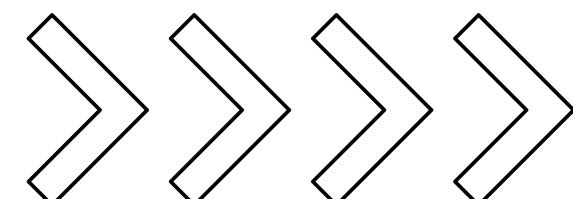


TEXT-TO-SQL COM SISTEMAS MULTIAGENTES

Artur Magalhaes dos Santos
Daniel Ferreira Schulz
Guilherme Narciso Lee
Pedro Rezende Mendonça

10297734
15610137
12543203
8961491



Disciplina: Modelagem de Banco de Dados
Professora: Kelly Rosa Braghetto

Índice

- 03** Introdução
- 04** Large Language Models (LLMs)
- 05** Sistemas Multiagentes
- 06** Text-to-SQL

- 07** Trabalhos Relacionados
- 09** Estudo de Caso
- 13** Resultados
- 14** Conclusão e Trabalhos Futuros

Introdução

Multiagentes vs. Agente Único em Text-to-SQL

- **Contexto:** Democratização de dados exige superação de barreiras técnicas do SQL.
- **Questão Central:** Arquiteturas multiagente superam métodos de agente único em Text-to-SQL?
- **Metodologia:** Experimento controlado (inspirado no benchmark Spider) comparando:
 - Abordagem de agente único
 - Abordagem multiagente (proposta/revisão/refinamento)
- **Métricas:**
 - Exact Match
 - Execution Accuracy

Large Language Models (LLMs)

- **Definição:**
 - Modelos de IA capazes de compreender e gerar linguagem natural, baseados em distribuição probabilística para previsão de sequências textuais.
 - **Arquitetura:**
 - Baseada em Transformers (Vaswani et al.).
 - Mecanismo de atenção para paralelização e tratamento de dependências de longo alcance.
 - **Escalabilidade:**
 - Treinados com bilhões de parâmetros em grandes volumes de dados.
 - Habilitam aprendizado em contexto (in-context learning): adaptação a tarefas via exemplos na entrada.

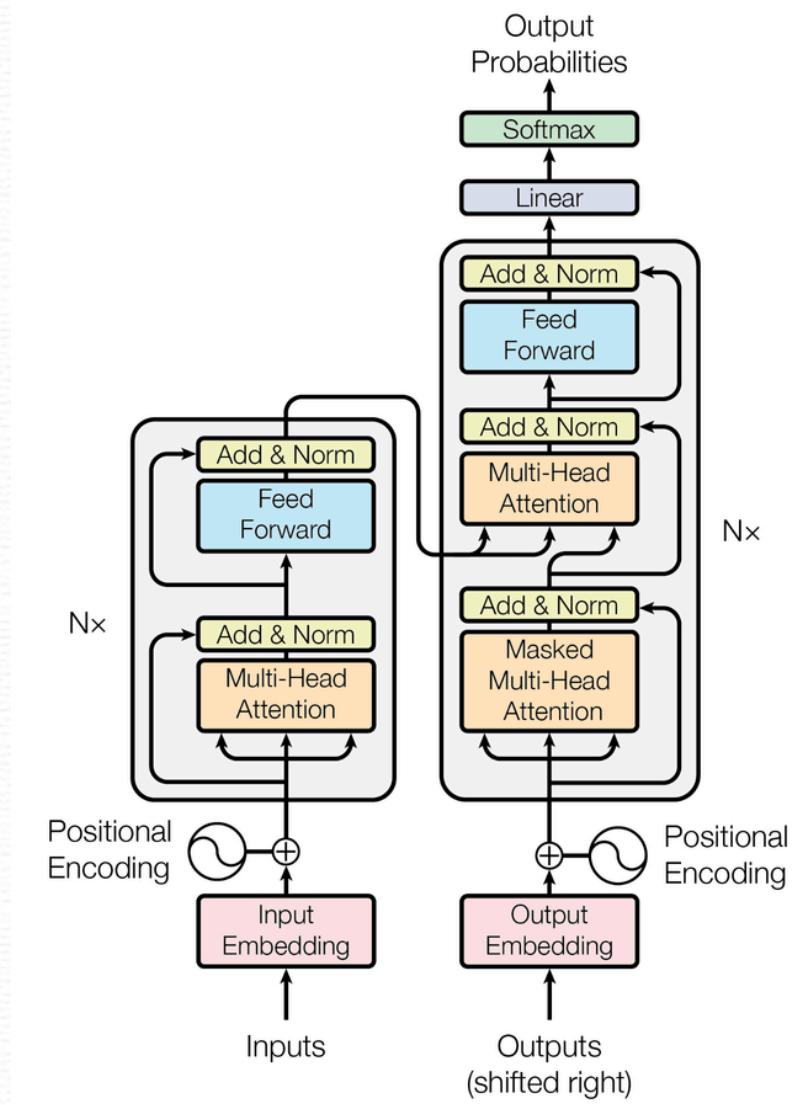


Figura 1: Arquitetura Transformer (Vaswani et al.)

Sistemas Multiagentes

Arquitetura de divisão de funcionalidades por agentes

- Agente: Entidade autônoma que percebe, decide e age com base em uma função para atingir determinado objetivo.
- Sistema Multiagente: Conjunto de agentes que funcionam entre si cooperando ou competindo para resolver problemas complexos e multicamadas.
- Integração com LLMs:
 - Agentes se comunicam via **propostas, críticas e refinamentos**.
- Tipos de Interação:
 - Cooperativa
 - Adversarial
 - Mista

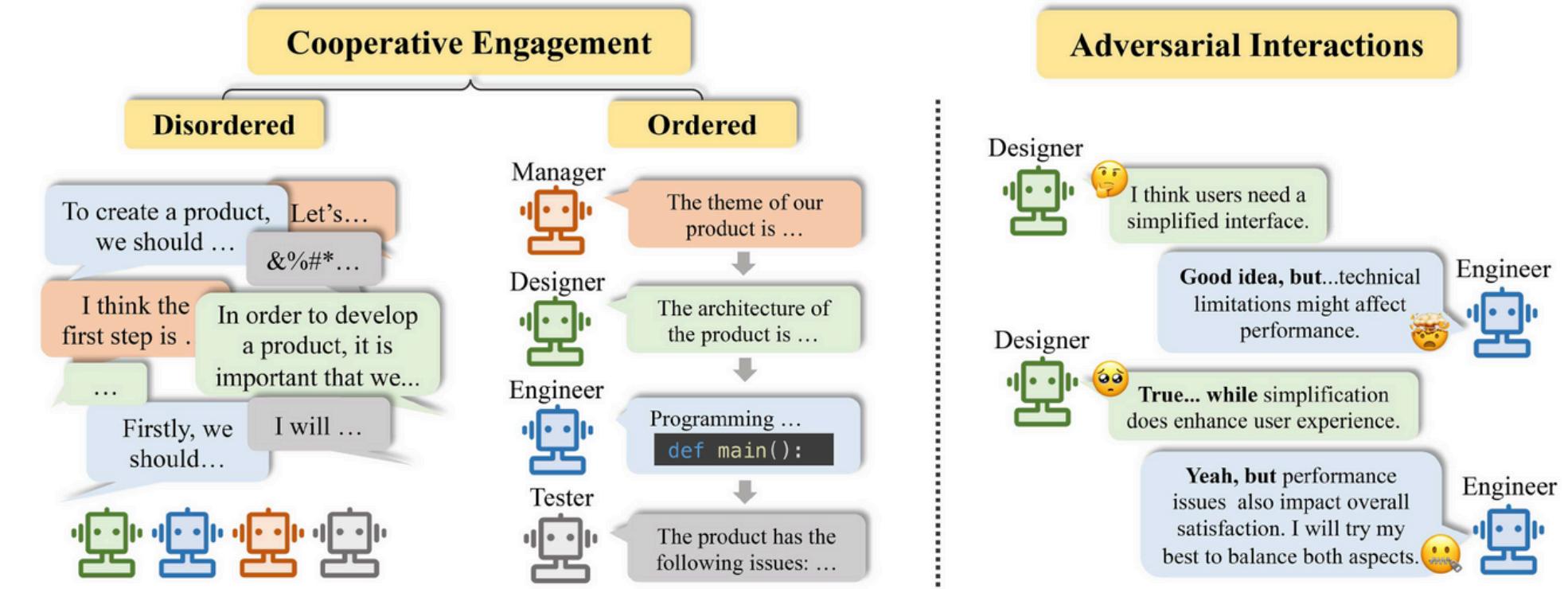


Figura 2: Diferentes Arquiteturas Multi-agents

Text-to-SQL

Transformando Linguagem Natural em Linguagem de Consulta

- Objetivo: Traduzir Linguagem Natural em Linguagens de Consulta para Banco de Dados Relacionais
- Utilizações:
 - Sistemas Conversacionais
 - Análise de Dados
- Desafios:
 - Ordenação de Predicados
 - Complexidade
 - Ambiguidade Semântica
- Benchmarks:
 - Spider 1.0, 2.0
 - BIRD

Leaderboard - Execution Accuracy (EX)						
	Model	Code	Size	Oracle Knowledge	Dev (%)	Test (%)
	Human Performance <i>Data Engineers + DB Students</i>			✓		92.96
1 Mar 11, 2025	AskData + GPT-4o AT&T CDO - DSAIR [Shkpenyuk et al. '25]	UNK		✓	75.36	77.14
2 Apr 16, 2025	CHASE-SQL + Gemini Google Cloud [Pourreza et al. '24]	UNK		✓	74.90	76.02
3 Feb 27, 2025	Contextual-SQL Contextual AI	UNK		✓	73.50	75.63
4 Dec 17, 2024	XiYan-SQL Alibaba Cloud [Yifu Liu et al. '24]	[link]	UNK	✓	73.34	75.63

Figura 3: BIRD Execution Accuracy Leaderboard (09/06/2025)

Histórico de técnicas e ferramentas em Text-to-SQL

- **Information Retrieval (IR) & Métodos por Palavras-Chave**
 - Extrai palavras-chave da pergunta do usuário e as mapeia para tabelas, colunas ou valores no banco de dados.
 - Usa templates pré-definidos (ex.: SELECT [coluna] FROM [tabela] WHERE [condição]).
 - **EX:** Microsoft English Query (1994), DISCOVER (2002)
- **Semantic Parsing**
 - Transforma a pergunta em uma árvore sintática ou representação lógica intermediária antes de virar SQL.
 - Usa gramáticas formais para entender a estrutura da pergunta.
 - **EX:** PRECISE (2003), SQLize (2017)

Trabalhos Relacionados

- **Statistical & Machine Learning (ML) Approaches**

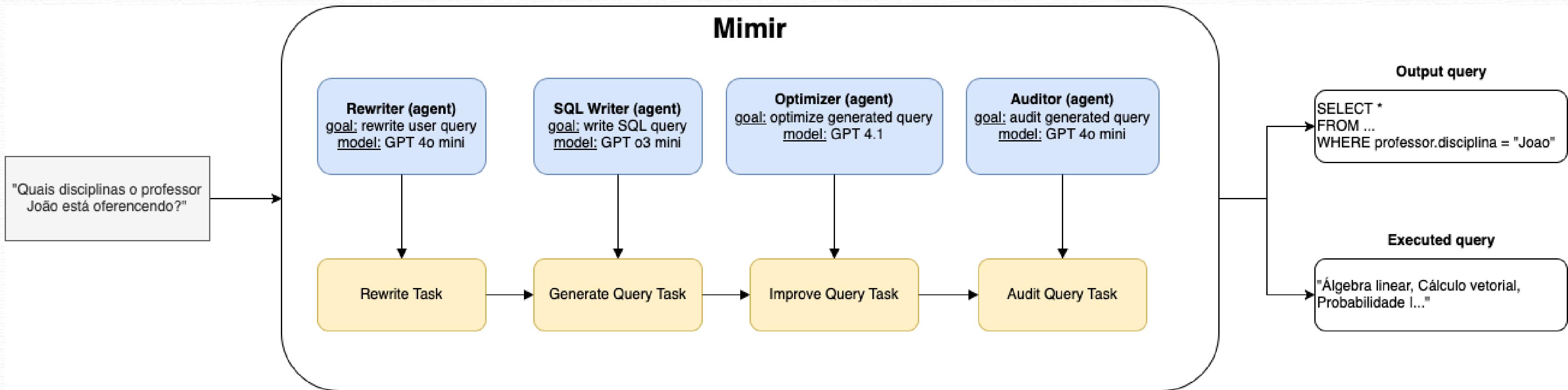
- Treina um modelo para aprender padrões entre perguntas e consultas SQL.
- Usa classificadores para identificar partes do SQL (ex.: tabelas, colunas, condições).
- **EX:** Seq2SQL

- **Deep Learning & Large Language Models (LLMs)**

- Modelos como **RAT-SQL** (2019) e **BRIDGE** (2020) usam transformers para entender o esquema do banco de dados.
- Modelos como **GPT-4** (2023) e **SQLcoder** (2024) usam conhecimento pré-treinado para gerar SQL sem treino específico.
- Dois modos:
 - Zero-shot: Gera SQL só com a pergunta e o esquema do banco.
 - Few-shot: Usa exemplos para melhorar a precisão.

Estudo de Caso

Mimir: aplicação Text-to-SQL



Estudo de Caso

Arquitetura

- 4 agentes sequenciais
- LLM-as-a-Judge
- GPT 4o, GPT o3-mini, GPT 4,1
- CrewAI + FastAPI + Node.js
- Definição de agentes e tasks
- Agente com uso de ferramentas

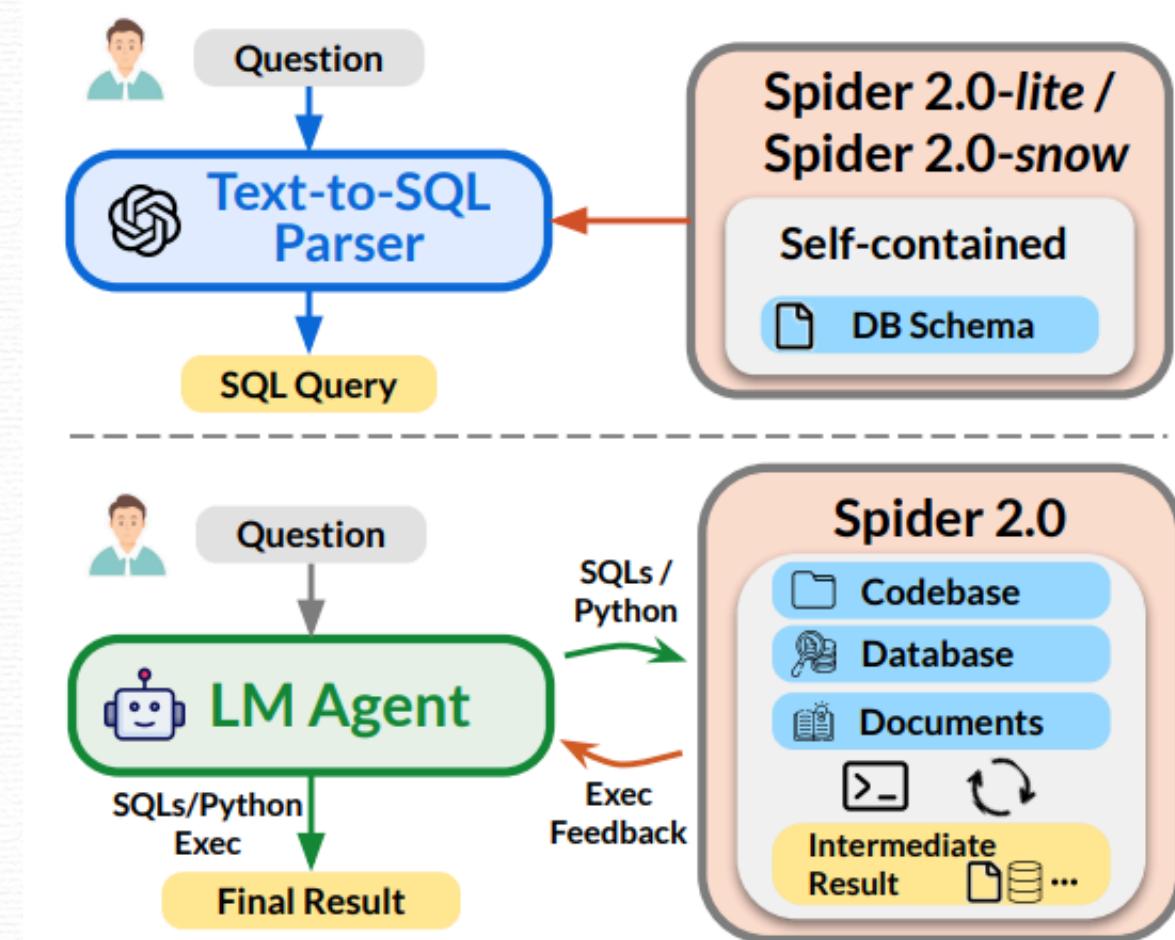
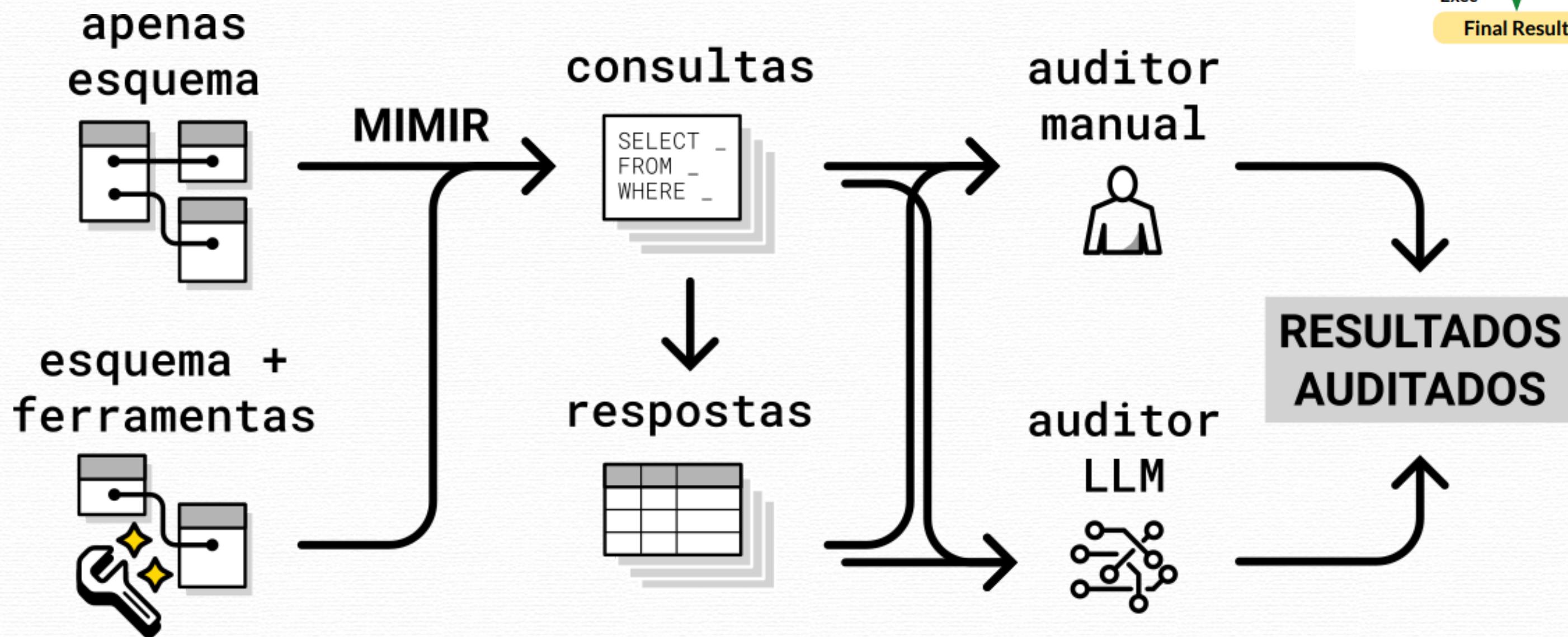
The screenshot shows a user interface for a database system named 'Mimir'. On the left, there is a sidebar titled 'Tables' with a list of tables: professors, classes, courses, students, and student_classes. The main area has a title 'Mimir' and a text input field containing the query: 'what are the names of the classes William Smith is offering?'. A blue 'Send' button is next to the input field. Below the input field, the results are displayed in a box:
Query:
SELECT courses.name AS class_name
FROM classes
JOIN professors ON classes.professor_id = professors.id
JOIN courses ON classes.course_id = courses.id
WHERE professors.name = 'William Smith'
ORDER BY courses.name ASC;
Result:
Algorithms, Introduction to Computer Science

Below this, there is a large black box containing the following text, which appears to be part of the optimizer's configuration or backstory:

```
29  ↘ optimizer:  
30  ↘   role: >  
31    | SQL Data Analyst that verifies the quality of a SQL query  
32  ↘   goal: >  
33    | Verify if SQL query provided solves the problem described in the provided  
34    | question. If not, correct the query.  
35  ↘   backstory: >  
36    | You are a skilled analyst with a background in data interpretation  
37    | and SQL queries. You are aware that people need results that are interpretable,  
38    | so you prefer to return textual results instead of considering just id's for tables.  
39    | You have a talent for identifying patterns and making sure the SQL solves the problem  
40    | described in the provided question.  
41  llm: openai/gpt-4.1-2025-04-14
```

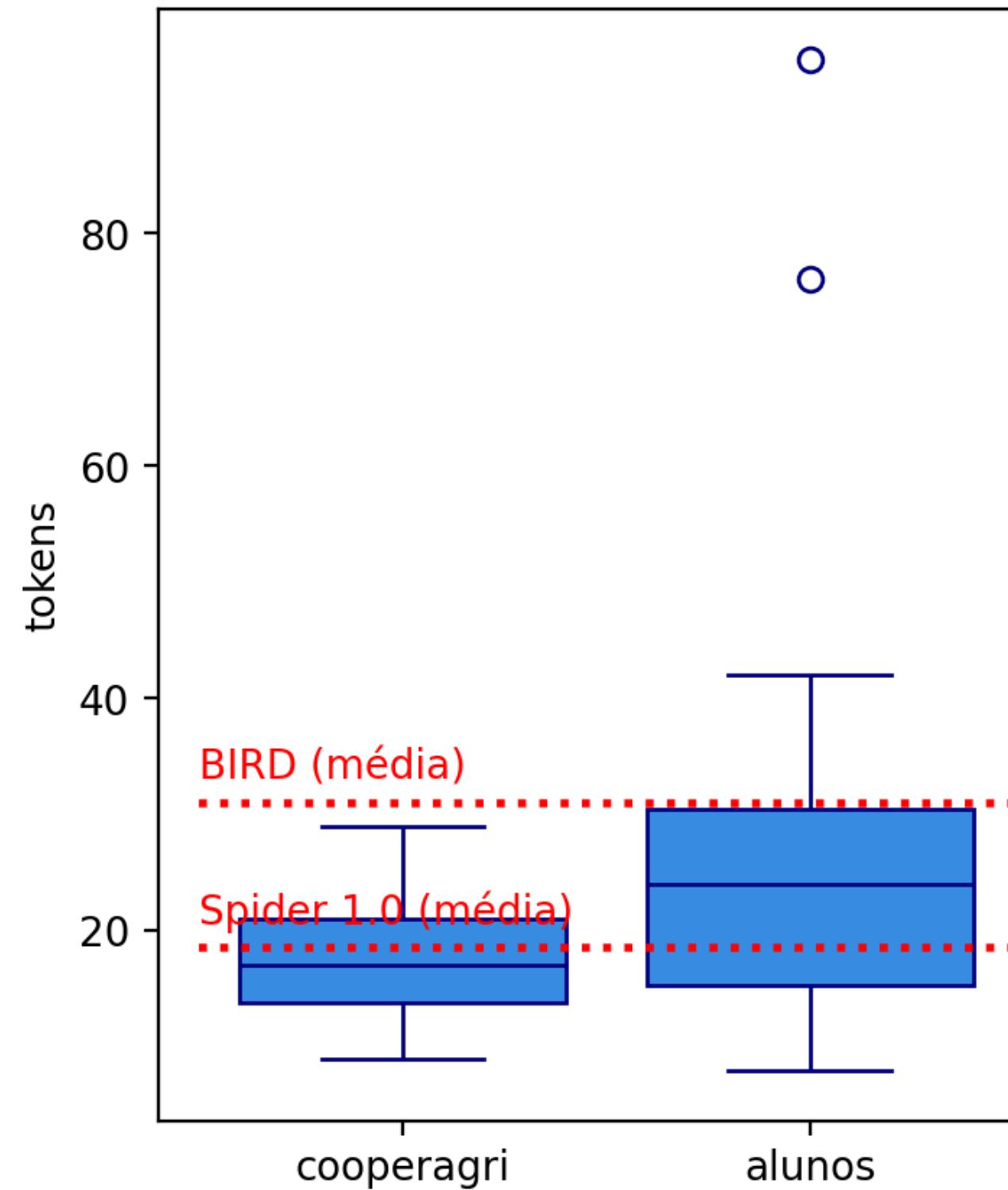
Testes

Adaptação da metodologia do benchmark Spider 2.0

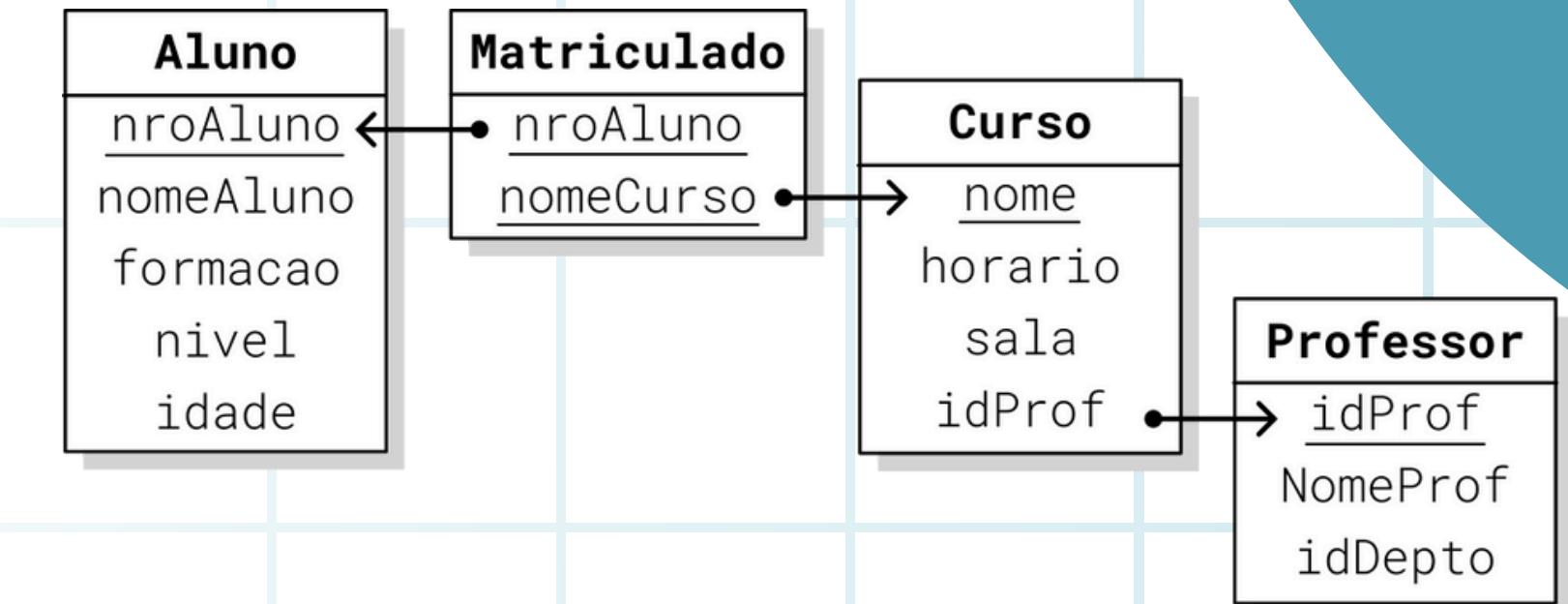


Testes

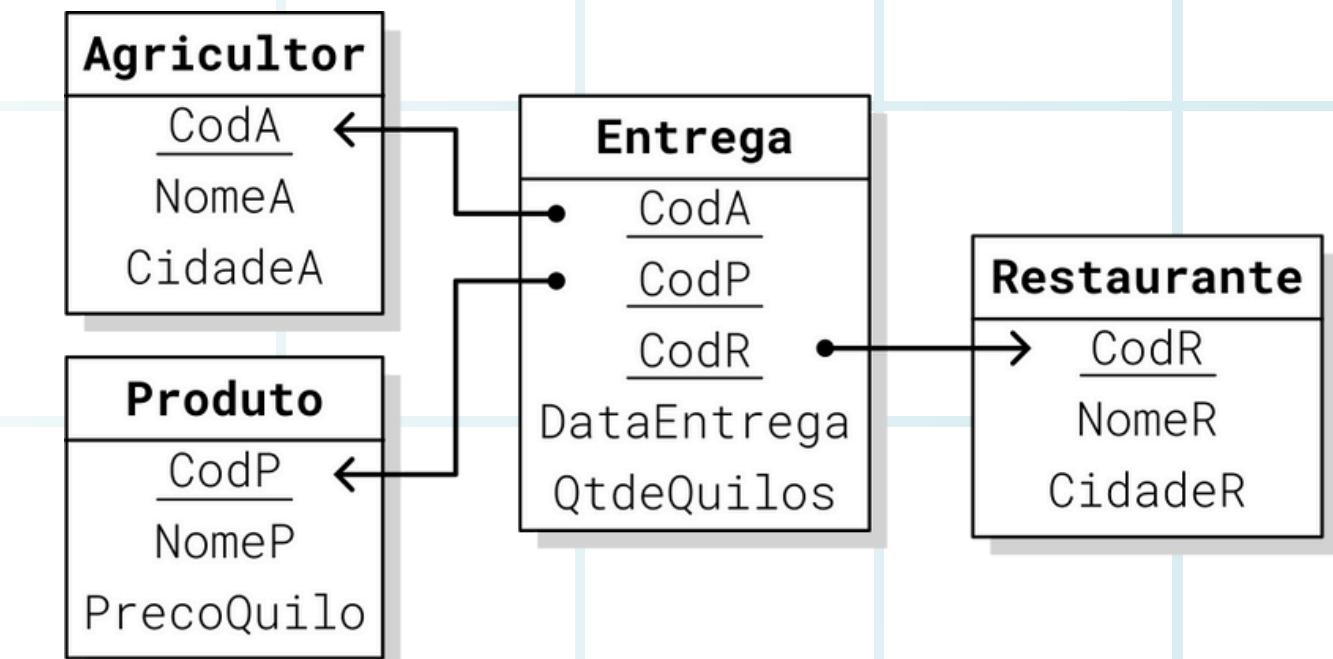
Distribuição da dificuldade por conjunto de teste



BD COOPERAGRI



BD ALUNOS



Resultados

consultas



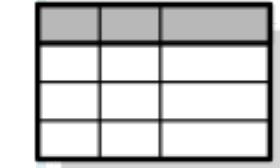
auditor
manual



auditor
LLM



respostas



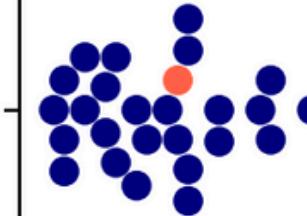
auditor
manual



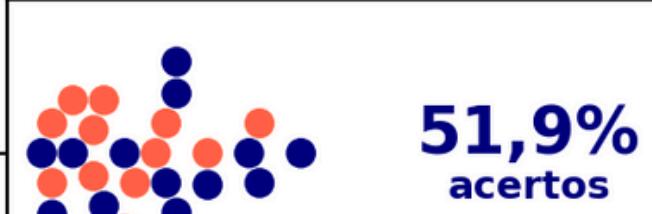
auditor
LLM



multiagente,
esquema +
ferramentas



92,6%
acertos



51,9%
acertos

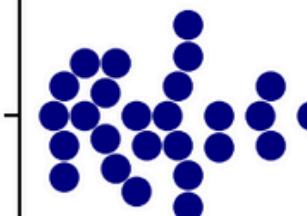


81,5%
acertos

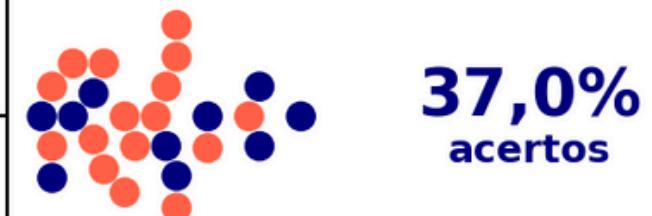


63,0%
acertos

multiagente,
apenas
esquema



96,3%
acertos



37,0%
acertos

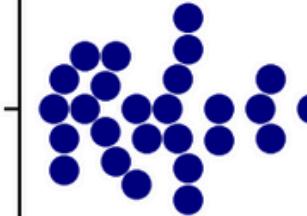


85,2%
acertos

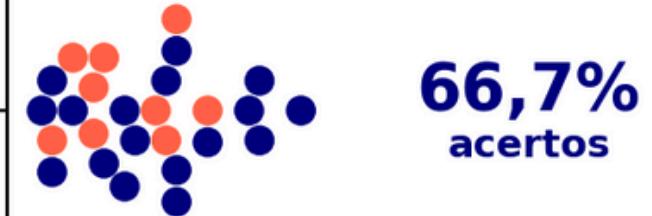


63,0%
acertos

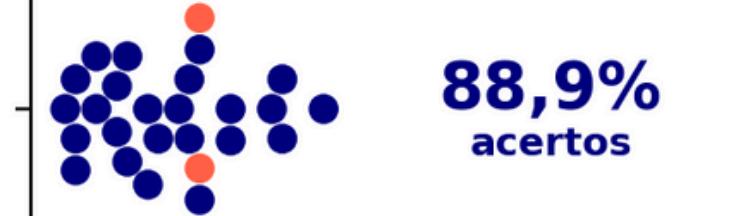
agente simples,
esquema +
ferramentas



96,3%
acertos



66,7%
acertos

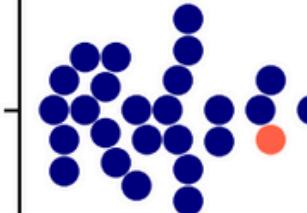


88,9%
acertos

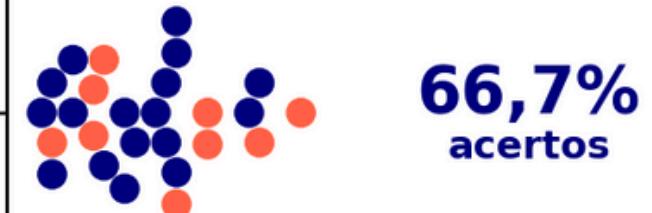


63,0%
acertos

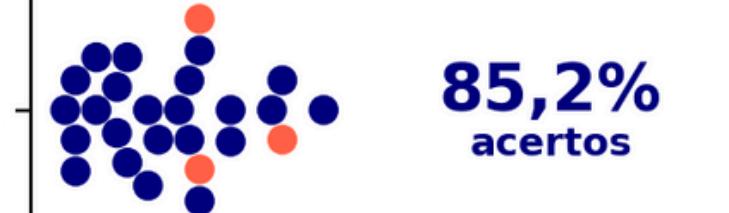
agente simples,
apenas
esquema



92,6%
acertos



66,7%
acertos



85,2%
acertos



70,4%
acertos

tokens

tokens

tokens

tokens

Resultados

Questão: Liste os códigos dos agricultores que já entregaram batatas, mas nunca entregaram cebolas.

**EXEMPLO
PADRÃO
OURO**

```
(select CodA
from Entrega natural join Produto
where NomeP = 'batata')
except
(select CodA
from Entrega natural join Produto
where NomeP = 'cebola');
```

**EXEMPLO
GERADO
POR LLM**

```
select distinct e.coda
from entrega e join produto p on e.codp = p.codp
where p.nomep = 'batatas' and e.coda not in
(select e2.coda
from entrega e2 join produto p2 on e2.codp = p2.codp
where p2.nomep = 'cebolas');
```

Conclusão e Trabalhos Futuros

A arquitetura multiagentes atingiu **resultados satisfatórios na auditoria manual**, com altos índices de acerto em consultas e resultados.

A utilização do **LLM-as-a-judge**, provou-se com um **índice elevado de falsos negativos** para utilização confiável nas especificidades de casos da nossa aplicação.

A comparação entre consultas SQL é a tarefa mais difícil.

Os próximos passos para aprimorar nossos testes seriam:

- Utilização por completo do Benchmark Spider
- Outros arranjos e papéis de agentes
- Comparativo de custos multi-agentes

Referências

- A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

Z. Xi, W. Chen, X. Guo et al., "The rise and potential of large language model based agents: A survey," *Sci. China Inf. Sci.*, vol. 68, no. 2, pp. 121101, 2025.

Blum, Adam. "Microsoft english query 7.5: Automatic extraction of semantics from relational databases and OLAP cubes." *VLDB*. Vol. 99. 1999.

V. Hristidis and Y. Papakonstantinou, "DISCOVER: Keyword search in relational databases," in *Proc. 28th Int. Conf. Very Large Databases (VLDB)*, pp. 670–681, 2002.

N. Yaghmazadeh, Y. Wang, I. Dillig et al., "SQLizer: Query synthesis from natural language," *Proc. ACM Program. Lang.*, vol. 1, no. OOP-SLA, pp. 1–26, 2017.

V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating structured queries from natural language using reinforcement learning," *arXiv preprint arXiv:1709.00103*, 2017.

B. Wang, R. Shin, X. Liu et al., "RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, pp. 7567–7578, 2020.

Lin, Xi Victoria, Richard Socher, and Caiming Xiong. "Bridging Textual and Tabular Data for Cross-Domain Text-to-SQL Semantic Parsing." *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020.

Gu, Jiawei, et al. "A survey on llm-as-a-judge." *arXiv preprint arXiv:2411.15594* (2024).