

Análise comparativa de um novo protótipo de IDE para desenvolvimento de Sistemas Multiagentes Embarcados



Elaine Maria Pereira Siqueira
Gabriel Ramos Alves Lima
Thácito Raboni
Carlos Eduardo Pantoja
Nilson Mori Lazarin

Sumário



- **Introdução**
- **Sistema Multiagente Embarcado**
- **Trabalhos relacionados**
- **Critérios de análise**
- **Novo protótipo**
- **Console – Agent Tracer**
- **Depurador – MindInspector integrado**
- **Versionamento através do Git**
- **Componentização**
- **Conclusão**

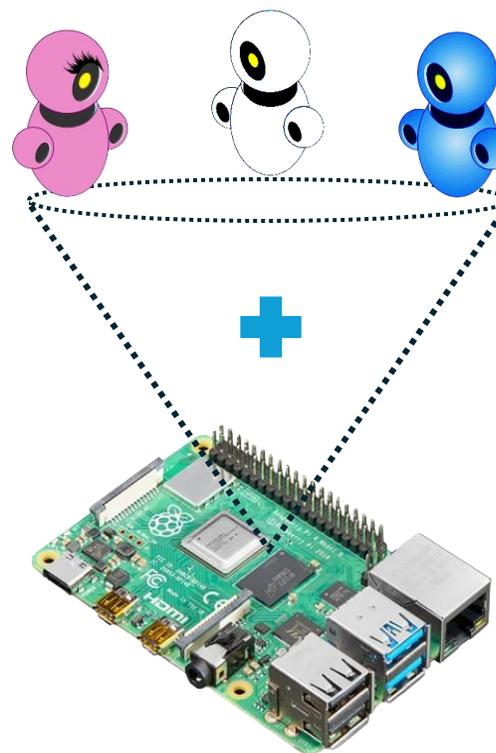


Introdução

- Processos de desenvolvimento;
- Exemplos atuais;
- Contribuições e limitações;
- Objetivo do artigo

Sistema Multiagente Embarcado

- Autonomia
- Capacidade deliberativa
- Proatividade



- Capacidade computacional
- Independência de Operação



Trabalhos Relacionados

- Escolha do Eclipse;
- Enfoque na estrutura;
- Aumento do campo de análise.

	chonIDE	Eclipse	VSCoDe	IntelliJ
Facilidade de Instalação	●●●○○	●●●●●	●●●●●	●●●●○
Multiplataforma	✗	✓	✓	✓
Multilinguagem	✗	✓	✓	✓
Livre para Uso	✓	✓	✓	✓
Interface clean	✓	✓	✓	✓
Debugger	✗	✓	✓	✓
Checagem de Sintaxe	✗	✓	✓	✓
Realce de Sintaxe	✗	✓	✓	✓
Autocompletar	✗	✓	✓	✓
Ajuda Rápida	✗	✓	✓	✓
Múltiplas Abas	✗	✓	✓	✓
Manual de Usuário	✗	✓	✓	✓
Visualizador de Arquivos	✓	✓	✓	✓
Contador de Linhas	✓	✓	✓	✓
Linha Atual Destacada	✗	✓	✓	✓
Indentação	✗	✓	✓	✓

Tabela 1. Comparativo de características entre as IDEs.

Critérios de Análise

Novo protótipo



A partir dessa análise comparativa explicada, foi possível chegar nas seguintes funcionalidades:

- Console integrado, para visualização de logs;
- Depurador integrado;
- Recursos de versionamento integradas usando o Git.

Novo protótipo



The screenshot shows the chonIDE IDE interface. The top bar includes the text "chonIDE" and navigation icons. Below the top bar, there are tabs for "helloAgent", "smith", and "Jason". The main workspace is divided into a left sidebar and a right editor. The sidebar shows a project structure with folders: "Multi-Agent System", "Agents", "Firmware", and "Libraries". The "Agents" folder is expanded, showing the "smith" agent. The "Libraries" folder is also expanded, showing "HCSR04", "Javino", and "LiquidCrystal". The right editor shows code for the "smith" agent, with line numbers 1 through 12. The code includes comments for "Initial beliefs and rules", "Initial goals", and "Plans", along with a print statement.

```
1 /* Initial beliefs and rules */
2
3
4 /* Initial goals */
5
6 !start.
7
8 /* Plans */
9
10 +!start
11 <-
12     .print("Never Send A Human To Do A Machine's Job!").
```

Versão antiga da IDE

Novo protótipo



The screenshot displays the chonIDE IDE interface for a multi-agent system simulation. The interface is divided into several panels:

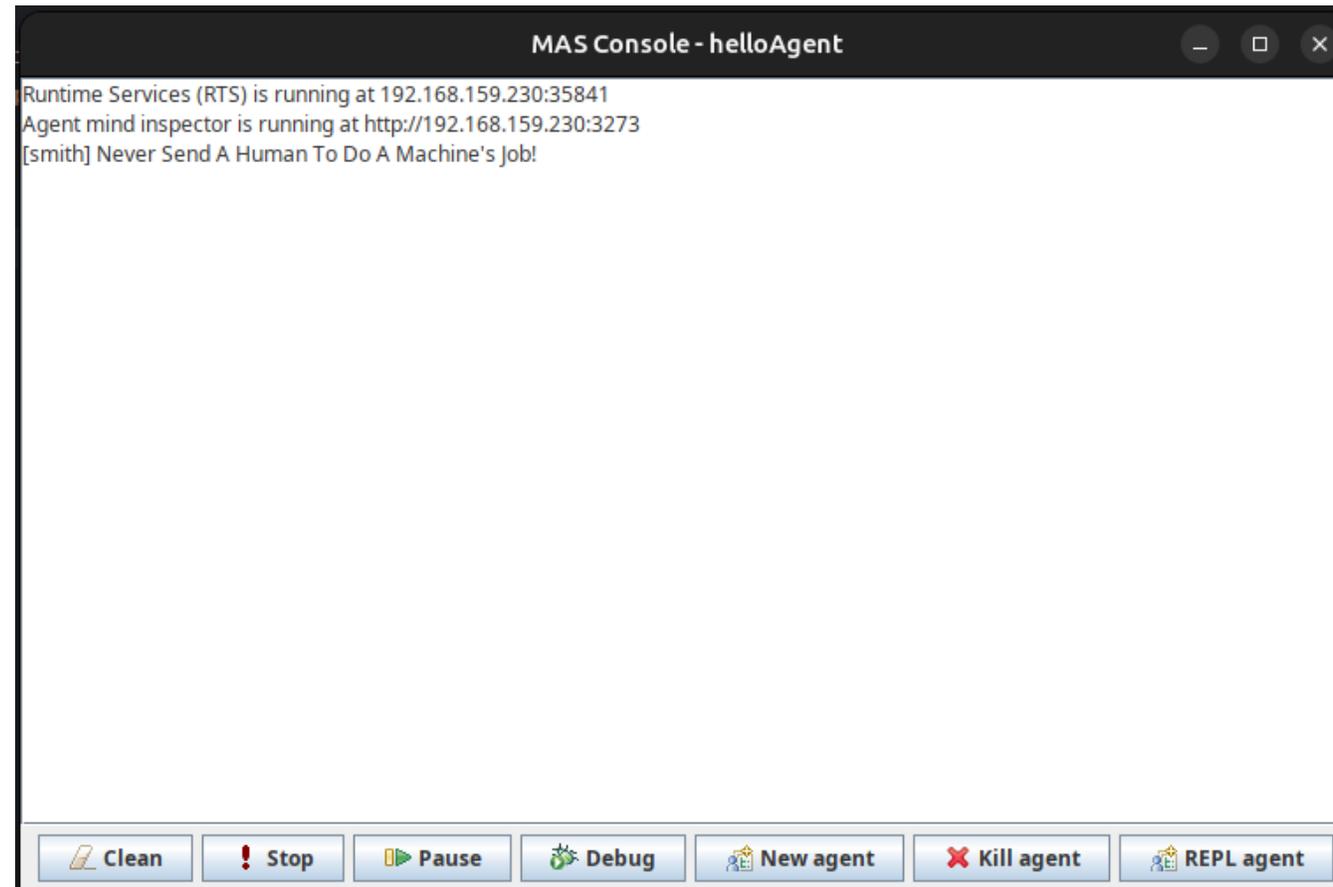
- Explorer:** Shows a project structure with folders for "Multi-Agent System", "Agents", "Firmware", and "Libraries". Under "Agents", agents "kate", "baker", "bob", and "teddy" are listed.
- Code Editor:** Displays the code for the "kate" agent, showing an order for beer and chocolate, and a method to process the order.
- Agent Tracer:** A log window showing the execution flow and messages between agents, such as "This is kate's Cart" and "I'm starting to process the order".
- Mind Inspector:** A panel for inspecting the internal state of agents, including Beliefs, Intentions, and Plans for both "kate" and "baker".

Versão nova da IDE

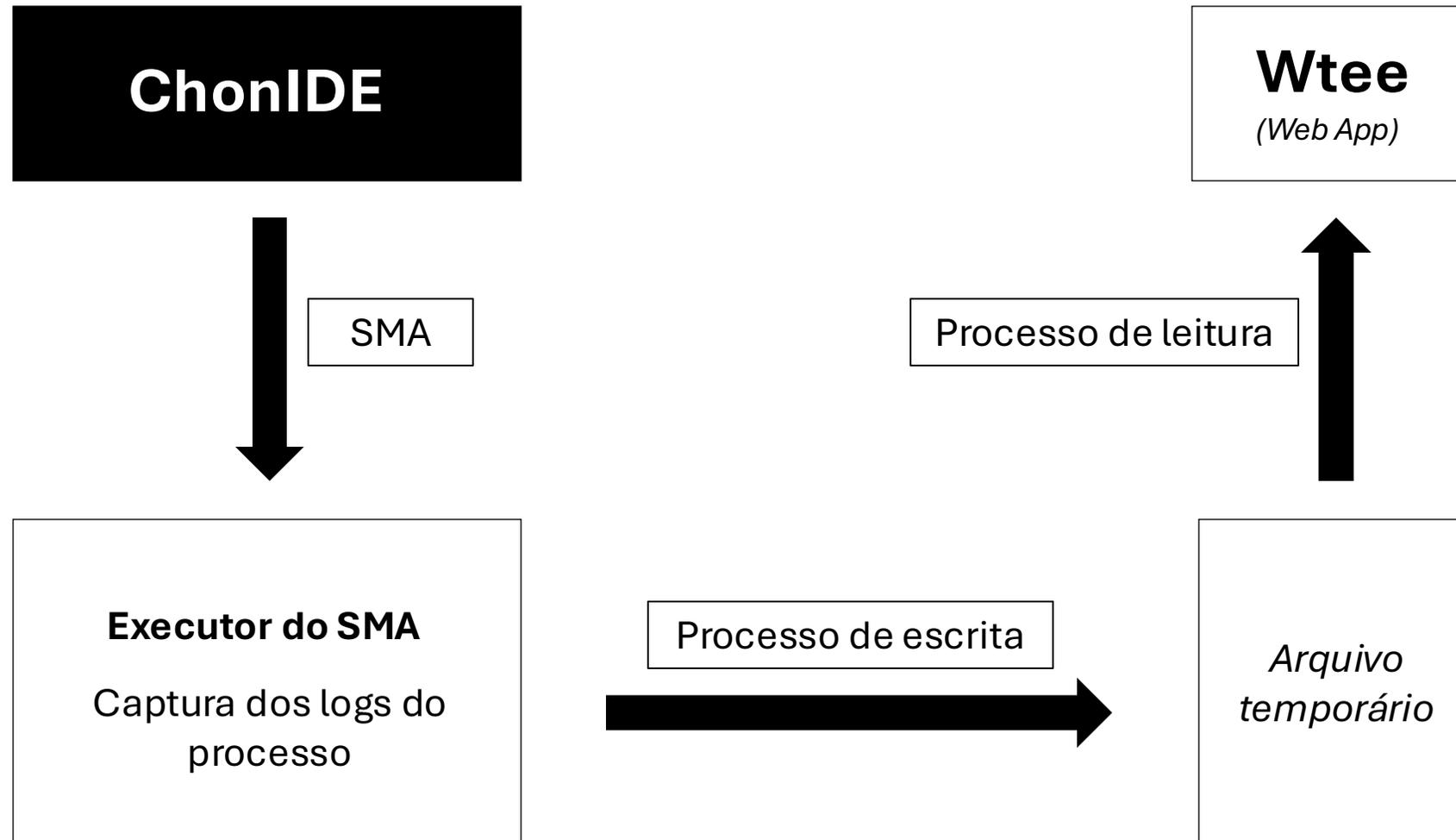
Console – Agent Tracer



O console é uma das principais mecânicas que um desenvolvedor possui para ter um retorno sobre o que está acontecendo em uma aplicação.



Console – Agent Tracer



Console – Agent Tracer



The screenshot displays the chonIDE interface. On the left is the Explorer panel with a tree view containing 'Multi-Agent System', 'Agents', 'smith', 'Firmware', and 'Libraries'. The 'Agents' folder is expanded, showing the 'smith' agent. The main editor area shows a code file named 'helloAgent' with the following content:

```
1 /* Initial beliefs and rules */
2
3
4 /* Initial goals */
5
6 !start.
7
8 /* Plans */
9
10 +!start
11 <-
12 .print("Never Send A Human To Do A Machine's Job!").
```

Below the code editor is the 'Agent Tracer' console, which shows the following output:

```
[ChonOS EmbeddedMAS] Starting the Multi-Agent System...
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
ago. 15, 2024 11:57:11 PM group.chon.jasonembedded.Main main
INFORMAÇÕES: Version 24.7.23 (Argo+Hermes+Mailer+Velluscinum)
Runtime Services (RTS) is running at 192.168.159.230:40175
Agent mind inspector is running at http://192.168.159.230:3272
[smith] Never Send A Human To Do A Machine's Job!
```

Console – Agent Tracer



The screenshot displays the chonIDE interface. On the left is the Explorer panel with a tree view containing 'Multi-Agent System', 'Agents', 'smith', 'Firmware', and 'Libraries'. The 'Agents' folder is expanded, showing 'smith'. The main editor area shows a code file with the following content:

```
1 /* Initial beliefs and rules */
2
3
4 /* Initial goals */
5
6 !start.
7
8 /* Plans */
9
10 +!start
11 <-
12 .print("Never Send A Human To Do A Machine's Job!").
```

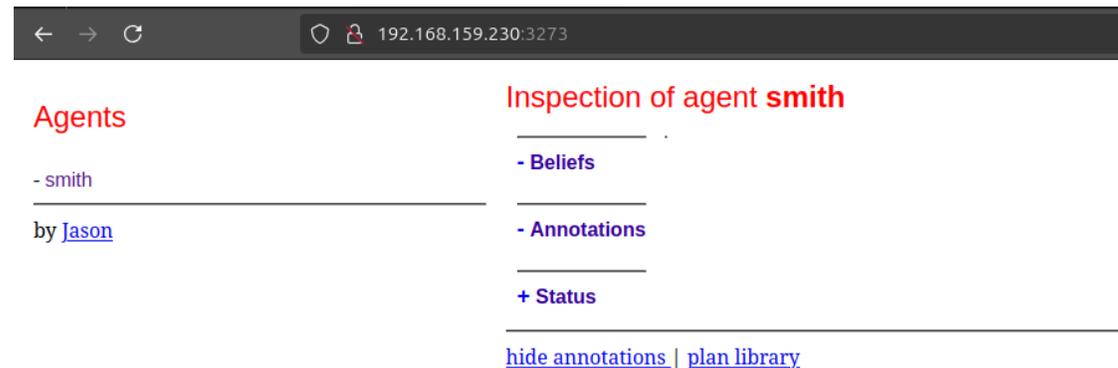
Below the code editor is the 'Agent Tracer' console window, which contains the following output:

```
[ChonOS EmbeddedMAS] Starting the Multi-Agent System...
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
ago. 15, 2024 11:57:11 PM group.chon.jasonembedded.Main main
INFORMAÇÕES: Version 24.7.23 (Argo+Hermes+Mailer+Velluscinum)
Runtime Services (RTS) is running at 192.168.159.230:40175
Agent mind inspector is running at http://192.168.159.230:3272
[smith] Never Send A Human To Do A Machine's Job!
```

Depurador – Mind Inspector integrado



- No contexto de SMA, um depurador deve ser possível visualizar as principais informações - crenças, intenções e planos executados - para cada ciclo de cada agente.



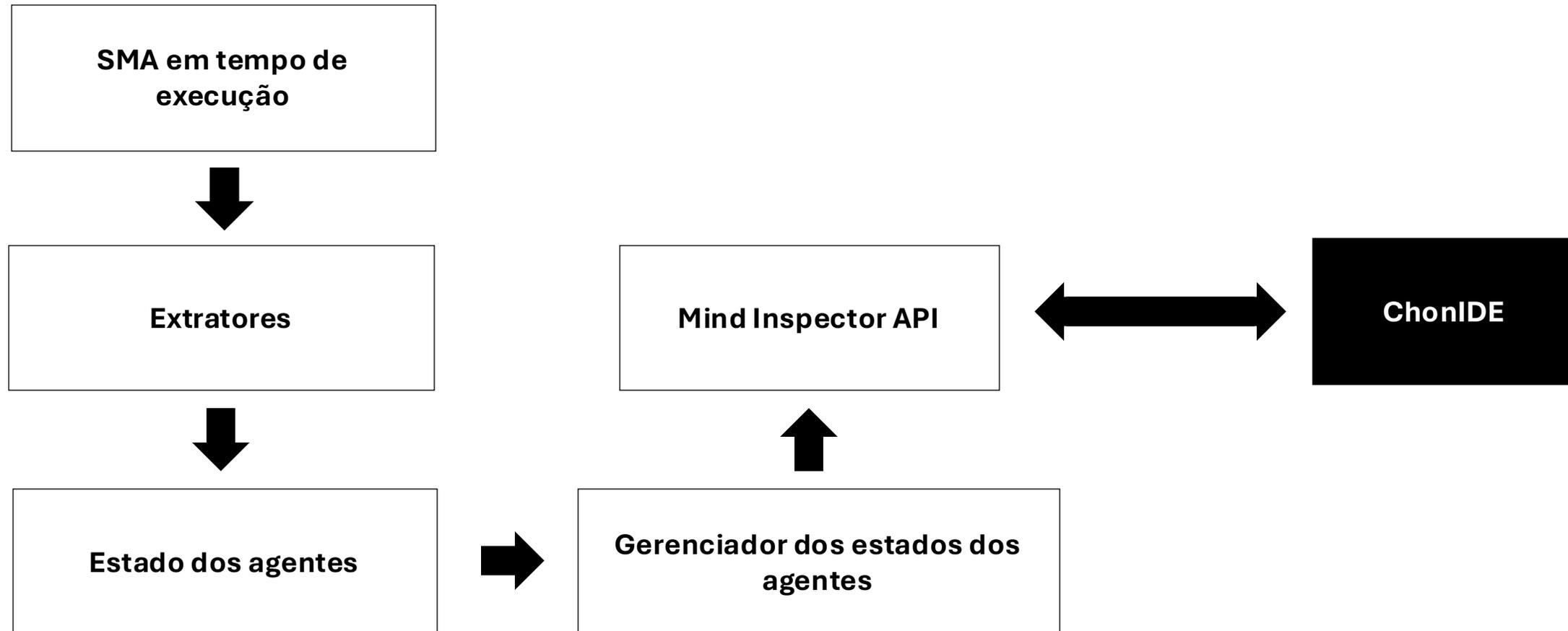
- Não existe benefício incorporar por conta da incompatibilidade visual.
- Logo, a proposta de solução para o Debugger é a criação de um Mind Inspector totalmente integrado a IDE.

Depurador – Mind Inspector integrado

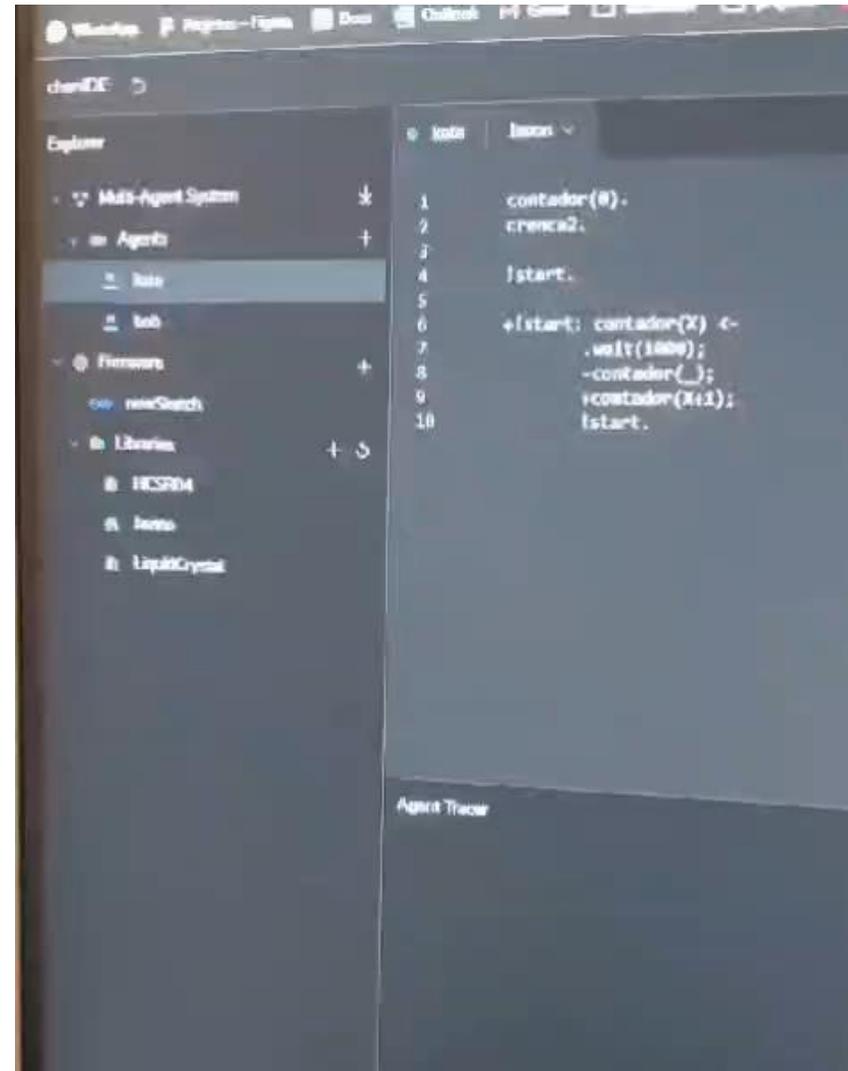


- A abordagem escolhida para o desenvolvimento do Mind Inspector integrado foi o desenvolvimento de uma API que disponibiliza o estado dos agentes sob demanda, a partir do agente e ciclo, para a interface da IDE.
- Foi realizado um estudo em cima do código fonte do Jason.
- Criação de algoritmos que extraem informações e instanciam o estado dos agentes em modelos conhecidos.
- Criação de uma instância que é responsável por gerenciar e armazenar o histórico dos estados.

Depurador – Mind Inspector integrado



Depurador – Mind Inspector integrado



Versionamento através do Git



- Gerenciamento de versões;
- Rastreamento de modificações;
- Local comum onde o código ficará armazenado.

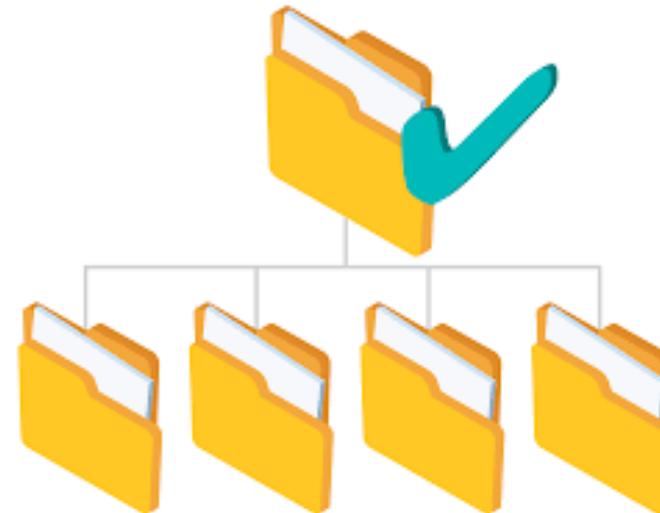
- Escalabilidade do projeto.

- A proposta da integração da IDE com o Git é disponibilizar:
 - Estado atual das alterações;
 - Interface para commit;
 - Mostrar o git log, onde será visto todos os commits do projeto;
 - Criação, exclusão e mescla entre branches.

Versionamento através do Git



- Hoje em dia todo o projeto é armazenado em um arquivo único em formato que torna inviável o versionamento.
- Arquivo de projeto deve virar uma estrutura de arquivo e pastas.



Conclusão

- Desafio das restrições técnicas;
- Contribuição da chonIDE;
- Análise, reestruturação e novas funcionalidades;
- Trabalhos Futuros.



Referências

- Souza de Jesus, V., Lazarin, N. M., Pantoja, C. E., Manoel, F. C. P. B., Alves, G. V., Ramos, G., and Filho, J. V. (2022). Proposta de uma ide para desenvolvimento de sma embarcados. In Anais do XVI Workshop-Escola de Sistemas de Agentes, Seus Ambientes e Aplicações (WESAAC 2022), pages 49–60, Blumenau. UFSC.
- Souza de Jesus, V., Mori Lazarin, N., Pantoja, C. E., Vaz Alves, G., Ramos Alves de Lima, G., and Viterbo, J. (2023). An IDE to Support the Development of Embedded MultiAgent Systems. In Mathieu, P., Dignum, F., Novais, P., and De la Prieta, F., editors, Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection, pages 346–358, Cham. Springer Nature Switzerland. DOI: https://doi.org/10.1007/978-3-031-37616-0_29.
- Russi, D. F. and Charão, A. S. (2011). Ambientes de desenvolvimento integrado no apoio ao ensino da linguagem de programação haskell. Revista Novas Tecnologias na Educação, 9(2).
- Ojo, M. O., Giordano, S., Procissi, G., and Seitanidis, I. N. (2018). A review of low-end, middle-end, and high-end iot devices. IEEE Access, 6:70528–70554.



**Muito
Obrigado!**