PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Avaliação comparativa de ferramentas para extração de tabelas em documentos PDF

Paulo de Saldanha da Gama de Moura Vianna

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Ciência da Computação

Rio de Janeiro, dezembro de 2025

Paulo de Saldanha da Gama de Moura Vianna

Avaliação comparativa de ferramentas para extração de tabelas em documentos PDF

Relatório de Projeto Final, apresentado ao Departamento de Informática da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Augusto Cesar Espíndola Baffa

Rio de Janeiro,
dezembro de 2025.

Resumo

VIANNA, Paulo S. G. M. BAFFA, Augusto (prof. orientador). Avaliação comparativa de ferramentas para extração de tabelas em documentos PDF. Rio de Janeiro, 2025. 43 p. Relatório de Projeto Final de Graduação. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho apresenta uma avaliação comparativa de ferramentas para extração de tabelas em documentos PDF financeiros brasileiros. Foram avaliadas ferramentas baseadas em regras geométricas, em deep learning especializado (IBM Docling) e em modelo multimodal, seguindo a metodologia de quatro níveis de Göbel et al. (2012): detecção de página, localização, estrutura celular e conteúdo textual. Os experimentos utilizaram relatórios de Fundos de Investimento Imobiliário, caracterizados por tabelas irregulares e células mescladas. Os resultados evidenciam diferenças significativas entre as abordagens e os desafios persistentes na extração automatizada de tabelas financeiras.

Palavras-chave:
Ferramentas, PDF, Documentos Financeiros, Certificado, Extração, Tabela, Detecção.

Abstract

VIANNA, Paulo S. G. M. BAFFA, Augusto. Comparative evaluation of tools for table extraction in PDF documents. Rio de Janeiro, 2025. 43 p. Relatório de Projeto Final de Graduação. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

This paper presents a comparative evaluation of table extraction tools for Brazilian financial PDF documents. The study assessed geometric rule-based tools (Camelot, Tabula, pdfplumber), specialized deep learning (IBM Docling), and a multimodal model (Google Gemini), following the four-level methodology proposed by Göbel et al. (2012): page detection, localization, cell structure, and textual content. Experiments were conducted using Real Estate Investment Fund (FII) reports, which are characterized by irregular tables and merged cells. The results highlight significant differences between the approaches and reveal the persistent challenges in the automated extraction of financial tables

Keywords:

Neural Networks, Extraction Tools, PDF, Financial Documents, Table, Detection, Extraction.

Sumário

1. Introdução

Na era digital, a disseminação de informações é mais abundante do que em qualquer outra época da história, sobretudo porque existem meios de comunicação imediatos e de amplo alcance, como a internet. Uma grande parte dessa difusão é feita por documentos eletrônicos, e o conteúdo que antigamente era veiculado em jornais e revistas de papel, hoje também é entregue aos consumidores por meio de artigos, vídeos e registros virtuais.

Há diversas formas de armazenar e apresentar esses documentos, de modo que a escolha de um determinado formato depende em grande parte das características e qualidades que cada opção oferece em relação ao que será transmitido.

O PDF foi projetado para manter o layout de documentos independentemente do dispositivo ou software usado para visualização (Adobe Systems Inc., 2006). Conforme Bodê (2005, p. 69), a independência de dispositivo significa que a aparência do documento permanece a mesma em qualquer plataforma, enquanto o auto-conteúdo garante que tudo necessário para visualização esteja incluído no arquivo. Diferentemente de formatos estruturados como CSV e JSON, que seguem regras claras de delimitação e permitem acesso direto aos dados, o PDF utiliza uma base de coordenadas para posicionar textos e elementos gráficos, priorizando a preservação visual ao custo de uma menor estruturação de dados.

O PDF foi projetado para manter o *layout* de documentos, independentemente do dispositivo ou *software* usado para visualização, conforme descrito pela *Adobe Systems Inc*., empresa criadora do formato. Essa extensão, em contraste com opções como CSV e JSON, se destaca por sua ênfase na preservação visual, ao custo de uma menor estruturação de dados. Isso porque o PDF utiliza uma base de coordenadas para posicionar textos e elementos gráficos na página (Adobe System Inc., 2006).

Por apresentar informações de forma coesiva para um leitor humano, o PDF se tornou um recurso frequentemente usado por muitas empresas e governos para a disseminação de seus dados. Sobre isso:

> Nowadays, government agencies, business corporate companies and personal individual disseminate their electronic documents in Portable Document Format (PDF) as the quickest and convenient way to publish information including text, tables and graphical images (Mohemad *et al.*, 2011, p. 1).

O problema aparece justamente quando se nota que o PDF reúne informações de modo não estruturado. Quando uma tabela está presente em um documento PDF, ela é armazenada, no melhor dos casos, célula por célula, como pedaços de texto com coordenadas definidas. Isso significa que a extração requer técnicas avançadas e, frequentemente, conhecimento prévio da estrutura do documento específico.

95.5% of published articles in PDF format are not semantically tagged. Therefore, extracting information from these documents remains a difficult problem. This is particularly important for tables that are generated taking into account the semantic information (e.g. from LATEX or MSWord) that is lost in the PDF. As a consequence, Table Extraction (TE) is still challenging (Gemelli et al., 2002).

Já no pior dos casos, a tabela pode ser representada por uma imagem, o que causa complicações por conta do uso de ferramentas OCR para a identificação e a extração de dados. Ou seja, a estrutura rígida e formatada dos arquivos PDF torna complexo o acesso a informações básicas em arquivos desse tipo. Esse desafio é ainda mais evidente quando se trata de tabelas, como constatam Li *et al.* (2020, tradução própria): “A análise (…) é uma tarefa difícil devido às variações nos layouts e formatos das tabelas”.

Tabelas podem apresentar grandes diferenças em termos de estrutura, estilo e formato, dificultando a criação de soluções generalizadas para sua extração. A complexidade aumenta quando se passa a lidar com tabelas irregulares, que se caracterizam por possuir uma quantidade variável de colunas por linha, possibilidade de alinhamentos inconsistentes, células mescladas e tipagem de dados que não seguem uma padronização, o que as torna algo mais comparável a uma planilha do que a um banco de dados.

Estudos recentes evidenciam a detecção e o reconhecimento de tabelas como tarefas críticas para muitas aplicações, incluindo análise de documentos e mineração de dados, já que as tabelas frequentemente apresentam informações essenciais de maneira não estruturada (Li *et al.*, *2020*). Isso reflete a crescente demanda por técnicas avançadas de extração, sobretudo em áreas como o mercado financeiro e consultorias, nas quais a precisão dos dados é fundamental para a tomada de decisões.

No mercado financeiro, que necessita de muitas informações relacionadas às empresas e ao governo, é de grande importância que os registros publicados por autarquias, como a Comissão de Valores

Mobiliários (CVM), ou pela Bolsa de Valores Brasileira (B3) sejam extraídos para alimentar modelos que influenciam decisões de analistas e gestores. Muitos desses conteúdos são apresentados em documentos PDF, o que gera dificuldades significativas quando seus dados precisam ser processados de forma automatizada.

Cabe considerar que muitos dos documentos entregues às instituições supracitadas, como os Relatórios Gerenciais publicados na Fundos.net[1] — um *site* que serve de canal para envio de documentos de fundos para a B3 e CVM — estão acessíveis no formato PDF e contêm informações quantitativas que podem ser encontradas tanto na forma de tabela quanto em textos avulsos.

As informações sobre Certificados Recebíveis Imobiliários (CRIs) e Certificados Recebíveis do Agronegócio (CRAs), que se encontram nos Relatórios Gerenciais acima mencionados, geralmente são apresentadas em tabelas dentro de arquivos PDF. Por isso, é necessário o uso de ferramentas de extração apropriadas, a fim de que os dados sejam obtidos corretamente e com consistência.

No entanto, mesmo recursos avançados, como Tabula[2] e Camelot[3], frequentemente falham ao tentar extrair dados de tabelas irregulares, ou seja, que não seguem um padrão uniforme. Isso exige desenvolver soluções específicas que possam lidar com a variação de formatos e a falta de estrutura típica de arquivos PDF. Tais ferramentas de extração facilitam o processo, mas ainda existem casos em que a intervenção humana é necessária para ajustar e validar os elementos de forma adequada, principalmente quando se trata de relacionar os dados de uma tabela com informações presentes em outras partes do documento. O artigo *Table Extraction and Understanding for Scientific and Enterprise Applications*, de Burdick *et al*. (2020, tradução própria), destaca essa situação ao afirmar que:

> (…) embora os humanos possam facilmente identificar, interpretar e contextualizar tabelas, o desenvolvimento de técnicas automatizadas e generalizadas para extração de informações é difícil devido à grande variedade de formatos de tabelas usados em corporações.

[1] https://fnet.bmfbovespa.com.br/fnet/publico/abrirGerenciadorDocumentosCVM
[2] https://tabula-py.readthedocs.io/en/latest/tabula.html. Acesso em 23/11/2025.
[3] https://camelot-py.readthedocs.io/en/master/. Acesso em 23/11/2025.

A Figura 1 ilustra a diferença entre dados estruturados, semiestruturados e não estruturados. Tabelas irregulares, com suas variações de colunas e células mescladas, comportam-se como dados semiestruturados, enquanto tabelas regulares equivalem a dados estruturados.

Figura 1 – Representação visual da diferença entre dados estruturados, semiestruturados e não estruturados

Fonte: Dataside[4]

Este trabalho se propõe a fazer um exame comparativo entre as diferentes ferramentas de extração de tabelas em documentos PDF, com ênfase na detecção e validação de tabelas irregulares. Em vez de desenvolver um novo modelo de rede neural, foram avaliadas soluções já existentes, como o Docling da IBM, que suporta a extração de estruturas de tabelas, e o OCR para PDFs escaneados e exportação para formatos estruturados. Além disso, foi realizada uma análise do uso de *Large Language Models* (LLMs), como o Google Gemini, na extração de artefatos para formatos estruturados, uma vez que esses modelos detêm a capacidade de “compreender” documentos em sua integridade.

Com base na revisão de literatura e nas dificuldades enfrentadas por abordagens tradicionais (*e.g.,* Tabula e Camelot), este estudo compara

[4] https://www.dataside.com.br/dataside-community/big-data/tipos-de-dados-estruturados-semi-estruturados-e-nao-estruturados. Acesso em 23/11/2025.

recursos atuais de referência com soluções consideradas avançadas, como modelos de linguagem e ferramentas que os utilizam desses modelos.

A comparação é conduzida nos três eixos clássicos da tarefa: (i) *Detecção da tabela*, (ii) *Reconhecimento da estrutura* (número de linhas e colunas, mesclagens e cabeçalhos) e (iii) extração do conteúdo (valores e tipagem), formalmente nomeada *Interpretação Tabular*, com foco em tabelas de relatórios gerenciais que listam CRIs e CRAs, além de outras tabelas relacionadas a fundos de investimento.

Neste trabalho, inicialmente apresentamos a situação atual da área, abordando as dificuldades inerentes ao formato PDF, as tarefas de extração e as ferramentas representativas existentes (Seção 2). Em seguida, definimos os objetivos e o escopo da avaliação proposta (Seção 3), detalhamos o plano de ação e a metodologia adotada (Seção 4), e descrevemos a implementação experimental com os resultados obtidos (Seção 5). Por fim, apresentamos as conclusões e direções para trabalhos futuros (Seção 6).

2. Situação Atual

2.1 Dificuldades inerentes ao formato PDF

O PDF não foi criado com a intenção de ser um arquivo estruturado de dados, o que resulta em dificuldades para a utilização de elementos como texto e, particularmente, tabelas. Por não ser um formato semântico, e sim de *layout*, isto é, por armazenar informações sobre como desenhar o documento, acaba ocorrendo uma omissão das relações estruturais, que são de grande importância para a extração, como a ordem de leitura, fronteiras de células, cabeçalhos, *spans* etc.

Devido à estrutura interna do PDF, não há uma forma de garantir que o conteúdo extraído atenda ao intuito do autor do documento, de modo que ferramentas de extração precisam fazer diversas suposições nesse caso.

Cabe considerar que, além dos *PDFs nativos*, que são escritos em um formato similar ao HTML e, portanto, permitem identificar os diferentes elementos que compõem uma página, existem também os *PDFs imagem*, que, como o nome indica, têm todo o seu conteúdo representado por fotos, o que exige uma abordagem ainda mais trabalhosa de extração.

2.2 Tarefas de extração

Para uma extração ser considerada bem-sucedida, existem métricas que devem ser abordadas com a finalidade de garantir o melhor resultado possível. A extração de tabelas em documentos é uma tarefa complexa que requer diversas etapas, como exemplificado pelo artigo *A methodology for Evaluating Algorithms for Table Understanding in PDF Documents* (Göbel *et al.*, 2012, tradução própria):

> It is commonly recognized that table understanding consists of three tasks of increasing complexity:
>
> - table detection: locating the regions of a document with tabular content
> - table structure recognition: Reconstructing the cellular structure of a table
> - table interpretation: rediscoverign the meaning of the tabular structure. This includes:
>
> (a) Functional Analysis: Determining the function of cells and their abstract logical relationships.
>
> (b) Semantic interpretation: Understanding the semantics of the table in terms of the

entities represented in the table, their attributes, and the mutual relationships between such entities.

Em resumo, é necessário que sejam realizadas a detecção, o reconhecimento e, finalmente, a extração do conteúdo, com cada uma dessas etapas possuindo sua própria metodologia de avaliação.

2.2.1 Detecção

A etapa de detecção consiste em localizar, dentro da página, as regiões que contêm conteúdo tabular. A avaliação deve comparar as caixas previstas com as caixas de referência (*ground truth*), usando correspondência 1-1 que maximize a *IoU* (*intersection-over-union*, isto é, a área de sobreposição entre duas tabelas sobre as suas áreas unidas). Nesse sentido, tem-se:

$$IoU = \frac{Área\ de\ Interseção}{Área\ de\ União}$$

Isso significa que quanto mais a área da tabela extraída se aproximar da área descrita no *ground truth*, melhor o resultado da extração. Se o *output* indicar uma área que está maior, menor ou deslocada da prevista, o valor do índice do ***IoU*** será mais baixo, o que o torna uma das métricas mais diretas de avaliação dessa tarefa.

2.2.2 Reconhecimento da estrutura

Esta seção visa reconstruir a malha celular (linhas x colunas), incluindo *spans* (*row/colspan*, células mescladas) e delimitação das células. Por se tratar da detecção de uma maior quantidade e granularidade de elementos, os erros que podem ocorrer são variados e, desse modo, cada item deve ser avaliado.

É necessário o uso de uma metodologia que capture tal granularidade e efetue comparações entre elementos e seus vizinhos. O *paper* de Göbel *et al.* (2012) propõe uma abordagem dinâmica que compara uma célula com seus vizinhos imediatos (adjacentes), avaliando se a direção e o conteúdo de uma célula condizem com o *ground truth*. A partir disso, são realizados os cálculos das métricas de *Recall* e *Precision* com as fórmulas a seguir:

Fórmula 1 - *Recall*

$$Recall = \frac{Relações\ de\ Adjacência\ Corretas}{Total\ de\ Relações\ de\ Adjacência}$$

Fórmula 2 - Precisão

$$Precision = \frac{Relações\ de\ Adjacência\ Corretas}{Relações\ de\ Adjacência\ Detectadas}$$

Figura 2 – Demonstração para ilustrar a comparação da tabela gabarito com a tabela extraída e apontar um caso de esturtura celular incorretamente detectada

Description	Initial balance	Increase	Decrease	Final balanc
Accrued income	1 669	0	1 269	40
Deferred income	26 676	0	26 079	59
Accrued expenses	49 734	0	14 467	35 26

(a) Original table as in ground truth

Description	Initial balance	Increase	Decrease	Final balanc
Accrued income	1 669	0	1 269	40
Deferred income	26 676	0	26 079	59
Accrued expenses	49 734	0	14 467	35 26

(b) Incorrectly recognized cell structure with split column

■ Correct adjacency relations □ Incorrect adjacency relations

$$\text{Recall} = \frac{\text{correct adjacency relations}}{\text{total adjacency relations}} = \frac{24}{31} = 77.4\%$$

$$\text{Precision} = \frac{\text{correct adjacency relations}}{\text{detected adjacency relations}} = \frac{24}{28} = 85.7\%$$

Fonte: Göbel *et al.*, 2012.

2.2.3 Interpretação tabular

A interpretação tabular corresponde à etapa final do processo de extração de tabelas, sendo responsável por atribuir significado funcional e semântico às células já identificadas e estruturadas. Após a detecção e o reconhecimento da grade da tabela, essa fase busca compreender o papel de cada célula dentro da estrutura (por exemplo, distinguir cabeçalhos de

valores, identificar totais ou notas) e, quando possível, inferir o tipo de dado representado, como texto, moeda, percentual e data.

Ainda de acordo com Göbel *et al*. (*2012*), essa etapa é tradicionalmente dividida em duas subtarefas principais: *análise funcional* (*functional analysis*), que determina a função lógica das células dentro da tabela, e *interpretação semântica* (*semantic interpretation*), que tenta compreender o conteúdo associando os valores a conceitos do domínio, como nomes de empresas, indicadores financeiros ou períodos de tempo.

Em suma, a interpretação tabular é responsável por transformar a estrutura geométrica detectada nas etapas anteriores em uma representação lógica e compreensível, além de permitir converter as tabelas extraídas de um PDF em dados estruturados e utilizáveis para outras tarefas.

2.3 Fundamentos técnicos

As ferramentas de extração de tabelas diferem principalmente nos sinais do PDF que utilizam e no tipo de raciocínio que aplicam para resolver as tarefas descritas na seção anterior. Essas diferenças produzem respostas distintas quando lidam com tabelas regulares, irregulares ou degradadas, e explicam por que os resultados variam tanto entre métodos tradicionais, modelos de visão e modelos de linguagem.

As abordagens tradicionais baseadas em *layout* geométrico utilizam apenas elementos explícitos no PDF: coordenadas de texto, espaçamentos, linhas vetoriais e padrões simples de alinhamento. Métodos como *Camelot* e *Tabula* assumem que colunas são formadas por blocos verticalmente alinhados ou por linhas de grade claramente visíveis. Essas heurísticas funcionam bem para tabelas regulares, mas perdem robustez quando há células mescladas, cabeçalhos em múltiplos níveis ou páginas densas com notas dispersas.

Em contraste, modelos especializados de visão computacional — como TableFormer no *framework* Docling — processam a página como imagem e aprendem padrões estruturais que não dependem de linhas de grade nem de alinhamentos perfeitos. Esses modelos usam arquiteturas baseadas em *Transformers* para representar relações entre elementos visuais, o que lhes permite inferir células e *spans,* mesmo quando a estrutura não está claramente marcada no PDF.

Por fim, modelos multimodais e de linguagem, como Gemini, seguem uma abordagem mais semântica: combinam a leitura do conteúdo textual com o entendimento visual global da tabela. Tais modelos conseguem interpretar cabeçalhos complexos, inferir o tipo de dado presente e até corrigir imprecisões estruturais, embora ainda sejam suscetíveis a erros de arredondamento, perda de sinais financeiros ou confusão em tabelas muito densas.

Essas três famílias, *geométrica*, *visual* e *multimodal*, correspondem a diferentes maneiras de atacar o mesmo conjunto de tarefas descrito na seção 2.2. Entender essas diferenças é essencial para interpretar as avaliações realizadas neste trabalho e para compreender por que determinadas ferramentas se destacam em cenários específicos.

2.4 Ferramentas representativas

Existem atualmente diversas ferramentas e bibliotecas disponíveis que permitem a extração de dados tabulares de PDFs, como Tabula, Camelot e PyMuPDF. Esses recursos apresentam soluções para extração de tabelas usando diferentes abordagens.

O Tabula, por exemplo, é uma opção de código aberto amplamente utilizada que oferece uma interface gráfica simples e capacidade de exportar dados para formatos como CSV e Excel. No entanto, sua eficácia está restrita a tabelas regulares.

O Camelot é similar ao Tabula, mas a sua principal diferença consiste em oferecer maior customização por meio de um maior número de parâmetros que podem ser definidos pelo usuário, o que, em teoria, lhe permite ser melhor adaptado para situações específicas.

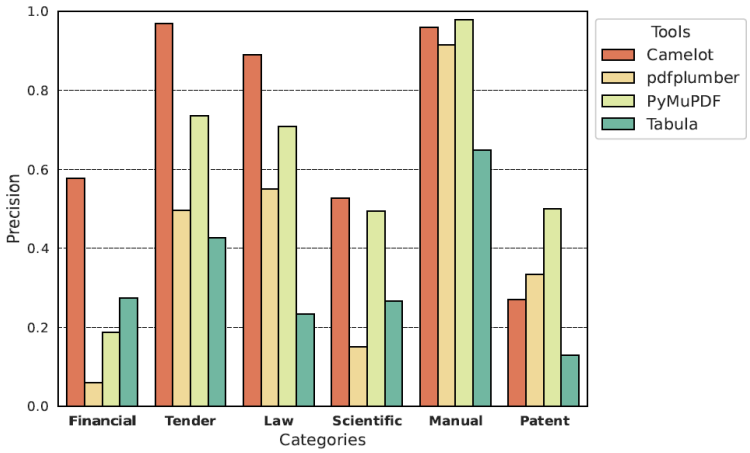
2.5 Falhas das abordagens tradicionais

As ferramentas supracitas funcionam bem com tabelas regulares, ou seja, aquelas que têm uma estrutura consistente, com um número fixo de colunas e uma disposição visual clara. No entanto, quando se trata de tabelas irregulares, que possuem variação no número de colunas, ordenação desalinhada ou células mescladas, tais opções muitas vezes falham em capturar dados corretamente.

Um estudo feito por Adhikari e Agarwal (2024) apontou que ferramentas de extração de tabelas em documentos PDF possuem uma

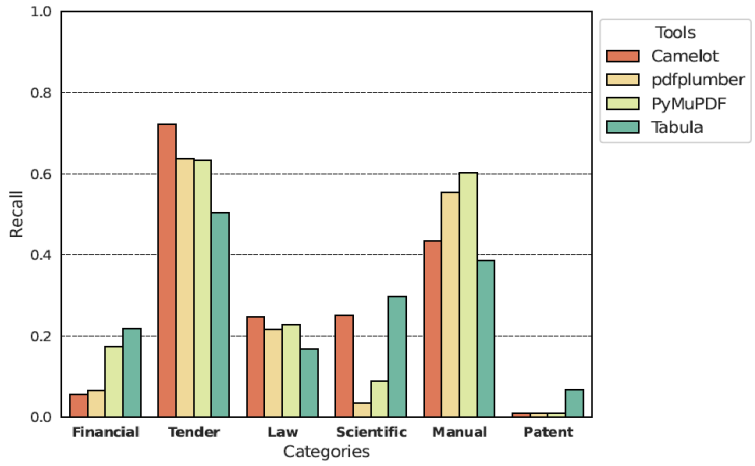
precisão significantemente menor do que as de extração de texto. Um outro fator importante que o artigo revela é que esses recursos registram um nível extremamente baixo de sensibilidade, apresentando um grande número de casos falso-negativos, o que não só resulta em perda de dados para o usuário, como há a possibilidade de ele mesmo sequer reconhecer que um documento possui dados. As figuras 3 e 4 representam esses achados:

Figura 3 – Precisão das ferramentas de extração de tabelas em PDF por tipo de documento



Fonte: Adhikari e Agarwal, 2024 (imagem 9)

Figura 4 – Sensibilidade das ferramentas de extração de tabelas em PDF por tipo de documento



Fonte: Adhikari e Agarwal, 2024 (imagem 10)

É importante notar que o referido estudo sobre essas ferramentas classifica os tipos de documentos que foram usados para a análise. Nesse sentido, as categorias de documentos financeiros e de patentes foram as que mais apresentaram problemas. O artigo não especifica o motivo para tais resultados, mas é fatídico que a falta de padronização e de estrutura clara seja um fator que contribui para essas métricas.

Em testes realizados com essas ferramentas para extrair dados de Certificados Recebíveis Imobiliários e Certificados Recebíveis do Agronegócio (CRIs e CRAs), disponíveis em relatórios gerenciais de fundos imobiliários, observou-se que as tabelas regulares são obtidas com precisão e consistência. No entanto, ao lidar com tabelas irregulares, o Tabula e o Camelot demonstraram limitações claras. Em muitas situações, os dados foram misturados entre colunas e a extração exigiu ajustes manuais para que fosse possível utilizá-los adequadamente.

Esse problema se torna mais grave porque os relatórios gerenciais publicados por fundos no Brasil não seguem uma padronização em seus formatos de apresentação. Tabelas nesses casos podem variar drasticamente, tanto em termos de estilo visual quanto na forma como os dados são dispostos. Não há uma obrigação explícita para que os fundos divulguem suas informações em arquivos CSV ou XML, o que poderia facilitar a extração automatizada. Consequentemente, esses relatórios muitas vezes estão em PDFs, com tabelas que podem ter espaçamentos inconsistentes e células mescladas, complicando o processo de extração.

2.6 Inovações das LLMs na tarefa de extração

As abordagens tradicionais baseadas em heurísticas e regras fixas apresentam limitações na interpretação de tabelas irregulares e na generalização para diferentes *layouts*.

Com o avanço do *deep learning*, surgiram modelos capazes de aprender representações estruturais e semânticas de forma automática. Um exemplo é o TableFormer, desenvolvido pela IBM dentro do *framework* Docling, que utiliza arquiteturas *Transformer* para identificar funções de células e reconstruir a lógica interna de tabelas complexas com alta fidelidade.

Mais recentemente, os Modelos de Linguagem de Grande Escala

(LLMs) ampliaram as possibilidades dessa etapa, já que conseguem analisar o conteúdo textual das células, realizar inferências sobre os dados e até responder perguntas sobre as tabelas. Li *et al.* (2024) destacam que a integração de LLMs com Modelos Visuais (VLMs) tem permitido interpretações mais completas de tabelas em documentos multimodais, combinando raciocínio textual, visual e estrutural.

2.7 Motivação para este trabalho

Existem duas motivações principais para este projeto. A primeira é a crescente adoção de Modelos de Linguagem de Grande Escala em tarefas de análise financeira, o que aumenta a demanda por dados estruturados e de alta qualidade, já que tais recursos dependem da consistência e confiabilidade das tabelas usadas em seus *pipelines*.

A segundo é a reconhecida dificuldade de extrair de forma precisa informações tabulares de documentos em PDF, extensão amplamente utilizada para divulgar relatórios gerenciais, demonstrações, materiais regulatórios, entre outros registros financeiros. O problema é que nesse formato as tabelas não possuem estrutura explícita e precisam ser reconstruídas a partir do *layout* do arquivo.

A combinação desses fatores evidencia a importância de analisar a qualidade das ferramentas de extração disponíveis, uma vez que erros nesse procedimento comprometem tanto análises tradicionais quanto aplicações baseadas em LLMs.

Como discutido nas subseções anteriores, essa etapa é suscetível a falhas que impactam diretamente análises quantitativas, auditorias e qualquer *pipeline* que utilize LLMs para resumo, consulta ou interpretação de dados.

Diante disso, torna-se necessário avaliar até que ponto ferramentas de extração conseguem produzir representações tabulares confiáveis em um domínio sensível como o de relatórios financeiros com informações de CRIs, CRAs e outros ativos. A motivação central deste trabalho, portanto, é analisar a qualidade dessas extrações e verificar sua adequação para usos posteriores, tanto em análises tradicionais quanto em aplicações baseadas em LLMs.

3. Objetivos / Proposta

3.1 Contexto do problema e escopo da avaliação

A heterogeneidade estrutural presente nos relatórios gerenciais de fundos, especialmente naqueles que apresentam informações de CRIs e CRAs, torna a extração automática de tabelas um desafio recorrente. Esses documentos combinam tabelas regulares e irregulares, com variações de *spans*, cabeçalhos multilinhas, colunas desalinhadas, notas agregadas ao corpo e formatação heterogênea entre páginas e emissões. Além disso, diferentes versões de um mesmo relatório podem alternar entre PDFs nativos e documentos digitalizados, ampliando a variabilidade discutida na seção 2 e dificultando a aplicação consistente de ferramentas tradicionais.

Neste trabalho, a avaliação concentra-se nesses cenários reais de uso, caracterizados por tabelas financeiras que precisam ser convertidas para formatos estruturados com alto grau de fidelidade. O estudo abrange ferramentas representativas de três abordagens técnicas: métodos geométricos baseados em análise de *layout* (como *Camelot, Tabula e PyMuPDF*); modelos de visão treinados para detecção e reconstrução estrutural (como o *IBM Docling* e seu modelo *TableFormer*); e modelos multimodais e linguísticos capazes de interpretar documentos completos, como o Gemini. Essas ferramentas foram aplicadas ao mesmo conjunto de tabelas extraídas de relatórios gerenciais recentes, permitindo observar como cada abordagem lida com irregularidades comuns nesse domínio.

O público-alvo desta análise inclui profissionais que dependem da conversão precisa de tabelas financeiras para fluxos de trabalho analíticos e pesquisadores interessados em compreender os limites de desempenho das ferramentas atuais. O escopo estabelecido nesta seção orienta a definição dos objetivos apresentados na seção 3.2 e delimita os cenários considerados nas comparações posteriores.

3.2 Objetivos

A avaliação conduzida neste trabalho tem como objetivo geral medir, de forma rigorosa e reprodutível, o desempenho de diferentes ferramentas de extração de tabelas aplicadas a relatórios gerenciais de fundos, com ênfase em tabelas irregulares que descrevem CRIs, CRAs e outras informações financeiras. A organização dos objetivos segue a orientação clássica da literatura para *table understanding* em documentos PDF, conforme discutido por *Göbel et al. (2012),* que estrutura a tarefa em três estágios: *detecção, reconhecimento da estrutura* e *extração do conteúdo.*

No primeiro eixo, *detecção*, o trabalho busca avaliar a capacidade das ferramentas de identificar corretamente as regiões tabulares presentes na página. A análise considera a correspondência entre as regiões previstas e as regiões anotadas como referência, respeitando critérios estabelecidos em estudos anteriores e aplicando protocolos de pareamento consistentes para comparação dos resultados.

No segundo eixo, *reconhecimento da estrutura*, o objetivo é medir o grau de fidelidade na reconstrução da malha de células, incluindo linhas, colunas, *spans*, múltiplos níveis de cabeçalho e alinhamentos irregulares. Essa etapa é particularmente relevante para tabelas financeiras dos relatórios analisados, que frequentemente apresentam variações de formato, células mescladas e seções heterogêneas. A avaliação foca em erros de granularidade fina, como divisões incorretas de células, detecção incompleta de linhas e colunas, e confusões na hierarquia de cabeçalhos.

No terceiro eixo, *extração do conteúdo*, a análise concentra-se na precisão dos valores textuais e numéricos obtidos, considerando normalizações necessárias para o domínio financeiro, tais como separadores de milhar, variações de escala e unidades monetárias. A avaliação compara os valores extraídos com a referência anotada após um processo controlado de padronização dos dados, de modo a isolar erros da extração propriamente dita.

Além dos três eixos centrais, o estudo inclui um objetivo transversal: examinar o comportamento das ferramentas em contextos difíceis, como tabelas sem linhas de grade, quebras tipográficas, notas incorporadas ao corpo da tabela, células multilinhas e diferenças visuais entre versões de um mesmo relatório. A inclusão desses casos reflete a diversidade encontrada em documentos financeiros reais.

Por fim, este trabalho objetiva comparar abordagens tradicionais e

modelos recentes baseados em visão ou linguagem, analisando, sob critérios homogêneos, as ferramentas *Camelot, PyMuPDF, Docling* e *Gemini*. Essa análise busca esclarecer em quais cenários os métodos geométricos permanecem competitivos e em quais contextos modelos mais avançados oferecem vantagens consistentes. O conjunto de resultados será utilizado para oferecer recomendações práticas de uso e orientar a seleção de ferramentas conforme o tipo de tabela e a qualidade do documento.

3.3 Relação com o estado da arte e avanços buscados

Os desafios apresentados pelas tabelas de relatórios gerenciais — irregularidade estrutural, variações visuais, ausência de linhas de grade e presença de notas que interferem no alinhamento — refletem limitações amplamente documentadas na literatura de extração de tabelas em PDF. Trabalhos como os de *Göbel et al. (2012)* e levantamentos posteriores, incluindo o de *Burdick et al. (2020),* enfatizam a necessidade de metodologias de avaliação mais consistentes, conjuntos de dados anotados e comparações sistemáticas entre diferentes famílias de ferramentas. O presente estudo alinha-se a essas diretrizes ao estruturar sua análise nos três eixos clássicos do problema e ao aplicar um protocolo de avaliação uniforme para opções heterogêneas.

A escolha das ferramentas analisadas indica a diversidade de abordagens atualmente disponíveis. Métodos geométricos, como *Camelot* e *PyMuPDF*, seguem princípios tradicionais baseados em alinhamento textual, detecção de espaçamento e identificação de linhas de grade. Essas técnicas continuam amplamente utilizadas na indústria pela transparência e maturidade, mas enfrentam limitações diante de tabelas irregulares e variações tipográficas frequentes em documentos financeiros.

Em contraposição, ferramentas recentes baseadas em visão computacional e modelos de linguagem ampliam as possibilidades de interpretação estrutural. O *IBM Docling*, por exemplo, incorpora modelos de detecção e de segmentação de *layout* treinados em *large-scale datasets*. Cabe ainda considerar o *TableFormer*, projetado especificamente para reconstrução estrutural de tabelas complexas. Já opções multimodais como o *Gemini* introduzem um paradigma distinto por serem capazes de analisar a página completa como contexto, sem depender exclusivamente da geometria explícita do PDF. A comparação desses métodos permite avaliar

até que ponto tais avanços representam melhorias concretas no domínio financeiro, caracterizado por formatos variados e inconsistência entre relatórios.

A contribuição buscada por este trabalho não está na criação de novos algoritmos, mas na consolidação de uma avaliação comparativa rigorosa, centrada em documentos reais e com anotações de referência específicas para a área de fundos. Isso permite identificar, com granularidade, quais ferramentas são adequadas para a detecção, quais preservam melhor a estrutura e quais extraem o conteúdo com maior fidelidade. O estudo também visa destacar casos que permanecem desafiadores mesmo para modelos recentes, fornecendo evidências práticas que podem orientar tanto o uso profissional quanto futuras pesquisas na área.

4. Revisão do Plano de Ação

4.1 Estudo e Seleção das Ferramentas

O primeiro passo no desenvolvimento do projeto foi a seleção das ferramentas apropriadas para lidar com a complexidade da extração de dados de documentos PDF. Diversas bibliotecas e *frameworks* foram considerados e testados. Entre as opções analisadas, a biblioteca *PDFMiner Six* foi escolhida para a extração do *layout* e do conteúdo textual dos documentos PDF, pois oferece uma abordagem detalhada e granular para a manipulação de elementos textuais e suas coordenadas.

Além disso, foram exploradas outras ferramentas já existentes, como *Tabula* e *Camelot*, a fim de compreender suas limitações e identificar os pontos em que o projeto poderia agregar melhorias, specialmente no tratamento de tabelas irregulares. Para esta analisei, optamos por usar somente o *Camelot*, devido a sua API de nível mais baixo, o que permite uma abordagem mais precisa.

Para a análise usando *LLM*, diversos fatores afetaram a minha decisão sobre qual ferramenta seria a ideal. Escolhemos o Gemini 2.5 Flash devido a sua alta performance, tanto em termos de tempo de execução quanto qualidade da extração, junto ao preço mais baixo comparado às outras APIs. Segue abaixo uma tabela comparativa dos custos de execução de cada modelo (a cada 1 milhão de tokens):

Tabela 1 – Tabela de custos dos modelos de linguagem

Empresa	Modelo	Input	Output
OpenAI	gpt-4o	$1.25	$10.00
OpenAI	gpt-4o-mini	$0.15	$0.75
Google	Gemini 2.5 Flash	$0.30	$2.50
Google	Gemini 2.5 Pro	$1.25	$10.00
Anthropic	Haiku 3.5	$0.80	$4.00
Anthropic	Sonnet 3.7	$3.00	$15.00

Observação: Seleção do modelo foi realizada em 25 de Junho de 2025. Modelos futuros não foram considerados.

Outro fator favorável à seleção do Gemini como modelo está na sua API intuitiva e que realiza a vetorização de documentos enviados, inclusive de PDFs.

Finalmente, escolhemos também o ***Docling***, que é uma ferramenta mais recente e muito robusta, capaz de parsear o documento em PDF e convertê-

lo para outros formatos, mas além disso, ele nos dá informações em alto nível de detalhes sobre o conteúdo do documento, inclusive sobre a presença de tabelas, junto com seus conteúdos e metadados.

4.2 Scraping de documentos para o grupo de testes

Os documentos usados como base para os testes das ferramentas foram os relatórios gerenciais de fundos presentes no site Fundos Net da Bovespa. Foi desenvolvido um simples algoritmo de webscraping para extrair uma quantia desejada de documentos, ordenados pela data de publicação de forma decrescente (mais recentes primeiro). O pseudocódigo para este algoritmo de scraping encontra-se a seguir:

Pseudocódigo 1 – Scraping Fundos NET

```
Algoritmo ScrapingFundosNet

Inicio
    sessao ← CriarSessaoHTTP()
    sessao.headers ← DefinirHeadersPadrao()

    tamanhoPagina ← TAMANHO_LISTA_RETORNO
    totalDocumentos ← ObterTotalDocumentos()
    numeroIteracoes ← Teto(totalDocumentos / tamanhoPagina)

    AcessarSite(sessao, idCategoria, idTipo)

    listaDocumentos ← []

    Para i de 0 até numeroIteracoes - 1 Faça

        numeroPagina ← i + 1
        numeroStart ← i * tamanhoPagina
        momentoRequisicao ← ObterCTime()

        url ← ConstruirURL(
            pagina: numeroPagina,
            start: numeroStart,
            tamanho: tamanhoPagina,
            categoria: idCategoria,
            tipo: idTipo,
            especie: idEspecie,
            ctime: momentoRequisicao
        )

        resposta ← sessao.GET(url)
        documentos ← ParseJSON(resposta)

        Para cada documento em documentos Faça

            idDocumento ← documento.id
            nomeOriginal ← documento.nome
            nomePregao ← documento.nomePregao
            dataReferencia ← documento.dataReferencia

            urlDocumento ← GerarURLDocumento(idDocumento)
            nomeArquivo ← Concatenar(nomePregao, "_", dataReferencia, "_", idDocumento)
            caminhoArquivo ← DefinirPath(nomeArquivo)

            registro ← {
                "url": urlDocumento,
                "caminho": caminhoArquivo
            }

            listaDocumentos.Adicionar(registro)

        Fim Para

    Fim Para

    dataFrame ← CriarDataFrame(listaDocumentos)

    Retornar dataFrame

Fim
```

Pseudocódigo 2 – Download dos documentos

```
Algoritmo DownloadDocumentos

Início
    sessao ← CriarSessaoHTTP()
    sessao.headers ← DefinirHeadersPadrao()

    listaThreads ← []

    Para cada documento em dataFrame Faça
        thread ← CriarThread(
            funcao: RealizarDownload,
            argumentos: (sessao, documento.url, documento.cami
        )
        listaThreads.Adicionar(thread)
        thread.Iniciar()
    Fim Para

    JuntarThreads(listaThreads)
Fim

Função RealizarDownload(sessao, url, caminho)
    resposta ← sessao.GET(url)
    EscreverArquivo(caminho, resposta.conteudo)
Fim Função
```

4.3 Planejamento dos testes das ferramentas

Serão realizados quatro tipos de avaliação para cada ferramenta. O primeiro será a sua capacidade de identificar a presença de tabela dentro das páginas, que é feito de duas formas:

(i) Comparar a lista de páginas encontradas por cada uma das ferramentas com a lista de páginas que realmente contém tabelas. Dessa examinação, extraímos os casos falso positivos e falso negativos também, pois nos ajudam a medir a acurácia.

(ii) Para páginas que contém tabela e o programa encontrou, comparamos a área da tabela encontrada pelo algoritmo com a área real da tabela, extraída de forma manual. Realizamos uma avaliação através de IoU (*intersection over union)* onde dividimos a área da união entre as bounding boxes encontradas pela ferramenta e as do gabarito sobre a interseção das duas (conjunto de coordenadas que indicam a localização da tabela).

(iii) Comparação estrutural, checar as adjacências de cada uma das células presentes na tabela. Contabilizamos a quantidade de adjacências

corretas dividimos pelo total de adjacências no gabarito.

(iv) Avaliação do conteúdo da tabela. Iremos célula a célula capturar o valor da tabela extraída pela devida ferramenta, normalizá-lo (isto é, para valores numéricos, converter vírgulas que dividem casas decimais para pontos, remover cifrão e definir o número de casa decimais para duas, caso existam), comparar com o seu respectivo valor na tabela gabarito. Para valores em formato de texto, terêmos que realizar uma comparação por *fuzzy* e verificar a acurácia do texto, que pode estar distorcido em casos que exijam o uso de *OCR*.

4.4 Criação do dataset de controle

Como o processo de validação manual deste tipo de extração é demorado, pois é necessário mapear os documentos um a um, para garantir um gabarito de alta qualidade, o teste foi limitado para 20 documentos que contém tabela, escolhidos de forma aleatória, priorizando uma grande diversidade de empresas.

Separamos cada tabela extraída desses documentos em outros documentos CSVs, cada um com uma nomenclatura que nos indica o documento de origem, número da página e ordem da tabela dentro da página por uma perspectiva top-bottom.

A extração das tabelas para o ground truth foi realizada utilizando o *IBM Docling*, uma ferramenta baseada em deep learning que fornece saída estruturada completa incluindo informações de células mescladas (rowspan/colspan). Essa abordagem semi-automática foi adotada devido à inviabilidade de anotação inteiramente manual dado o volume de dados. O processo incluiu validação por amostragem, com conversão das tabelas extraídas para formato Excel para inspeção visual. As limitações desta abordagem, incluindo o potencial viés introduzido, são discutidas na Seção 5.3.

Ao final, as tabelas extraídas são carregadas para *DataFrames* utilizando-se do módulo *Pandas* no *Python*. Usaremos desta ferramentas somente para normalizar os valores numéricos e de datas. Graças a funcionalidades já presentes dentro no *Pandas*, como a função *to_datetime*, conseguiremos realizar essa etapa de forma rápida e com acompanhamento exporádico. Em seguida, analisarei os documentos a fim de verificar se não foi cometido

algum erro durante a execução deste script.

4.5 Execução e Protocolo de Extração de Métricas

Para garantir a consistência experimental e a reprodutibilidade dos resultados, foi desenvolvido um pipeline de avaliação automatizado. Este pipeline recebe como entrada o documento PDF bruto e o arquivo de *Ground Truth* (Gabarito), submete o documento às ferramentas de extração e, por fim, executa algoritmos de comparação sobre os resultados normalizados.

A execução foi segregada em módulos específicos para cada família de ferramentas, respeitando suas particularidades de configuração e formato de saída, conforme detalhado a seguir.

4.5.1 Ferramentas Geométricas (Tabula e Camelot)

Essas ferramentas operam diretamente sobre a camada de objetos do PDF (*LTTextObject* e linhas vetoriais). O protocolo de execução foi definido para maximizar a capacidade de detecção em tabelas sem bordas (irregulares).

O *Camelot* será executado no modo ***flavor='stream'***, que utiliza heurísticas de espaçamento em branco (whitespaces) para inferir colunas, em vez de buscar linhas de grade (***lattice***), já que os relatórios financeiros analisados frequentemente carecem de demarcações visuais explícitas. O parâmetro **row_tol** (tolerância de linha) será ajustado para 10, permitindo agrupar textos com ligeiros desalinhamentos verticais.

Extração de Coordenadas (IoU):

Ambas as bibliotecas retornam, além do conteúdo, as coordenadas da tabela detectada. No caso do *Camelot*, o atributo *_bbox* da tabela extraída será capturado. Como o sistema de coordenadas do PDF pode variar (origem no canto inferior esquerdo vs. superior esquerdo), será aplicada uma transformação linear para normalizar todas as coordenadas para o sistema do *PDFMiner* (usado no *Ground Truth*), garantindo que o cálculo de *Intersection over Union (IoU)* seja geometricamente válido.

4.5.2 Ferramentas de Visão Computacional (IBM Docling)

O Docling introduz uma camada de abstração baseada em modelos de Deep Learning (TableFormer). A execução seguirá os passos:

Instanciação do Pipeline: Será instanciado um *DocumentConverter* configurado para permitir o uso de *OCR* (*ocr_enabled=True*) apenas como fallback, priorizando a extração programática para manter a fidelidade dos dados textuais.

Mapeamento de Objetos: O output do *Docling* é um grafo de documento serializado em JSON. O script de teste iterará sobre os nós do tipo TableItem.

Recuperação de Bounding Boxes: Diferente das ferramentas geométricas, o Docling pode identificar tabelas rotacionadas ou distorcidas. As coordenadas serão extraídas do metadado *prov* (provenance) de cada célula e, a partir delas, será calculada a *Bounding Box* envolvente mínima da tabela inteira (*min_x, min_y, max_x, max_y*) para o cálculo do IoU.

Serialização: A estrutura interna da tabela reconstruída será exportada para o formato *pandas.DataFrame*, preservando as informações de *row_span* e *col_span* (células mescladas), essenciais para a avaliação da métrica de estrutura.

4.5.3 Modelos de Linguagem Multimodais (Google Gemini)

Dada a natureza generativa e não determinística dos LLMs, o protocolo de execução foca na engenharia de prompt e no parsing de saída, excluindo-se a métrica de IoU (área), uma vez que modelos de linguagem não fornecem coordenadas espaciais precisas nativamente.

Engenharia de Prompt: Será utilizado o modelo Gemini 2.5 Flash via API. O prompt de sistema foi desenhado para atuar como um "Extrator de Dados Financeiros", com instruções estritas para:

- Receber a imagem da página do PDF como entrada;
- Identificar a presença de tabelas;
- Extrair o conteúdo exclusivamente em formato **JSON**, representando um array de dicionários, seguindo um schema predefinido (lista de listas para linhas e colunas);
- Abster-se de gerar texto conversacional (ex: "Aqui está a tabela...").

Pós-processamento: A resposta textual (string JSON) passará por um validador sintático. Respostas que não respeitarem o formato JSON ou alucinarem campos inexistentes serão marcadas como falha de extração. O JSON válido será convertido para DataFrame para comparação de conteúdo.

4.5.4 Algoritmos de Comparação (Métricas)

Após a normalização dos dados de todas as ferramentas para objetos DataFrame e listas de coordenadas, aplicam-se os algoritmos de avaliação:

Fórmula 3 - Interseção sobre união

$$\frac{Area\ (A \cap B)}{Area\ (A \cup B)}$$

Implementa-se a fórmula do IoU conforme demonstrado na ***Fórmula 3*** onde A é a caixa delimitadora do Ground Truth e B, a caixa retornada pela ferramenta.

Avaliação de Estrutura (Adjacências): Conforme proposto por *Göbel et al. (2012),* cada célula da tabela extraída é transformada em um nó de um grafo, contendo ponteiros para seus vizinhos (cima, baixo, esquerda, direita). Esse grafo é comparado com o grafo do gabarito. A precisão estrutural é definida pela razão entre o número de relações de vizinhança corretas e o total de relações detectadas.

Avaliação de Conteúdo (Fuzzy Matching): Antes da comparação, aplica-se um filtro de Regex para remover formatação monetária ("R$", separadores de milhar). Para valores numéricos, após o tratamento mencionado anteriormente, comparamos eles sem threshold algum. Para textos, utiliza-se a **Distância de Levenshtein**, considerando um match positivo se a similaridade for superior a 90%, mitigando penalidades por pequenos erros de encoding.

5. Implementação

Esta seção apresenta a avaliação experimental das ferramentas de extração de tabelas selecionadas. São descritos o dataset utilizado, a configuração das ferramentas, o processo de construção do ground truth, os resultados obtidos em cada dimensão de avaliação e uma análise comparativa do desempenho observado.

5.1 Configuração Experimental

5.1.1 Dataset de Avaliação

O conjunto de dados utilizado neste estudo é composto por relatórios gerenciais de Fundos de Investimento Imobiliário (FIIs) obtidos do portal Fundos Net, mantido pela B3 (Brasil, Bolsa, Balcão). Esses documentos foram selecionados por representarem um cenário real e desafiador para extração de tabelas: são PDFs nativos (não digitalizados) que contêm tabelas financeiras com formatação irregular, frequentemente sem bordas visíveis, com células mescladas e alinhamentos variados.

A coleta foi realizada por meio de um algoritmo de web scraping desenvolvido especificamente para este trabalho, conforme descrito na Seção 4. Os documentos foram ordenados por data de publicação de forma decrescente, priorizando relatórios mais recentes. A seleção final priorizou a diversidade de emissores, de modo a capturar variações estilísticas entre diferentes gestoras de fundos.

O dataset final é composto por 15 documentos PDF contendo tabelas, selecionados aleatoriamente dentre os relatórios coletados. A Tabela X apresenta as estatísticas descritivas do conjunto de dados.

Tabela 2 - Estatística do Dataset de Avaliação

Métrica	**Valor**
Total de Documentos	15
Total de páginas analisadas	287
Páginas contendo tabelas	125
Total de tabelas anotadas	178
Tota de células	10783
Média células/tabela	60.6
Tabelas com células mescladas	24

As tabelas presentes nesses documentos incluem demonstrações de resultados, composições de carteira, históricos de rendimentos e indicadores financeiros. A estrutura típica dessas tabelas apresenta cabeçalhos em múltiplas linhas, colunas de períodos temporais (meses ou trimestres), valores monetários formatados com separadores de milhar e casas decimais, além de células mescladas para agrupamento de categorias.

5.1.2 Ferramentas Avaliadas

Foram avaliadas quatro ferramentas representativas de diferentes abordagens tecnológicas para extração de tabelas em PDFs, conforme discutido na Seção 2 (Revisão da Literatura) e detalhado na Seção 4 (Metodologia). A Tabela 2 sumariza as ferramentas e suas configurações.

Tabela 3 - Ferramentas Avaliadas e Configurações

Ferramenta	**Versão**	**Abordagem**	**Configuração Principal**
Camelot	0.11.0	Geométrica	Flavor='stream', row_tol=10
Tabula (tabula-py)	2.9.3	Geométrica	stream=True
IBM Docling	2.60.0	Deep Learning (TableFormer)	Ocr_enabled=True (fallback) Do_table_structure=True mode = ACCURATE Do_cell_matching = False
Google Gemini	2.5 Flash	LLM Multimodal	Prompt Estruturado com saída JSON
Pdfplumber	0.11.8	Geométrica	Find_tables() padrão

Camelot é uma biblioteca *Python* que utiliza heurísticas geométricas para detectar tabelas. O modo stream foi selecionado por não depender de linhas de grade visíveis, utilizando espaçamentos em branco para inferir a estrutura de colunas. O parâmetro *row_tol=10* permite agrupar elementos textuais com pequenos desalinhamentos verticais em uma mesma linha lógica.

Tabula opera de forma similar ao Camelot, analisando a disposição espacial dos objetos de texto no PDF. O modo *stream=True* foi utilizado pelos mesmos motivos: adequação a tabelas sem bordas explícitas. A ferramenta é amplamente utilizada na indústria e em trabalhos acadêmicos como baseline para extração tabular.

Google Gemini 2.5 Flash é um modelo de linguagem multimodal capaz de processar documentos PDF como entrada visual. Diferente das demais ferramentas, o *Gemini* não realiza uma análise estrutural explícita do PDF, mas interpreta a página como uma imagem e gera uma representação estruturada da tabela com base em sua compreensão visual e semântica. A extração foi conduzida via API, utilizando um prompt sistemático que instrui o modelo a retornar as tabelas em formato JSON

estruturado.

pdfplumber é uma biblioteca *Python* construída sobre o pdfminer.six, oferecendo uma interface de alto nível para extração de texto e tabelas. Diferente do *Camelot* e *Tabula*, o *pdfplumber* permite acesso granular aos objetos do PDF (caracteres, linhas, retângulos) e implementa algoritmos próprios de detecção de tabelas baseados na identificação de interseções de linhas e agrupamento de texto. A ferramenta foi configurada com os parâmetros padrão de *find_tables(),* que utiliza heurísticas de espaçamento para inferir bordas implícitas em tabelas sem linhas de grade visíveis.

5.1.3 Ambiente de Execução

Os experimentos foram executados em um ambiente computacional com as seguintes especificações:

Sistema Operacional: *Windows 11*
Processador: *AMD Ryzen 5 4500*
Memória RAM: 32 GB
Linguagem: *Python 3.11*
Bibliotecas principais: *Pandas 2.3.3, NumPy 2.0.1, PyMuPDF (fitz) 1.26.6, PdfPlumber 0.11.8, tabula-py 2.10.0, camelot-py 1.0.9*

Para as ferramentas que utilizam modelos de *Deep Learning (Docling)*, não foi utilizada aceleração por GPU, de modo que os tempos de execução reportados refletem processamento em CPU. As chamadas à API do *Google Gemini* foram realizadas via rede, estando sujeitas à latência de comunicação. Para mitigar o tempo de execução, as chamadas foram realizadas utilizando a API paga com paralelização de até 20 *threads*.

5.2 Construção do Ground Truth

5.2.1 Estrutura do Ground Truth

Figura 5 – Estruturação do *Ground Truth* em JSON

```
Document {
    id_document: string
    filename: string
    pages: [
        Page {
            page_number: int
            width: float
            height: float
            tables: [
                Table {
                    bbox: {x0, y0, x1, y1}
                    rows: int
                    cols: int
                    cells: [
                        Cell {
                            row: int
                            col: int
                            rowspan: int
                            colspan: int
                            content: string
                        }
                    ]
                }
            ]
        }
    ]
}
```

O ground truth foi estruturado em formato JSON hierárquico, conforme ilustrado na **Figura *5***. Cada documento contém uma lista de páginas, e cada página contém uma lista de tabelas detectadas. Para cada tabela, são armazenados:

- *Bounding box*: coordenadas (x0, y0, x1, y1) da região da tabela na página, utilizadas para avaliação de localização (Nível 2);
- Dimensões: número de linhas e colunas da tabela;
- Células: lista completa de células, cada uma contendo:
  - Posição (row, col): índices da célula na grade;
  - Extensão (rowspan, colspan): número de linhas/colunas ocupadas;
    - Conteúdo textual: texto extraído da célula.

A inclusão de rowspan e colspan é essencial para a correta avaliação do Nível 3 (estrutura celular), pois permite reconstruir as relações de

adjacência entre células mesmo em tabelas com células mescladas.

5.2.2 Processo de Validação

O processo de validação seguiu três etapas:

a. **Extração Automática:** Todos os 15 documentos foram processados pelo *Docling*, gerando arquivos JSON com a estrutura completa de cada tabela detectada.

b. **Validação por Amostragem:** Uma amostra das tabelas foi inspecionado visualmente, convertendo-as para formato *Excel* com preservação de células mescladas para facilitar a verificação. Casos de erro evidente foram documentados.

c. **Exclusão de casos problemáticos**: Dentro dos documentos utilizados, as Tabelas em páginas com orientação landscape, para as quais o *Docling* não extraiu estrutura celular, foram identificadas e excluídas da avaliação dos Níveis 3 e 4.

5.2.3 Limitações Metodológicas

Reconhece-se que esta abordagem introduz um potencial viés favorável ao *Docling*, uma vez que ele é simultaneamente gerador do *ground truth* e poderia ser uma ferramenta avaliada. Por esta razão, o *Docling* foi excluído da comparação final, sendo utilizado exclusivamente como referência. O objetivo primário desta avaliação é a comparação relativa entre as ferramentas geométricas (*Camelot, Tabula, pdfplumber*) e o modelo *multimodal* (*Gemini*), para a qual o *ground truth* permanece válido como referência consistente.

Trabalhos futuros poderiam validar estes resultados utilizando anotação manual independente ou benchmarks públicos estabelecidos como *ICDAR* ou *PubTabNet*.

5.3 Resultados

Esta seção apresenta os resultados da avaliação comparativa das ferramentas selecionadas nos quatro níveis da metodologia de *Göbel et al.* (2012). As métricas reportadas são *Precision* (P), *Recall* (R) e *F1-score*, calculadas conforme descrito na Seção 2.2.

5.3.1 Nível 1: Detecção da Página

O primeiro nível avalia a capacidade das ferramentas de identificar corretamente quais páginas do documento contêm tabelas. A Tabela apresenta os resultados agregados.

Tabela 4 – Resultados de encontro das tabelas nas páginas

Ferramenta	Precision	Recall	F1-score
Camelot	0.5833	0.224	0.3237
Pdfplumber	0.5365	0.824	0.6498
Tabula	0.2857	0.0339	0.606
Gemini	0.6281	1.0	0.7716

5.3.2 Nível 2: Localização

As ferramentas geométricas apresentam IoU médio elevado quando há match, isto é, a tabela encontrada pela ferramenta é compatível com a encontrada pelo *Docling*. No entanto, devido à variedade de formatos de tabelas presentes nos documentos, algumas das quais estas ferramentas apresentam dificuldade na avaliação, como tabelas sem grades explícitas, esse match acaba sendo relativamente baixo.

O *Gemini* foi excluído desta análise pois ele é incapaz de analisar geograficamente o documento, portanto as suas bounding boxes eram aleatórias, muitas vezes geradas por uma análise do modelo em cima do conteúdo da tabela.

Das ferramentas incluídas nessa seção, o *Camelot*, com abordagem mais conservadora, detecta poucas tableas, mas apresenta uma alta precisão dos limites das tabelas, enquanto o *PDFPlumber* apresenta um maior recall, mas com muitos falsos positivos.

Tabela 5 - Resultados da localização dos limites das tabelas

Ferramenta	Precision	Recall	F1-score	IoU
Camelot	0.266	0.118	0.163	0.92
Pdfplumber	0.083	0.326	0.132	0.88
Tabula	0.013	0.012	0.012	0.59

5.3.3 Nível 3: Estrutura Celular

O teceiro nível avalia o reconhecimento da estrutura interna das tabelas através das relações de adjacência entre células, conforme metodologia de *Göbel et al. (2012).* Para cada par de células vizinhas (horizontal ouverticalmente), verifica-se se a relação foi corretamente identificada.

Tabela 6 - Resultados da avaliação da Estrutura Celular

Ferramenta	**Precision**	**Recall**	**F1-score**	**Adjacências Avaliadas**
Camelot	0.8526	0.3637	0.5099	4667
Pdfplumber	0.6402	0.3213	0.4279	20027
Tabula	0.206	0.5831	0.3044	1192
Gemini	0.933	0.9265	0.9298	20073

5.3.4 Nível 4: Conteúdo Textual

O quarto nível avalia a precisão do conteúdo textual extraído de cada célula.A comparação utiliza similaridade de Levenshtein normalizada com threshold de 0.7, conforme metodologia de *Meuschke et al. (2023)*. Antes da comparação, os textos são normalizados: conversão para minúsculas, remoção de símbolos monetários (R$), normalização de separadores decimais e remoção de espaços extras.

Tabela 7 - Resultados da comparação do conteúdo textual

Ferramenta	**Precision**	**Recall**	**F1-score**	**Adjacências Avaliadas**
Camelot	0.1895	0.0887	0.1209	3374
Pdfplumber	0.2808	0.1468	0.1928	13085
Tabula	0.0176	0.0457	0.0254	776
Gemini	0.7058	0.6976	0.7016	13747

5.3.5 Visão Consolidada

A Tabela 7 apresenta uma visão consolidada dos resultados (F1-Score) em cada um dos níveis de avaliação.

Tabela 8 - Resultados Consolidados (F1-Score)

Ferramenta	Nível 1	Nível 2	Nível 3	Nível 4
Camelot	0.1895	0.0887	0.5099	0.1209
Pdfplumber	0.2808	0.1468	0.4279	0.1928
Tabula	0.0176	0.0457	0.3044	0.0254
Gemini	0.7058	-	0.9288	0.7016

5.4 Análise e Discussão

Os resultados evidenciam uma diferença substancial de desempenho entre as abordagens avaliadas, com clara vantagem para o modelo multimodal.

Ferramentas geométricas: As ferramentas baseadas em regras (*Camelot, Tabula, pdfplumber*) apresentaram desempenho limitado em todos os níveis. O *pdfplumber* destacou-se entre elas com F1 de 0,65 na detecção de páginas, beneficiado por alto recall (0,82), porém às custas de muitos falsos positivos. O *Camelot* mostrou maior precisão estrutural (0,85) mas baixo recall, indicando uma abordagem conservadora que detecta poucas tabelas. O *Tabula* teve o pior desempenho geral, com F1 inferior a 0,10 na detecção de páginas e apenas 0,03 na extração de conteúdo, sugerindo inadequação para documentos financeiros com tabelas irregulares.

Modelo multimodal (Gemini): O *Google Gemini* superou todas as ferramentas geométricas por ampla margem. Na detecção de páginas, foi a única ferramenta a identificar todas as páginas com tabelas (recall = 1,0). No reconhecimento de estrutura, alcançou F1 de 0,93, demonstrando capacidade de reconstruir corretamente as relações de adjacência mesmo em tabelas com células mescladas. Na extração de conteúdo, obteve F1 de 0,70 — mais de três vezes superior ao segundo colocado (*pdfplumber* com 0,19). Estes resultados confirmam que a interpretação visual e semântica dos LLMs multimodais oferece vantagens significativas sobre abordagens puramente geométricas.

Os resultados são consistentes com a literatura recente: *Adhikari et al. (2024)* reportaram que documentos financeiros apresentam desafios particulares para ferramentas tradicionais, enquanto modelos baseados em visão computacional têm demonstrado avanços significativos neste domínio.

6 Conclusão e Trabalhos Futuros

A extração automatizada de tabelas em documentos PDF permanece um desafio relevante para aplicações que exigem precisão e escalabilidade, como análises financeiras, auditorias e integração de dados. Este trabalho apresentou uma avaliação comparativa sistemática de ferramentas representativas de diferentes paradigmas tecnológicos, aplicada a documentos financeiros reais do mercado brasileiro.

6.1 Contribuições do Estudo

A principal contribuição deste trabalho é a implementação de um framework de avaliação reproduzível, baseado na metodologia de quatro níveis proposta por *Göbel et al. (2012),* aplicado ao domínio específico de relatórios de Fundos de Investimento Imobiliário. O *framework* desenvolvido permite avaliar ferramentas de extração em múltiplas dimensões: detecção de páginas com tabelas, localização espacial via IoU, reconhecimento de estrutura celular através de relações de adjacência, e precisão do conteúdo textual utilizando similaridade de Levenshtein.

A avaliação comparativa abrangeu quatro ferramentas de três categorias distintas: abordagens geométricas tradicionais (*Camelot, Tabula, pdfplumber*), modelo de deep learning especializado (*IBM Docling* com *TableFormer*), e modelo de linguagem multimodal (*Google Gemini*). Os resultados demonstraram uma superioridade expressiva do modelo multimodal: o *Gemini* alcançou F1 de 0,93 no reconhecimento de estrutura celular e 0,70 na extração de conteúdo, enquanto a melhor ferramenta geométrica (pdfplumber) obteve apenas 0,43 e 0,19 nos mesmos níveis, respectivamente. Na detecção de páginas, o Gemini foi a única ferramenta a atingir recall perfeito (1,0), identificando todas as 125 páginas com tabelas do dataset.

Outra contribuição relevante foi a criação de um *dataset* anotado de tabelas financeiras brasileiras, composto por 15 documentos contendo 178 tabelas e 10.783 células, com ground truth estruturado incluindo informações de células mescladas. Este recurso pode ser utilizado em trabalhos futuros para benchmarking de novas ferramentas no domínio financeiro.

6.2 Limitações do Trabalho

O estudo apresenta limitações metodológicas que devem ser consideradas na interpretação dos resultados.

Primeiramente, o ground truth foi gerado de forma semi-automática utilizando o *IBM Docling*, o que introduz um potencial viés favorável a esta ferramenta. Por esta razão, o *Docling* foi excluído da comparação final, sendo utilizado exclusivamente como referência. Trabalhos futuros poderiam validar os resultados com anotação manual independente.

O Nível 2 (localização) apresentou F1 baixo para ferramentas geométricas (máximo de 0.16 para *Camelot*), embora o IoU médio das tabelas corretamente pareadas seja alto (>0.88). Isso indica que as ferramentas detectam poucas tabelas, mas quando detectam, a localização espacial é precisa. O *Google Gemini* não pôde ser avaliado neste nível por não fornecer coordenadas espaciais.

O dataset, embora representativo do domínio financeiro brasileiro, é limitado em tamanho (15 documentos, 178 tabelas). A generalização dos resultados para outros tipos de documentos financeiros ou outros domínios requer cautela.

Observou-se também que páginas em orientação landscape não foram adequadamente processadas pelo pipeline de ground truth, sendo excluídas da avaliação dos níveis 3 e 4. O *Google Gemini*, por não fornecer coordenadas espaciais (*bounding boxes*), não pôde ser avaliado no Nível 2 - limitação inerente à abordagem de modelos de linguagem.

6.3 Trabalhos Futuros

Os resultados obtidos sugerem diversas direções para pesquisas futuras.

Aprimoramento do Nível 2: Investigar estratégias para aumentar o recall na localização de tabelas, possivelmente combinando múltiplas ferramentas ou ajustando parâmetros de detecção..

Extração de Bounding Boxes via LLM: Investigar técnicas de prompt engineering ou modelos especializados que permitam ao *Gemini* fornecer coordenadas espaciais, viabilizando sua avaliação no Nível 2.

Abordagens Híbridas: Os resultados sugerem complementaridade entre ferramentas: enquanto o *Gemini* excele em estrutura e conteúdo, ferramentas geométricas podem oferecer localização espacial precisa. Uma direção promissora seria a criação de um processo utilizando o *LangChain* para misturar modelos especializados em cada tarefa da extração.

Expandir seleção de modelos de IA: Devido às limitações de orçamento, foi somente possível testar o *Gemini 2.5 Flash* por ser uma modelo relativamente barato. Com um investimento adequado, seria possível testar outras ferramentas como o *Claude 4.5 Sonnet* ou *ChatGPT 5*.

Dataset com avaliação humana: Nesse projeto, o *Docling* foi utilizado para definir o ground truth das tabelas, porém, com um prazo maior e equipe adequada, seria possível utilizar um Dataset extraído de forma semi-automatizada e incluir o *Docling* na avaliação.

Análise de Custo-Benefício: O campo de modelos multimodais evolui rapidamente. Trabalhos futuros poderiam avaliar modelos mais recentes e incluir análise de custo-benefício considerando tempo de processamento e custos de API, fatores relevantes para aplicações em escala.

7. Referências

ADHIKARI, Narayan; AGARWAL, Shradha. **A Comparative Study of PDF Parsing Tools Across Diverse Document Categories.** arXiv – Cornell University, 2024. Disponível em: arXiv:2410.09871. Acesso em 28/11/2025.

ADOBE. **Adobe Acrobat** - Explore all Acrobat features. Get to know the essential PDF app. Disponível em: https://www.adobe.com/acrobat/features.html. Acesso em 28/11/2025.

ADOBE SYSTEMS INC. **PDF Reference - Sixth Edition:** Adobe Portable Document Format Version 1.7. 2006. Disponível em: https://opensource.adobe.com/dc-acrobat-sdk-docs/pdfstandards/pdfreference1.7old.pdf. Acesso em 28/11/2025.

BODÊ, Ernesto Carlos. **Preservação de documentos digitais: o papel dos formatos de arquivo**. 2008. 153 f. Dissertação (Mestrado em Ciência da Informação) – Faculdade de Economia, Administração, Contabilidade, e Ciência de Informação, Universidade de Brasília, 2008. Disponível em: https://repositorio.unb.br/handle/10482/2034. Acesso em 28/11/2025.

BURDICK, Douglas *et al*. Table extraction and understanding for scientific and enterprise applications. **Proceedings of the VLDB Endowment**. v. 13, n. 12, aug. 2020, 1710 p. Disponível em: https://www.vldb.org/pvldb/vol13/p3433-burdick.pdf. Acesso em 28/11/2025.

DOUGLASS, Michael JJ. (Book Review) Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow (2nd edition by Aurélien Géron, O'Reilly Media, 2019). **Physical and Engineering Sciences in Medicine**. v. 43, 2020, p. 1135–1136. Disponível em: https://doi.org/10.1007/s13246-020-00913-z. Acesso em 28/11/2025.

GEMELLI, Andrea; VIVOLI, Emanuele; MARINAI Simone. **Graph neural networks and representation embedding for table extraction in PDF documents**. 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022.

GÖBEL, Max *et al*. A methodology for evaluating algorithms for table understanding in PDF documents. In: **Proceedings of the 2012 ACM symposium on Document Engineering**. 2012. p. 45-48.

KHUSRO, Shah; LATIF, Asima & ULLAH, Irfan. On methods and tools of table detection, extraction and annotation in PDF documents. **Journal of Information Science**, 41(1), 2015, p. 41-57. Disponível em: https://doi.org/10.1177/0165551514551903. Acesso em 28/11/2025.

LI, Minghao *et al.* Tablebank: Table benchmark for image-based table detection and recognition. **Proceedings of the Twelfth Language Resources and Evaluation Conference**. p. 1918-1925. Marseille, France. European Language Resources Association. 2020. Disponível em: https://aclanthology.org/2020.lrec-1.236/. Acesso em 28/11/2025.

MOHEMAD, Rosmayati *et al*. Automatic document structure analysis of structured PDF files. **International Journal of New Computer Architectures and their Applications** (IJNCAA), v. 1, n. 2, 2011, p. 404-411.

PALIWAL, Shubham *et al*. **Tablenet**: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images**.** 2019. International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.

PYMUPDF. Solving Common Issues With Table Detection and Extraction: How to navigate common problems with extracting tables from unstructured documents. **Medium**, 27 set. 2023. Disponível em: https://medium.com/@pymupdf/solving-common-issues-with-table-detection-and-extraction-4df5de2b8d88. Acesso em 28/11/2025.

ZANIBBI, Richard; BLOSTEIN, Dorothea; CORDY, James R. A survey of table recognition: Models, observations, transformations, and inferences. **International Journal on Document Analysis and Recognition** (IJDAR). v. 7, 2004, p. 1-16. Disponível em: https://doi.org/10.1007/s10032-004-0120-9. Acesso em 28/11/2025.

ZHANG, Mengshi *et al*. An integrated approach of deep learning and symbolic analysis for digital PDF table extraction. **25th International Conference on Pattern Tecognition** (ICPR, 2020). Milan (Italy), IEEE, 2021, p. 4062-4069. Disponível em: https://ieeexplore.ieee.org/abstract/document/9413069/. Acesso em 28/11/2025.