falha sistemática ou impacto direto no uso do output para as próximas fases.

Critério	Docling	Marker	MinerU	PyMuPDF4LLM	Unstructured
Separação cabeçalho/rodapé	✓	✓	~	~	✗
Ordem de leitura	~	✓	✓	~	~
Hierarquia de secções	~	✓	✗	~	✗
Deteção de listas	~	✓	✗	~	~
Extração de tabelas	~	✓	✓	✗	~
Associação legenda/figura	✓	✓	✓	✗	✗
Notas de rodapé	✓	~	~	~	✗
Vocabulário tipológico	✓	✓	~	~	~
Coerência tipológica	~	✓	✗	~	✗
Extração de diagramas	✗	✗	✗	~	~
Índice (TOC)	✗	✓	~	✗	✗
Metadados ao nível da página	✓	✓	~	✓	~

Tabela 5. Avaliação de cada ferramenta para o documento ME 3-31-18 Manual Escolar: Emprego da Secção Mini UAV

Critério	Docling	Marker	MinerU	PyMuPDF4LLM	Unstructured
Qualidade de OCR em conteúdo	~	✓	✓	~	~
Qualidade de OCR em cabeçalho/rodapés	~	✗	✓	~	✗
Separação cabeçalho/rodapé	~	✓	✓	✗	✗
Ordem de leitura	~	✓	✓	✓	~
Hierarquia de secções	~	~	✗	~	✗
Deteção de listas	~	✓	✗	~	~
Caixas de exemplo coloridas	~	~	~	✗	✗
Extração de tabelas	~	~	✓	~	~
Notas de rodapé	✓	~	✓	~	✗
Consistência de classificação	~	~	✗	~	✗
Separação cabeçalho/rodapé	~	~	~	✗	✗

Tabela 6. Avaliação de cada ferramenta para o documento NAT 02.00 - Serviços de Diretório e Serviços Básicos de Rede da RDE

4.3.2 Discussão

As ferramentas Unstructured e PyMuPDF4LLM tiveram a pior prestação de forma geral. Os resultados do **Unstructured,** em específico, foram indesejáveis, a avaliação do mesmo ficou muito abaixo daquilo que era esperado e do que foi referido no estado de arte, embora tenha tido um tempo de extração médio relativamente rápido de 1 minuto. Os problemas observados teriam um grande impacto nas próximas fases, nomeadamente as associações indevidas da hierarquia e estrutura do documento.

Embora o **PyMuPDF4LLM** seja a ferramenta mais rápida com uma extração média de 30 segundos, esta demonstrou um desempenho medíocre em todos os aspetos, principalmente na qualidade de OCR. Não foram detetados problemas muito impactantes e sistemáticos, mas existem outras alternativas melhores.

Já o **MinerU** teve o efeito oposto, este subiu consideravelmente na avaliação relativamente ao seu posicionamento referido no estado de arte com um tempo de extração médio de 2 minutos. Foi das ferramentas com melhor desempenho e qualidade a nível de extração de conteúdo e OCR. No entanto, possui o mesmo tipo de problema que o Unstructured no que toca à preservação da hierarquia do documento.

Por fim, tanto o Docling como o Marker apresentam uma natureza mais robusta que vai de encontro com os objetivos do projeto. Embora com alguns erros na ordem de leitura, para o primeiro documento nativo, o **Docling** funcionou como esperado, uma extração robusta de diversos elementos com uma boa preservação de hierarquia e estrutura, e com um tempo de extração médio de 1 minuto. No entanto, para o documento digitalizado, a qualidade de OCR não foi a mais rica, sendo inferior a outras ferramentas

como o Marker e MinerU. Adicionalmente, ocorreram algumas perturbações na ordem de leitura do documento.

Nesta última avaliação, o **Marker** produziu resultados mais positivos relativamente ao Docling. O Marker é consistentemente a ferramenta com melhor desempenho estrutural em ambos os documentos. Este apresentou o vocabulário tipológico mais rico com uma a ordem de leitura fiável e com uma excelente capacidade de extração. Muitos dos problemas encontrados e referidos não são impactantes, sendo que alguns até podem vir a ser corrigidos com um processamento posterior. Este acaba por ser preferível relativamente a todas as outras ferramentas.

No entanto, o Marker realizou a extração de forma significativamente mais lenta, com um tempo de extração médio de 40 minutos. Isto pode ser explicado devido à natureza do modelo de OCR Surya, que usa um modelo Transformer. Este modelo é significativamente lento em CPU, mas de acordo com a documentação oficial do mesmo, este é capaz de chegar a uma velocidade extremamente rápida de 25 páginas por segundo com uma GPU H100, que é precisamente aquilo que temos disponível no servidor. Assim, o Marker acaba por ser uma excelente opção, tendo em conta que podemos acelerar significativamente o processo de extração em GPU.

5 Conclusão

Este relatório, que documenta a escolha da ferramenta de extração é especialmente importante dado que, para além de servir como base, a própria pipeline de extração pode vir a alterar, tendo em conta possíveis imprevistos e constrangimentos.

Nós optamos pelo **Marker**, que foi a ferramenta com melhor desempenho nos critérios avaliados, mas é importante referir que esta foi apenas testada para algumas páginas de dois documentos distintos, sendo que é também a **ferramenta mais exigente a nível computacional**, assim como a mais lenta. Embora tenhamos GPU disponível na infraestrutura do servidor que pode vir a acelerar e suavizar o processo de extração, esta abordagem ainda não foi testada, nem para todo o dataset de documentos. Assim, **esta escolha ainda pode vir a alterar**, realçando mais uma vez a utilidade deste relatório, com base no mesmo, podemos vir a considerar outras opções previamente avaliadas como o Docling.

5.1 Considerações Futuras

As cinco ferramentas avaliadas são todas elas pipelines completas e pré-configuradas. Como estamos perante um dataset muito específico de documentos, seria benéfico considerar uma **pipeline customizada** ou **configurar com mais detalhe a pipeline existente do Marker**. Ao escolher ferramentas especializadas em certas fases do processo de extração, poderíamos superar possíveis limitações e erros atuais, e obter resultados mais enquadrados nos objetivos do projeto.

Um exemplo seria por exemplo ter uma pipeline como na seguinte figura. Poderíamos usar o Surya (OCR do Marker) diretamente, o que levaria à mesma qualidade de OCR do Marker, mas com controlo total sobre o output e sobre as outras fases da extração.

Fase	Ferramenta
Extração nativa	pdf2image / PyMuPDF
Deteção de layout	Surya Layout Model
Ordem de leitura	Surya Reading Order
OCR por região	Surya OCR, Tesseract (fallback)
Extração de tabelas	TableFormer (Docling) / Table Transformer
Output	JSON com hierarquia de secção

Tabela 7. Exemplo de pipeline customizada.

Esta abordagem requer obviamente muito esforço, tanto ao nível de pesquisa como de implementação, não sendo por isso uma prioridade e ficando como consideração.