

Análise Evolutiva De Guidelines Para Agentes De Inteligência Artificial

Ana Luísa Messias F. Mendes

Depto. Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
analuisa.mendes@dcc.ufmg.br

André Hora

Depto. Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
andrehora@dcc.ufmg.br

Abstract—This project presents an evolutionary investigation of AI Agent guidelines in public GitHub repositories, comparing a dataset collected in 2025 with a new dataset collected in 2026. To this end, a tool developed in Python was employed, utilizing the GitHub REST API to identify guideline files through various naming patterns and to store their contents in a database for subsequent analysis. The comparison considers indicators such as the number of guidelines, their distribution across repositories, file sizes, naming conventions adopted, and the frequency of the main topics addressed. The results reveal a significant increase in the number of guidelines, indicating greater adoption of this type of documentation. However, it is observed that file naming remains weakly standardized, with several new conventions emerging across projects. These findings suggest an expanding ecosystem undergoing a maturation process, in which concern for agent documentation is increasing, although challenges related to standardization and the consolidation of best practices still persist.

Resumo— Este projeto apresenta uma investigação evolutiva das *guidelines* de Agentes de IA em repositórios públicos do GitHub, comparando uma base coletada em 2025 com uma nova base coletada em 2026. Para isso, foi utilizada uma ferramenta desenvolvida em Python que emprega a API Rest do GitHub para identificar arquivos de *guidelines* por meios de diferentes padrões de nomenclatura, armazenando seu conteúdo em uma base de dados para análise posterior. A comparação considera indicadores como quantidade de *guidelines*, distribuição por repositório, tamanho dos arquivos, nomenclaturas

adotadas e frequência dos principais temas abordados. Os resultados evidenciam um crescimento expressivo na quantidade de *guidelines*, indicando maior adoção desse tipo de documentação. Todavia, observa-se que a nomenclatura dos arquivos permanece pouco padronizada, surgindo diversas novas convenções entre os projetos. Esses achados sugerem um ecossistema em expansão e processo de amadurecimento, no qual cresce a preocupação com a documentação dos agentes, embora ainda persistam desafios relacionados à padronização e à consolidação de boas práticas.

Palavras-Chave: Agentes de Inteligência Artificial; Guidelines; Repositórios; API Rest; GitHub; Evolutiva

1. Introdução

Nos últimos anos, Agentes de Inteligência Artificial baseados em modelos de linguagem (LLM-based agents), como **GPT-4**, **GEMINI**, **Copilot**, têm se tornado cada vez mais presentes em aplicações práticas. Esses agentes são utilizados para automatizar fluxos de trabalho, auxiliar no desenvolvimento de software, realizar análise de dados, coordenar ferramentas externas e interagir com usuários de forma dinâmica. Dentre os exemplos mais utilizados, é possível destacar: **CLAUDE**, **LobeChat**, **OpenHands**, **GitHub Agent**, **Cursor**, **AutoGPT**. Contudo, esses Agentes estão se destacando e crescendo cada vez mais com sua capacidade cognitiva e entrega de resultados específicos em diferentes domínios apresentados por seus usuários. Na Engenharia de Software, eles representam uma evolução importante em relação aos modelos tradicionais de geração de código. [Hora e Hobbes 2026]

explicam que esses Agentes conseguem decompor objetivos complexos, adicionar ferramentas externas e refinar resultados com mínima intervenção humana. E, essa autonomia demonstra que eles não são sistemas estáticos, mas tecnologias que incorporam novas funcionalidades e níveis de complexidade com o tempo.

Com a rápida expansão desse ecossistema e com o aumento da complexidade dos Agentes, surgem novas arquiteturas e protocolos para o desenvolvimento dos mesmos. [Xu et. al. 2026] indica que os Agentes estão evoluindo rapidamente em capacidade, incorporando múltiplas ferramentas, mecanismos de coordenação e novos padrões de iteração. Logo, a documentação das guidelines (ou diretrizes) passa a ser essencial para garantir que os agentes funcionem de forma mais segura e previsível. Essas diretrizes definem o comportamento esperado, os limites operacionais e as práticas de integração com outras ferramentas, atuando na prática como artefatos de governança e engenharia de software. Além disso, o surgimento de novos protocolos para agentes reforça a necessidade de padronizar esse desenvolvimento, principalmente em projetos de código aberto e ambientes colaborativos.

A pesquisa foi dividida em duas etapas onde, na primeira parte, buscou-se entender o funcionamento dos agentes, bem como suas guidelines e como elas estavam sendo construídas. A partir do entendimento, foi criada uma base de dados contendo repositórios públicos de Agentes de Inteligência Artificial e seus respectivos arquivos de diretrizes. Com a base de dados, foi feita uma análise para observar a quantidade de guidelines encontradas, principais temas tratados nos arquivos, conteúdo encontrado e padronização de nomes dos arquivos. Os resultados mostraram uma baixa padronização na nomenclatura, grande variação na quantidade de guidelines por projeto e baixa presença de temas relacionados à segurança e limitações. A partir desses achados, surgiu a necessidade de compreender como esse cenário evoluiu com o amadurecimento do ecossistema que compõe a segunda etapa da pesquisa.

Dessa forma, este trabalho realiza uma análise evolutiva de guidelines de Agentes de IA em repositórios Open-Source, comparando os resultados obtidos em 2025 com uma nova coleta realizada em 2026. O objetivo é identificar mudanças na quantidade de arquivos, nomenclaturas adotadas, tamanho da documentação e temas abordados, buscando compreender indícios de maturidade, especialização e padronização no de-

envolvimento dos Agentes de IA. As principais contribuições deste projeto incluem:

- 1 - Ferramenta que usa a API do GitHub para percorrer repositórios selecionados e detectar arquivos candidatos por meio de um conjunto ampliado de padrões de nomenclatura.
- 2 - Ferramenta que analisa conteúdo a partir da base de dados permitindo visualizar sua distribuição, tamanho, frequência e presença de seções comuns.
- 3 - Resultados evolutivos a partir das análises.

Por fim, este relatório está organizado da seguinte forma:

- **Seção 2** apresenta um breve referencial teórico sobre a evolução dos agentes de IA e documentação técnica.
- **Seção 3** descreve as ferramentas desenvolvidas, bem como o processo de coleta e critérios utilizados para identificação das diretrizes.
- **Seção 4** apresenta os resultados obtidos e as análises realizadas.
- **Seção 5** discute conclusões e aponta direções para trabalhos futuros, incluindo a ampliação da base de dados e análises mais aprofundadas previstas para a continuação deste projeto

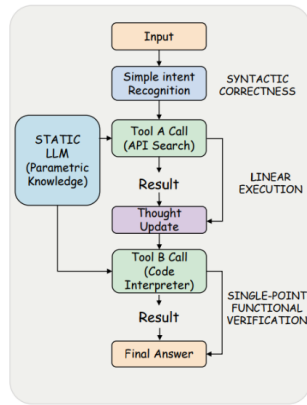
2. Referencial Teórico

O avanço recente dos Agentes baseados em Grandes Modelos de Linguagem (LLMs) tem impulsionado o desenvolvimento de sistemas inteligentes capazes de executar tarefas de forma autônoma. Por meio de mecanismos de planejamento e tomada de decisão, esses agentes interagem com diferentes ferramentas e adaptam seu comportamento a contextos variados, combinando raciocínio, memória, interação com o ambiente para resolver problemas mais complexos.

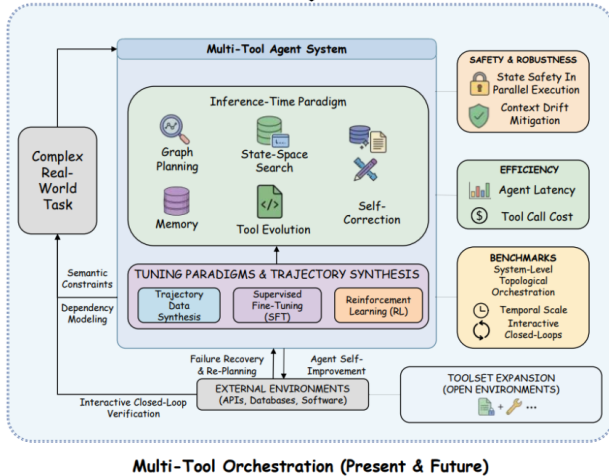
Apesar de recentes, o avanço desses sistemas tem ocorrido de forma acelerada, impulsionado pela crescente demanda por aplicações capazes de integrar múltiplas ferramentas e automatizar fluxos de trabalho. [Lin et. al, 2025] argumentam que essa evolução ocorre de maneira incremental com a incorporação contínua de novas capacidades e arquiteturas que tornam os agentes progressivamente mais sofisticados. Dessa forma, eles devem ser compreendidos como sistemas dinâmicos, cuja estrutura e comportamento evoluem ao longo do tempo.

2.1. Evolução Da Complexidade Dos Agentes

Uma das consequências desse processo evolutivo é o aumento da complexidade das arquiteturas utilizadas. [Xu et. al, 2026] mostram que os Agentes passaram de modelos capazes de realizar chamadas isoladas para sistemas que coordenam múltiplas ferramentas de maneira integrada, executando tarefas compostas e adaptando estratégias conforme o contexto que lhes é apresentado. Esse cenário reforça a importância da documentação técnica para orientar o comportamento esperado dos agentes e garantir consistência durante sua execução.



Linear Reasoning (Past works)



Multi-Tool Orchestration (Present & Future)

Figure 1. Mudança de Paradigma no Uso de Ferramentas LLM: De Chamada de Uma Única Ferramenta para Orquestração de Múltiplas Ferramentas [Xu et. al. 2026]

2.2. Documentação Dos Agentes

Nesse contexto, as *guidelines* tornam-se elementos fundamentais do desenvolvimento desses sistemas. Esses arquivos documentam as regras de comportamento, restrições, estratégias, ferramentas e demais instruções necessárias para orientar a atuação dos agentes. Em projetos de código aberto, essa documentação facilita a colaboração entre desenvolvedores e contribui para a manutenção da qualidade do software. [Falcucci et. al, 2026] demonstram que diretrizes bem definidas favorecem a padronização dos processos de desenvolvimento, reduzem inconsistências e auxiliam na comunicação entre os colaboradores de um projeto.

Além da documentação, o amadurecimento do ecossistema tem impulsionado iniciativas voltadas à padronização da comunicação entre agentes. [Yang et. al, 2025] destacam que protocolos específicos são essenciais para garantir interoperabilidade, segurança e consistência entre diferentes plataformas e arquiteturas. No contexto desta pesquisa, a adoção crescente de arquivos como **AGENTS.md**, **CLAUDE.md** e outros formatos de diretrizes representa um indício desse movimento em direção à padronização, ainda que diferentes projetos adotem convenções próprias.

Em conjunto, esses estudos mostram que a análise evolutiva das *guidelines* torna-se uma estratégia relevante para compreender como os Agentes estão se desenvolvendo diante do rápido crescimento da tecnologia na incorporação desses sistemas com o mundo real e com o aumento da complexidade em que eles estão trabalhando.

3. Metodologia E Desenvolvimento

A metodologia deste trabalho foi desenvolvida em duas etapas complementares. Na primeira, foi construída uma ferramenta capaz de minerar repositórios públicos do GitHub, identificar arquivos de *guidelines* e armazenar essas informações em um Banco de Dados estruturado. Na segunda etapa, essa mesma infraestrutura foi reutilizada para realizar uma análise evolutiva, permitindo comparar a evolução das *guidelines* dos mesmos projetos ao longo do tempo.

3.1. Arquitetura Geral Do Sistema

A solução foi organizada como um pipeline composto por quatro etapas principais:

- **1 - Entrada de Repositórios:** Construção manual de uma lista de repositórios públicos de Agentes de IA hospedados no GitHub. Foram considerados projetos consolidados com histórico de desenvolvimento ativo e mais de 50 estrelas.
- **2 - Coleta automática de metadados:** Utilização da API Rest do GitHub para obter informações dos repositórios, como nome, proprietário, linguagem principal, quantidade de forks, data de criação e última atualização
- **3 - Identificação das Guidelines:** Varredura recursiva da árvore de diretórios dos repositórios para localizar arquivos candidatos a guidelines por meio de um conjunto de padrões previamente definidos.
- **4 - Construção do dataset de análise:** Armazenamento das informações coletadas em um banco de dados SQLite para posterior realização das análises estatísticas e comparativas.

Essa arquitetura tornou possível reproduzir automaticamente toda a coleta sempre que necessário, permitindo acompanhar a evolução dos mesmos projetos ao longo do tempo.

3.2. Ferramenta De Coleta De Repositórios

A primeira ferramenta desenvolvida tem como objetivo registrar e armazenar repositórios públicos do **GitHub**, junto com seus principais metadados em uma base **SQLite**. A ferramenta foi desenvolvida em linguagem **Python** e utiliza a biblioteca 'pathlib' que oferece uma forma moderna e orientada a objetos para manipular caminhos de arquivos e diretórios.

A ferramenta funciona da seguinte forma:

- Recebe uma lista de URLs de repositórios fornecida manualmente ou via arquivo .txt (para este estudo foi inserida via arquivo .txt).
- Extrai automaticamente o identificador do repositório (owner e nome).
- Consulta a **GitHub REST API v3** (entrada via terminal) utilizando um token de autenticação.
- Obtém metadados importantes como: número de estrelas, número de forks, número de colaboradores, tamanho dos repositórios, data do commit mais recente e árvore de diretórios (via endpoint 'git/trees').

Todos os dados são armazenados na tabela '**repos**' de um banco **SQLite**, permitindo consultas rápidas e independência de serviços externos.

```

1 def main():
2     g = Github(GITHUB_TOKEN)
3     conn = sqlite3.connect(DB_PATH)
4
5     with open(REPOS_FILE, "r", encoding="utf-8")
6         as f:
7         urls=[line.strip() for line in f if line.
8             strip()]
9
10    for url in tqdm(urls, desc="Inserindo"):
11        repo_name =
12            "/".join(url.rstrip("/").split("/")[-2:])
13        try:
14            repo = g.get_repo(repo_name)
15            insert_repo(repo, conn)
16        except Exception as e:
17            print(f"Erro ao inserir {repo_name}: {e}")
18
19    conn.close()
20    print("Processo concluído.")

```

Figure 2. Algoritmo de coleta e armazenamento dos Repositórios

Ao todo, foram coletados manualmente **333** links de repositórios que tinham "AI AGENT" no seu conteúdo no GitHub. Esses links foram inseridos em um arquivo .txt que foi usado no programa.

3.3. Ferramenta De Identificação E Extração De Guidelines

A segunda ferramenta desenvolvida para este trabalho é responsável por localizar arquivos de guidelines relacionados a agentes de IA, extrair seu conteúdo e organizá-lo para análise. Também foi desenvolvida na linguagem **Python** e também utiliza a biblioteca 'pathlib' para navegar nos repositórios de maneira mais fácil e rápida para identificar as guidelines.

Um desafio encontrado durante a execução deste trabalho foi que, por ser um assunto ainda recente, não existe um padrão específico para nomear as guidelines dos agentes de IA. A maioria dos arquivos são no formato **Markdown (.md)**, porém, cada projeto pode dar o nome que quiser para seu arquivo, sendo uma decisão final do desenvolvedor. Portanto, foi necessário criar um conjunto extenso de padrões para capturar os diferentes formatos utilizados pelos desenvolvedores. Foram definidos **45** variações de identificação dos arquivos contemplando diferentes convenções encontradas na literatura e nos projetos analisados. Dentre os exemplos estão:

- AGENT.md; AGENTS.md; CLAUDE.md; GEMINI.md; CURSOR.md; README_AI.md
- agent_instructions.md, ai_agent_instructions.md, copilot_instructions.md, gemini_instruicions.md, cursor_instructions.md, ai_instructions.md
- agent_guidelines.md, assistant_guidelines.md, ai_agent_guideline.md, llm_guidelines.md
- agent_prompt.md, ai_agent_prompt.md, ai_prompt.md
- arquivos armazenados em pastas como **.claude/** ou **.agents/**

Cada repositório tem sua árvore percorrida recursivamente, e todo arquivo *.md* é comparado aos padrões definidos. Quando há correspondência, o arquivo é baixado via **API Rest GitHub**, decodificado (**Base64** → **texto**), armazenado na tabela **'guidelines'** com: ID-repositório, caminho do arquivo, tamanho e conteúdo textual completo.

O banco utiliza uma chave única (repo-id, file-path) para evitar duplicação mesmo se o script for executado várias vezes.

```

1 def collect_guidelines(token, db_path):
2     # Conecta ao GitHub e ao BD
3     g = Github(token)
4     conn = sqlite3.connect(db_path)
5     repos = fetch_repos(conn)
6     # Busca repositórios pre-existent
7
8     for repo_id, full_name in repos:
9         try:
10            repo = g.get_repo(full_name)
11            # A busca recursiva (DFS) pelos arquivos
12            files = repo.search_all_files()
13
14            for file in files:
15                if is_guideline(file.path):
16                    content=
17                    file.decoded_content.decode("utf-8")
18                    insert_guideline(repo_id, file,
19                                    content, conn)
20
21            except Exception as e:
22                print(f"Erro: {e}")
23                continue
24
25 conn.close()
```

Figure 3. Algoritmo de busca das Guidelines a partir dos repositórios guardados no banco

3.4. Atualização Para Análise Evolutiva

A segunda etapa do trabalho foi realizar a atualização das bases estudadas para garantir a análise evolutiva. A base construída em 2025 foi armazenada

como referencial histórico. A partir da mesma lista de repositórios usados na coleta inicial, um novo banco de dados foi criado (*agents_analysis_2026.db*) e todas as ferramentas foram executadas novamente. Na execução foi mantida a mesma quantidade de informações armazenadas, bem como a mesma lista de nomes das guidelines para avaliar evolução também em relação à padronização de nomenclatura.

Essa estratégia permitiu capturar a versão mais recente das guidelines dos mesmos projetos, mantendo independentes as duas bases de dados. Dessa forma, foi possível fazer a comparação quantitativa e qualitativa da evolução das guidelines.

A execução total para capturar as guidelines dos projetos foi de **14 Horas e 35 Minutos**. Um aumento de 5 Horas e 15 Minutos, em relação à execução feita em 2025, que foi de 9 Horas e 20 Minutos na primeira análise.

3.5. Métricas Analisadas

Com a conclusão das duas bases de dados foram realizadas consultas SQL e análises sobre os arquivos coletados. As principais métricas consideradas neste trabalho para análise foram:

- Crescimento da quantidade total de guidelines;
- Distribuição de guidelines por repositório;
- Evolução dos repositórios com maior quantidade de documentação;
- Tamanho dos arquivos;
- Frequência dos principais temas presentes nas guidelines;
- Evolução das nomenclaturas adotadas para as guidelines

Essas métricas permitiram caracterizar a evolução da documentação dos agentes e identificar indícios de amadurecimento do ecossistema ao longo do período analisado.

4. Resultados

Ao todo, foram coletados **333 repositórios públicos** do GitHub relacionados a agentes de IA ou aplicações com componentes baseados em agentes. Os repositórios foram buscados e inseridos de forma manual na lista de repositórios (*.txt*) a fim de fazer uma busca mais refinada. A comparação foi realizada entre a base construída na primeira etapa da pesquisa, em 2025, e uma nova coleta executada em 2026 utilizando a mesma ferramenta, os mesmos repositórios

e um conjunto ampliado de padrões de identificação de arquivos de guidelines. Essa estratégia permitiu avaliar como a documentação dos projetos evoluiu ao longo do tempo, reduzindo interferências causadas por mudanças na amostra analisada.

As análises contemplam diferentes aspectos da evolução das guidelines, incluindo o crescimento da quantidade de arquivos, sua distribuição entre os repositórios, a evolução dos projetos com maior volume de documentação, o tamanho dos arquivos, os principais temas abordados e as nomenclaturas adotadas pelos desenvolvedores. Durante as análises, foram identificados alguns pontos de atenção que fornecem evidências de grau de maturidade dos projetos, tendências de especialização e possíveis desafios futuros em relação à padronização de nomenclatura.

4.1. Evolução Quantitativa de Guidelines

A primeira análise buscou verificar se houve crescimento na adoção de arquivos de *guidelines* entre os mesmos repositórios analisados na primeira etapa da pesquisa.

Na coleta realizada em 2025 foram identificados **352** guidelines, distribuídas entre 98 dos 333 repositórios. Uma coleta inicial foi feita em Fevereiro de 2026 para verificar a viabilidade do estudo e encontrou **1274** guidelines. Por fim, a última coleta realizada em Junho/2026 encontrou **1329** arquivos de guidelines distintos em 115 repositórios, representando um crescimento de aproximadamente **278%** em menos de 1 ano nos arquivos de guidelines.

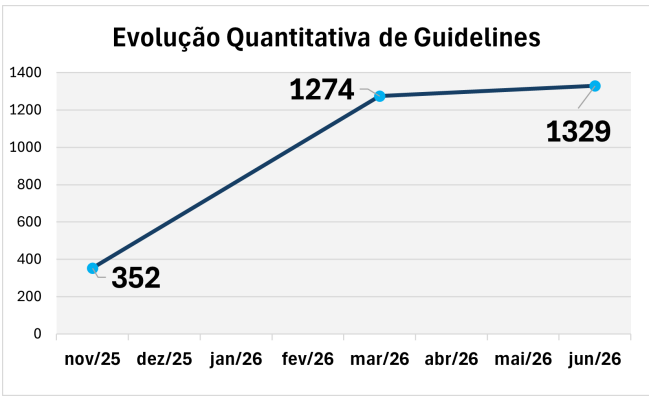


Figure 4. Gráfico Evolução Quantidade Guidelines

Esse crescimento evidencia que a documentação voltada para Agentes de IA passou a ocupar um papel significativamente importante no desenvolvimento dos

projetos. A expansão da quantidade de guidelines sugere que os agentes estão se tornando mais complexos, exigindo mais documentos específicos para diferentes tarefas, ferramentas e comportamentos, na tentativa de refinar ainda mais os agentes. Além disso, o resultado está alinhado com a evolução descrita por **Lin et. al. (2025)**, segundo os quais os agentes modernos incorporam continuamente novas capacidades e componentes especializados, aumentando naturalmente a necessidade de documentação.

Uma outra observação é que esse aumento, em um curto espaço de tempo, mostra que os agentes estão evoluindo muito rapidamente com o avanço da tecnologia e a popularização do uso dos agentes. Usuários e aplicações estão exigindo sistemas cada vez mais robustos, com mais ferramentas, mais seguros e que tenham capacidade de resolver problemas maiores e mais complexos. Dessa forma, os desenvolvedores precisam inserir cada vez mais instruções para os Agentes para conseguir atender a alta demanda de seus usuários.

4.2. Distribuição De Guidelines Por Repositório

Além do crescimento global, foi analisada a distribuição da quantidade de guidelines entre os diferentes repositórios.

Os resultados mostram que a maioria dos projetos ainda possui poucas guidelines, enquanto um número reduzido de repositórios concentra grande quantidade de arquivos de documentação. Isso pode indicar que alguns projetos estão atingindo níveis elevados de especialização documental.



Figure 5. Gráfico Evolução Guidelines Por Repositório

Entre os projetos analisados, destacam-se o **Rufflo Multiagent for CLAUDE and CODEX** com 521 guidelines, e o **Microsoft AI for Beginners** com 285 guidelines, valores significativamente superiores aos observados na maioria dos repositórios. Esse elevado número de arquivos evidencia a complexidade desses projetos, que envolvem múltiplos agentes, ferramentas, fluxos de execução e componentes especializados. [Xu et. al. 2026] argumentam que a evolução dos Agentes de IA tem sido marcada pela transição de arquiteturas simples para sistemas capazes de coordenar múltiplas ferramentas e módulos de forma integrada (Conforme Fig. 1), tornando necessária uma documentação cada vez mais detalhada para orientar seu funcionamento.

Outro aspecto observado é que o crescimento da quantidade de guidelines não ocorre apenas pelo aumento de conteúdo, mas também pela adoção de uma estratégia de modularização da documentação. Em vez de concentrar todas as instruções em um único arquivo, projetos mais maduros distribuem guidelines em diversos documentos especializados, dedicados a aspectos como segurança, testes, ferramentas, prompts ou componentes específicos do sistema. Essa organização reduz a complexidade de manutenção, facilita a evolução independente de cada módulo e melhora a reutilização das guidelines.

Portanto, a distribuição observada sugere que a quantidade de guidelines pode ser utilizada como um indicador indireto da complexidade arquitetural dos Agentes. Embora projetos menores continuem adotando documentação centralizada, é observado que sistemas mais sofisticados tendem a apresentar uma estrutura documental mais extensa e fragmentada, refletindo a evolução do ecossistema de Agentes de IA e o aumento da especialização de seus componentes. Mas, isso ainda varia bastante entre os desenvolvedores.

4.3. Evolução Do Tamanho Dos Arquivos

Além da quantidade de documentos, foi feita uma análise do tamanho dos arquivos de guidelines. Observou-se um aumento tanto da quantidade quanto do tamanho médio dos arquivos entre as duas coletas. Na base de 2025 a média foi de **5KB** por guideline, enquanto na base de 2026, a média aumentou para **11KB** por arquivo. Esse resultado indica que a documentação passou a conter instruções mais detalhadas, descrições maiores e mais completas, regras mais específicas e maior quantidade de exemplos.

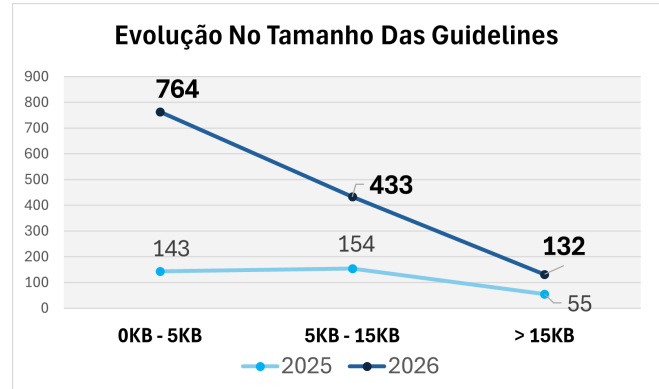


Figure 6. Gráfico Evolução Tamanho das Guidelines

O comportamento observado no gráfico indica que a evolução da documentação ocorreu em duas frentes complementares. Por um lado, alguns projetos passaram a produzir documentos mais completos, contendo descrições detalhadas de ferramentas, regras de funcionamento e instruções específicas para diferentes componentes do sistema. Por outro, observa-se uma quantidade significativamente maior de arquivos menores, sugerindo que muitos desenvolvedores passaram a fragmentar a documentação em diretrizes especializadas, cada uma destinada a um aspecto do agente, como segurança, testes, prompts ou integração com ferramentas.

Esse resultado reforça a tendência observada na seção 3.2. Em vez de concentrar todas as instruções em um único documento extenso, projetos mais maduros tendem a modularizar suas guidelines em diversos arquivos menores e independentes. Essa estratégia facilita a manutenção da documentação, melhora sua reutilização e acompanha o aumento da complexidade arquitetural dos agentes modernos, cujos componentes passaram a ser desenvolvidos e evoluídos de forma cada vez mais especializada.

4.4. Evolução Dos Temas Presentes Nas Guidelines

A análise de conteúdo teve o objetivo de identificar a presença de temas recorrentes como: *Segurança (Safety)*, *Proteção (Security)*, *Ferramentas (Tools)*, *Instruções (Instructions)*, *Comandos (Prompts)*, *Comportamento (Behavior)*, *Testes (Tests)*, *Compilação (Compile)*, *Limitações (Limitations)*. A detecção foi realizada por meio da identificação de seções e palavras-chave presentes no conteúdo dos arquivos.

Os resultados mostram que **Testes (46.05%)** e **Ferramentas (36.95%)** permaneceram como os temas mais frequentes nas guidelines, indicando que a documentação continua fortemente voltada para aspectos de desenvolvimento, validação e integração dos agentes com ferramentas externas. Outra observação é que, em relação à coleta de 2025, todos os temas tiveram crescimento, o que indica um reflexo direto da expansão da base de documentos analisados.

TABLE 1. COMPARAÇÃO DOS PRINCIPAIS TEMAS IDENTIFICADOS NAS GUIDELINES EM 2025 E 2026

	2025		2026	
	Ocorrências	%	Ocorrências	%
Safety	31	8.81%	75	5.64%
Security	58	16.48%	283	21.29%
Tools	142	40.34%	491	36.95%
Limitations	22	6.25%	34	2.56%
Instructions	151	42.90%	268	20.17%
Behavior	58	16.48%	185	13.92%
Prompt	60	17.05%	173	13.02%
Tests	255	72.44%	612	46.05%
Compile	20	5.68%	42	3.16%

Fazendo uma comparação gráfica:

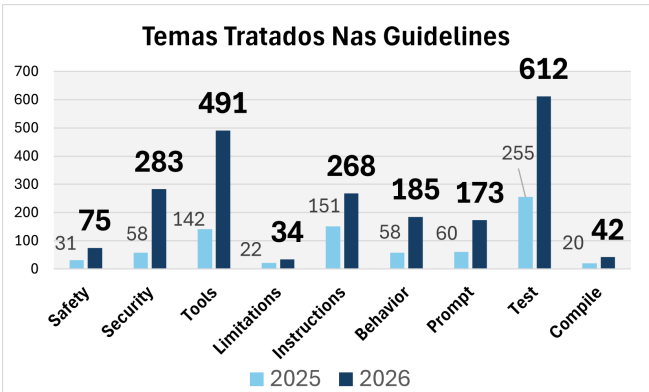


Figure 7. Gráfico Evolução Temas Guidelines

Ao analisar a frequência relativa dos temas, observa-se uma mudança no perfil da documentação. Enquanto **Security** aumentou a sua participação (de 16.48% para 21.29%), temas como **Instructions**, **Tests**, **Prompt**, **Behavior**, **Safety** e **Limitations** apresentam redução percentual. Esse comportamento sugere que o crescimento das guidelines ocorreu de forma mais diversificada, incorporando novos tipos de documentação e distribuindo as informações em arquivos mais especializados, em vez de concentrar

grandes conjuntos de instruções em um único documento.

Outro resultado relevante é que, embora os temas relacionados à segurança também tenham apresentado crescimento absoluto, **Safety** continua presente em apenas **5.64%** das guidelines analisadas, enquanto **Limitations** aparece com apenas **2.56%**. Esses resultados indicam que aspectos técnicos e operacionais ainda recebem maior atenção dos desenvolvedores do que tópicos relacionados à governança, restrições de uso e mecanismos de segurança, reforçando que essas dimensões permanecem como oportunidades para o amadurecimento da documentação dos Agentes de IA.

4.5. Evolução Das Nomenclaturas Das Guidelines

Por fim, foi analisada a evolução das nomenclaturas utilizadas para identificar arquivos de *guidelines* nos repositórios avaliados. Para essa análise, os arquivos foram agrupados de acordo com os principais padrões de nomenclatura encontrados durante a mineração dos repositórios. A busca foi feita através de uma ferramenta em Python que, com base nos nomes definidos na busca de guidelines, classifica os arquivos e faz a contagem de quantos com aquele nome existem. Ao final, ela retorna todos os arquivos, bem como o caminho de armazenamento deles que estão classificados em "OUTROS".

Em 2025 foram encontrados os principais nomes de guidelines:

TABLE 2. PRINCIPAIS NOMENCLATURAS BD_2025

Nomenclatura	Quantidade
AGENTS.md	118
OUTROS	116
CLAUDE.md	58
CLAUDE_INSTRUCTIONS.md	11
AGENT.md	8
GEMINI.md	8
AI_INSTRUCTIONS.md	7
AGENTS_INSTRUCTIONS.md	5
GEMINI_INSTRUCTIONS.md	5
AGENT_INSTRUCTIONS.md	4

Já em 2026, o resultado foi bem diferente:

TABLE 3. PRINCIPAIS NOMENCLATURAS BD_2026

Nomenclatura	Quantidade
OUTROS	841
AGENTS.md	208
SKILL.md	135
CLAUDE.md	86
AGENT.md	20
GEMINI.md	11
CLAUDE_INSTRUCTIONS.md	8
AI_INSTRUCTIONS.md	7
GEMINI_INSTRUCTIONS.md	5
AGENTS_INSTRUCTIONS.md	5

Embora AGENTS.md permaneça como a nomenclatura mais difundida entre os formatos padronizados, com 208 ocorrências, observou-se que a categoria Outros tornou-se predominante, totalizando 841 arquivos, quase quatro vezes superior ao principal padrão identificado. Entre os demais nomes recorrentes destacam-se **SKILL.md** (135 ocorrências), **CLAUDE.md** (86), **AGENT.md** (20) e **GEMINI.md** (11).

Uma inspeção manual dos arquivos classificados como "OUTROS" revelou que essa categoria não representa documentação irrelevante, mas sim uma ampla diversidade de arquivos especializados, distribuídos em estruturas como .claude/, .agents/, .github/ e .roo/. Foram identificados documentos voltados para comandos específicos (commands), agentes especializados (subagents), habilidades (skills), fluxos de trabalho (workflows), regras (rules), templates e documentos auxiliares de configuração.

Esse resultado sugere uma mudança importante no ecossistema de agentes de IA. Em vez de concentrar todas as instruções em um único arquivo, como proposto inicialmente pelo padrão AGENTS.md, projetos mais recentes passaram a modularizar suas diretrizes em diversos arquivos menores e especializados. Essa estratégia facilita a manutenção da documentação, aumenta sua reutilização entre diferentes agentes e permite que cada componente possua instruções específicas para sua função.

Ao mesmo tempo, a grande diversidade de nomenclaturas mostra que ainda não existe um consenso sobre como organizar a documentação de agentes. Apesar das iniciativas recentes de padronização propostas pela comunidade, os desenvolvedores continuam adotando convenções próprias, adaptadas às arquiteturas de seus projetos. Essa falta de uniformidade dificulta a mineração automática de guidelines, revela

um crescimento desenfreado, reduz a interoperabilidade entre ferramentas e reforça a necessidade de protocolos mais consolidados para documentação de agentes de Inteligência Artificial.

5. Conclusão E Trabalhos Futuros

O objetivo deste trabalho foi alcançado ao desenvolver uma ferramenta capaz de identificar, coletar e analisar automaticamente arquivos de guidelines em projetos públicos de Agentes de Inteligência Artificial, além de investigar a evolução desse ecossistema entre 2025 e 2026. A metodologia proposta mostrou-se eficaz para a construção de uma base de dados reprodutível e para a extração de métricas relacionadas à documentação desses agentes.

Os resultados evidenciaram um crescimento expressivo tanto na quantidade de guidelines quanto na diversidade de nomenclaturas e na complexidade da documentação presente nos repositórios analisados. Observou-se também uma tendência de modularização das diretrizes, com projetos mais maduros distribuindo instruções em múltiplos arquivos especializados. Apesar desse amadurecimento, a análise revelou que ainda não existe um padrão consolidado para organização e nomenclatura das guidelines, dificultando a mineração automática, a interoperabilidade entre ferramentas e a adoção de boas práticas pela comunidade.

Outro aspecto relevante é que os temas predominantes continuam relacionados à utilização de ferramentas, testes e instruções operacionais, enquanto assuntos como segurança, limitações e governança permanecem proporcionalmente menos frequentes. Esse resultado sugere que o desenvolvimento atual dos agentes tem priorizado aspectos funcionais em detrimento de mecanismos explícitos de controle e confiabilidade.

Os resultados também demonstram que iniciativas recentes de padronização, como a especificação AGENTS.md proposta pela OpenAI, representam um avanço importante para o ecossistema. Embora o formato AGENTS.md já apareça como a nomenclatura mais frequente entre os arquivos identificados, a grande quantidade de nomes personalizados observada nesta pesquisa indica que essa padronização ainda está em processo de adoção. Esse cenário reforça a necessidade de protocolos mais consolidados para facilitar a documentação, manutenção e evolução de agentes de IA.

Como trabalhos futuros, pretende-se expandir a base de repositórios analisados, realizar análises semânticas do conteúdo das guidelines, investigar sua evolução temporal em períodos mais longos e estudar a relação entre a qualidade da documentação e características dos projetos, como popularidade, arquitetura e desempenho dos agentes.

References

- [1] A. Hora and R. Robbes, “Are Coding Agents Generating Over-Mocked Tests? An Empirical Study,” *MSR 2026*, 2026. Available: <https://doi.org/10.48550/arXiv.2602.00409>
- [2] B. Falcucci, F. Gomide, and A. Hora, “What Do Contribution Guidelines Say About Software Testing?,” *MSR 2025*, 2025. Available: <https://doi.org/10.48550/arXiv.2602.02966>
- [3] M. Lin, H. Lu, Z. Shi, B. He, R. Mao, Z. Zhang, Z. Wu, X. Tang, H. Liu, Z. Dai, X. Zhang, S. Wang, B. Dumoulin, and J. Pei, “Position: Agentic Evolution is the Path to Evolving LLMs,” 2025.
- [4] H. Xu, C. Li, X. Ma, X. Ou, Z. Zhang, T. He, X. Liu, Z. Wang, J. Liang, Z. Chu, R. Liu, R. Mu, D. Tu, M. Liu, and B. Qin, “The Evolution of Tool Use in LLM Agents: From Single-Tool Call to Multi-Tool Orchestration,” *CS-SE*, 2026. Available: <https://doi.org/10.48550/arXiv.2603.22862>
- [5] Y. Yang, H. Chai, Y. Song, S. Qi, M. Wen, N. Li, J. Liao, H. Hu, J. Lin, G. Chang, W. Liu, Y. Wen, Y. Yu, and W. Zhang, “A Survey of AI Agent Protocols,” 2025.
- [6] K. Hayawi and S. Shahriar, *AI Agents from Copilots to Coworkers: Historical Context, Challenges, Limitations, Implications, and Practical Guidelines*. Abu Dhabi: Zayed University; Dubai: AI CEO FZ-LLC, 2024.
- [7] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang *et al.*, “A Survey on Large Language Model Based Autonomous Agents,” *Frontiers of Computer Science*, vol. 18, Article ID 186345, 2024. Available: <https://doi.org/10.1007/s11704-024-40231-1>
- [8] T. Xie and A. E. Hassan, “Mining Software Engineering Data,” in *Proc. 31st Int. Conf. Software Engineering (ICSE 2009) Tutorials*, Vancouver, Canada, 2009.
- [9] L. Luo, T. Zhang, Y. Zhou *et al.*, “Taxonomy and Evaluation of Large Language Model Agents: Architectures, Applications, and Challenges,” *Journal of Artificial Intelligence Research*, vol. 78, pp. 145–182, 2025.
- [10] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “ReAct: Synergizing Reasoning and Acting in Language Models,” in *International Conference on Learning Representations (ICLR)*, 2023.
- [11] H. P. Zou, W.-C. Huang, Y. Wu *et al.*, “LLM-Based Human-Agent Collaboration and Interaction Systems: A Survey,” *arXiv preprint arXiv:2505.00753*, 2025.
- [12] M. Kraus, S. Zepf, R. Westhäußer, I. Feustel, N. Zargham, I. Aslan, J. Edwards, S. Mayer, D. Kontogiorgos, N. Wagner, and E. André, “BEHAVE AI: Best Practices and Guidelines for Human-Centric Design and Evaluation of Proactive AI Agents,” in *Companion Proc. 30th Int. Conf. Intelligent User Interfaces (IUI '25 Companion)*, ACM, New York, NY, USA, pp. 175–178, 2025. Available: <https://doi.org/10.1145/3708557.3716155>
- [13] N. Watson, A. Hessami, F. Fassihi, S. Abbasi, H. Jahankhani, S. El-Deeb, I. Caetano, S. David, M. Newman, S. Moriarty, M. Cuhadaroglu, V. Tashchev, Z. Murahwi, R. Pihlakas, K. Crockett, S. Essafi, A. Hessami, and L. Dajani, “Guidelines for Agentic AI Safety, Volume 1: Agentic AI Safety Experts Focus Group – Sept. 2024,” Universal Ethics Community of Practice Working Group, Working Paper, 2024.
- [14] Amazon Web Services, “O que são Agentes de IA?,” Available: <https://aws.amazon.com/pt/what-is/ai-agents/>. Accessed: Sep. 6, 2025.
- [15] Google Cloud, “O que são Agentes de IA? Definição, exemplos e tipos,” Available: <https://cloud.google.com/discover/what-are-ai-agents>. Accessed: Sep. 5, 2025.
- [16] OpenAI, “AGENTS.MD,” Available: <https://agents.md/>.