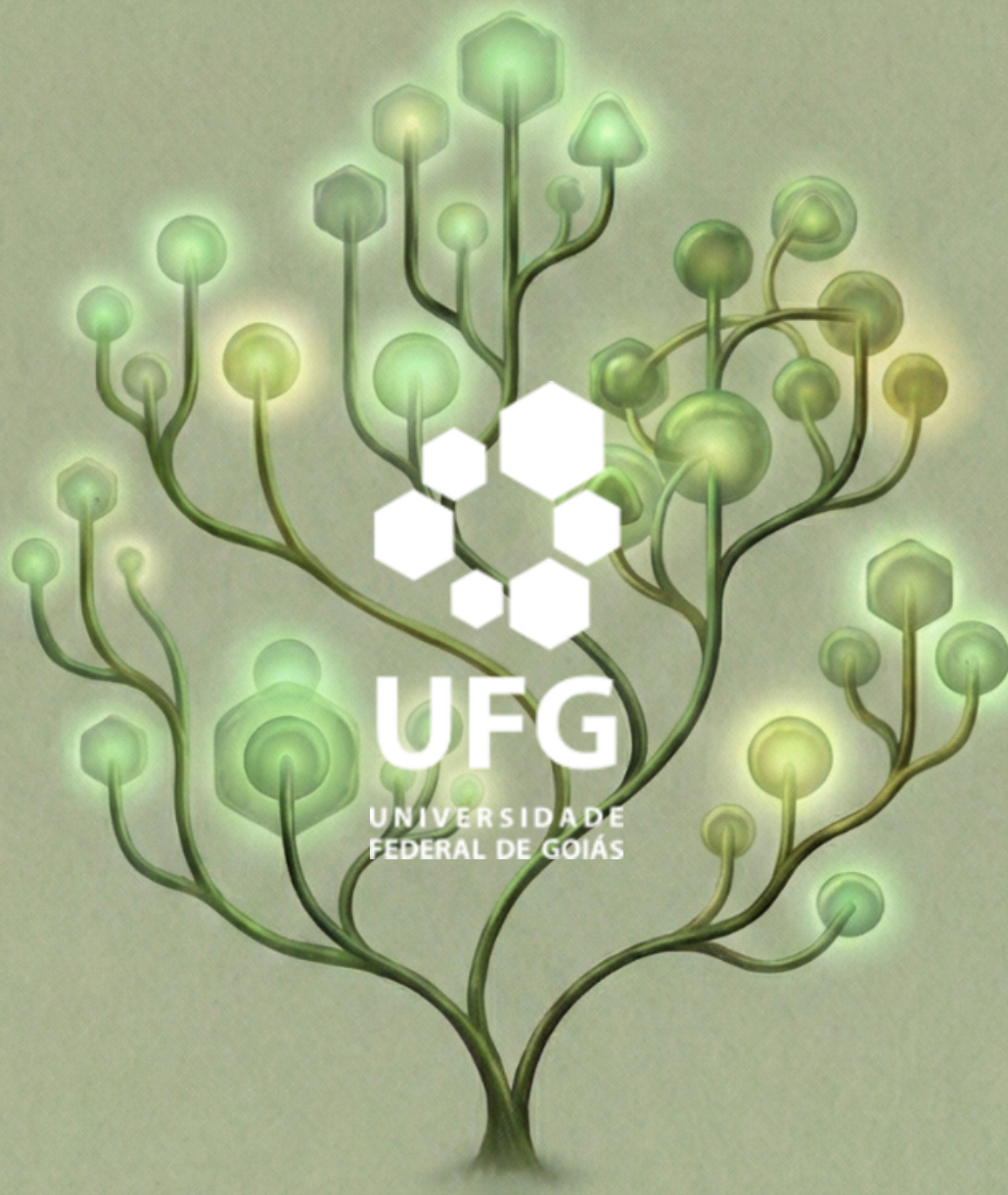


Evolução de Agentes Inteligentes

Desenvolvimento de Framework com Critério de Eficiência e
Seleção Natural

André Cerqueira Castro



UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

ANDRÉ CERQUEIRA CASTRO

Evolução de Agentes Inteligentes

Desenvolvimento de Framework com Critério de Eficiência e Seleção Natural

Goiânia
2025



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): ANDRÉ CERQUEIRA CASTRO

Título do trabalho: Evolução de Agentes Inteligentes

Desenvolvimento de Framework com Critério de Eficiência e Seleção Natural

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [] NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.



Documento assinado eletronicamente por **André Cerqueira Castro**, Usuário Externo, em 04/02/2026, às 16:46, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Fernando Marques Federson, Professor do Magistério Superior**, em 13/03/2026, às 11:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5956255** e o código CRC **FAEE5439**.

Referência: Processo nº 23070.005475/2026-10

SEI nº 5956255

ANDRÉ CERQUEIRA CASTRO

Evolução de Agentes Inteligentes

Desenvolvimento de Framework com Critério de Eficiência e Seleção Natural

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

CASTRO, ANDRÉ CERQUEIRA
Evolução de Agentes Inteligentes [manuscrito]: Desenvolvimento de Framework com Critério de Eficiência e Seleção Natural / ANDRÉ CERQUEIRA CASTRO. - 2025.
57 f.: 2025

Orientador: Prof. Dr. Fernando Marques Federson
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. Inteligência Artificial. 2. Agentes Inteligentes. 3. Seleção Natural.

I. Federson, Fernando Marques , orient. II. Título.

CDU 004


ANDRÉ CERQUEIRA CASTRO

Evolução de Agentes Inteligentes


Desenvolvimento de Framework com Critério de Eficiência e Seleção Natural

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.


Data da Aprovação: 09 de dezembro de 2025.



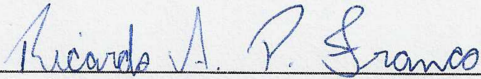
Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)



Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)



Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)



Prof. Dr. Ricardo Augusto Pereira Franco
(INF-UFG)

ANDRÉ CERQUEIRA CASTRO

Evolução de Agentes Inteligentes

Desenvolvimento de Framework com Critério de Eficiência e Seleção Natural

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Evolução de Agentes Inteligentes**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: Inteligência artificial; Agentes inteligentes; Seleção natural.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Evolution of Intelligent Agents**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

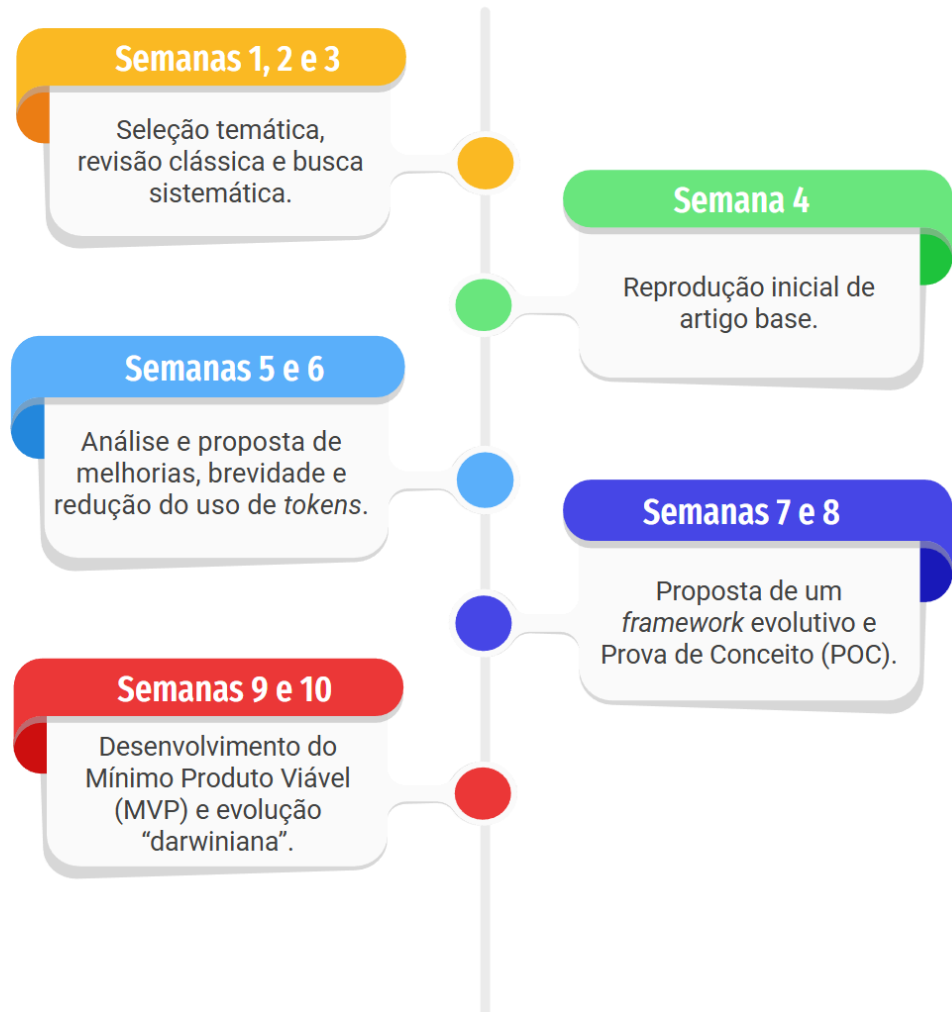
Keywords: Artificial intelligence; Intelligent agents; Natural selection.

Goiânia
2025

Minha Jornada

André Cerqueira Castro

Especialista em:
Evolução de Agentes Inteligentes



MINHA JORNADA

Nome: André Cerqueira Castro

Especialidade: Evolução de Agentes Inteligentes

Objetivo deste documento

Durante o processo da disciplina Residência em IA¹, foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha Jornada começou com uma imersão teórica e metodológica intensa durante as **Semanas 1, 2 e 3**, período que considerei fundamental para estabelecer o "solo fértil" em que toda a pesquisa cresceria. Inicialmente, na **Semana 1**, dediquei-me a explorar os anais e chamadas da *The Int'l Conf on Artificial Intelligence (ICAI)*. Foi fascinante perceber como os grandes temas da área convergem, porém meu interesse foi capturado especificamente pela intersecção entre algoritmos evolucionários e agentes inteligentes. Para solidificar essa escolha, recorri às leituras clássicas de John H. Holland, o pai dos algoritmos genéticos, e Stuart Russell e Peter Norvig, referências em agentes inteligentes. Essa base teórica não apenas justificou minha escolha, mas moldou a proposta de pesquisar a "evolução de agentes inteligentes". Com o tema definido, a **Semana 2** foi inteiramente dedicada à estruturação da Revisão Bibliográfica Sistemática. Compreendi que não bastava apenas ler; era necessário garantir a reprodutibilidade e a cientificidade da busca. Por isso, defini protocolos rigorosos: questões de pesquisa claras, *strings* de busca otimizadas com sinônimos e critérios de exclusão precisos. Validei todo esse fluxo na plataforma Parsifal, o

¹ Dez Semanas, entre setembro de 2025 e dezembro de 2025.

que me deu segurança para prosseguir. A execução prática ocorreu na **Semana 3**, um momento de verdadeira "mineração" de conhecimento. Utilizei o Gemini como motor de busca auxiliar no Google Scholar para uma triagem massiva inicial. O funil foi impressionante: de cerca de 3.500 artigos brutos, cheguei a um corpus refinado de 28 trabalhos essenciais após a desduplicação e filtragem manual. Foi neste momento crucial que encontrei o artigo "Darwin Godel Machine: Open-Ended Evolution of Self-Improving Agents" (DGM). A empolgação inicial, no entanto, logo encontrou a realidade técnica: ao tentar reproduzir o trabalho utilizando os créditos de Google Cloud (gentilmente cedidos pelo professor Sávio), o modelo Gemini 2.5 Flash enfrentou severas dificuldades com o tamanho da janela de contexto, falhando no processo de geração. Essa falha foi instrutiva, pois evidenciou logo no início as limitações dos LLMs atuais. Todo o detalhamento desse planejamento, o fluxo de filtragem e as referências fundamentais estão documentados no **Apêndice 1**.

Superada a fase de definição, a **Semana 4** marcou o início da "mão na massa" e do aprofundamento conceitual. Antes de insistir no código, realizei a leitura crítica do artigo "The Confluence of Evolutionary Computation and Multi-Agent Systems: A Survey". Esta leitura foi um divisor de águas, pois me permitiu categorizar mentalmente as abordagens existentes em duas grandes classes: a *Agent-Based Evolutionary Computation*, que injeta inteligência de agentes em algoritmos evolutivos, e a *EC-Assisted Multi-Agent Systems*, que usa a evolução para otimizar sistemas de agentes. Com esse mapa mental claro, retornei ao desafio da reprodução do DGM. Desta vez, munido de recursos mais robustos do Google Cloud Platform, substituí o modelo pelo Gemini 2.5 Pro. O resultado foi uma vitória técnica significativa: consegui realizar uma reprodução reduzida de 10 iterações. Ao observar a "árvore de evolução" gerada, notei que o agente de codificação (*coding agent*) conseguiu evoluir seus acertos no benchmark *Swe-bench* de 20% para 30%. Embora seja uma melhoria moderada se comparada às 80 iterações do artigo original (que atinge 50%), foi a validação prática de que o sistema funcionava. Ver o agente reescrever seu próprio código e obter melhores resultados foi a prova de conceito que eu precisava para seguir adiante. Os logs dessa execução, bem como a análise comparativa com o artigo original, encontram-se detalhados no **Apêndice 2**.

O período compreendido pelas **Semanas 5 e 6** foi de intensa reflexão crítica e engenharia de software. Durante a **Semana 5**, debrucei-me sobre o código e o artigo do DGM para identificar lacunas. Uma estatística do artigo original me chamou a atenção: o custo de 22 mil dólares em API apenas para *benchmarks*. Isso evidenciou que a "força bruta" não seria sustentável. Além disso, questionei a natureza mono-agente do sistema: por que evoluir apenas um agente de código? E se pudéssemos evoluir advogados, químicos ou matemáticos? Identifiquei três pilares de melhoria: Nicho (expandir para outras funções), Custo Computacional (reduzir o gasto de *tokens*) e Arquitetura (mover para multi-agentes). Na **Semana 6**, transformei essas hipóteses em matemática e código. Fui além da simples reprodução e propus uma alteração no "DNA" do algoritmo de seleção. Desenvolvi e integrei um novo critério ao mecanismo de seleção dos agentes no processo evolutivo. Em vez de avaliar apenas o sucesso na tarefa, o sistema passou a considerar também a extensão do código gerado, penalizando soluções excessivamente longas ou verbosas. Dessa forma, a pontuação final de cada agente passou a ser um equilíbrio entre sua capacidade de acerto, a novidade da solução apresentada e a sua concisão. O objetivo prático dessa alteração foi orientar a evolução para selecionar agentes que não apenas resolvam o problema corretamente, mas que o façam de maneira mais enxuta e eficiente. O detalhamento matemático completo dessas fórmulas e o código modificado estão disponíveis para consulta no **Apêndice 3**.

Nas **Semanas 7 e 8**, a jornada focou na validação dessas inovações e na expansão do horizonte da pesquisa. Na **Semana 7**, coloquei à prova a minha hipótese de brevidade. Para eliminar a variável de custo de API e ter controle total, migrei o experimento para um modelo local, o Qwen 3 Coder Instruct. Executei novamente o ciclo de 10 iterações com o "bônus de brevidade" ativado. A análise dos dados foi reveladora: ambas as versões (original e modificada) mantiveram a evolução de acurácia de 20% para 30% no *Swe-bench*, mas a minha versão modificada utilizou significativamente menos *tokens* ao longo das gerações. Isso confirmou que é possível evoluir eficiência sem sacrificar eficácia. Contudo, percebi que apenas "tunar" o DGM não resolveria a limitação do nicho (ser apenas para código). Por isso, na **Semana 8**, dei um passo ousado: desenhei um *framework* totalmente novo de evolução. Rascunhei um pipeline em que a evolução nasce da necessidade do usuário. Defini os componentes desse novo sistema: um Agente-Alvo (o objetivo), um

Agente Criador, um Planejador de Otimização e um Avaliador independente. Criei diagramas visuais para mapear essas interações e desenvolvi uma Prova de Conceito (POC) inicial. Entendi que, para mudar o domínio (de código para matemática, por exemplo), a lógica de auto-melhoria precisava ser externa ao agente, e não interna. Todos os diagramas, a estrutura do framework e os resultados da validação de *tokens* constam no **Apêndice 4**.

A reta final, nas **Semanas 9 e 10**, foi dedicada a transformar conceitos em um produto tangível e funcional. Na **Semana 9**, foquei no desenvolvimento do Mínimo Produto Viável (MVP). Escolhi o dataset matemático GSM8K devido às suas *labels* discretas, o que facilita a avaliação objetiva. Implementei os agentes de criação e desenvolvimento e rodei uma evolução sequencial em 50 amostras. O salto de performance foi emocionante: o sistema partiu de uma acurácia de 10% e, através de ciclos de auto-crítica e reescrita de *prompts*, atingiu picos de 80%, estabilizando em 70%. Na **Semana 10**, refinei esse MVP para o que batizei de *Specialist Agent Scaffold-System* (SASS). Implementei uma estratégia estritamente inspirada na seleção natural “darwiniana”, introduzindo rigor metodológico com a separação de dados em treino (para evolução), validação (para seleção) e teste (para prova final), evitando assim o *overfitting*. Adicionei um controle adaptativo do parâmetro lambda, tornando a seleção mais “gulosa” em fases iniciais. O teste final, com 250 amostras e modelos OpenAI distintos para evitar contaminação, mostrou uma evolução de acurácia de 0,1333 para 0,9333 em apenas 10 iterações. Todo esse sistema foi modularizado, documentado e disponibilizado em repositório aberto, permitindo que outros pesquisadores troquem *datasets* e modelos conforme a necessidade. Os detalhes finais do SASS, incluindo o link para o repositório público, estão no **Apêndice 5**.

Concluo esta jornada com a certeza de que a transição da teoria para a construção de um sistema evolutivo funcional representou um marco definitivo em minha formação. Todo este percurso, no entanto, não seria possível sem a base sólida e o direcionamento visionário que recebi. Registro aqui meu profundo agradecimento aos professores doutores **Fernando Marques Federson**, **Cedric Luiz de Carvalho** e **Leonardo Antônio Alves**, que conduziram e possibilitaram a realização deste trabalho. Suas orientações precisas,

provocações intelectuais e suporte contínuo foram os pilares que sustentaram cada descoberta e transformaram desafios técnicos em conhecimento consolidado.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 4 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Em minha primeira Semana, realizei as seguintes atividades:

- Selecionei meus tópicos de interesse na *The Int'l Conf on Artificial Intelligence (ICAI)*. São eles: *AI and evolutionary algorithms* e *intelligent agents*;
- Propus meu tema de pesquisa: otimização de agentes inteligentes com busca evolucionária;
- Levantei referências clássicas sobre algoritmos genéticos, uma técnica de busca evolucionária;
- Estudei essa temática sob a ótica de John H. Holland, pesquisador referência;
- Levantei referências clássicas sobre agentes inteligentes;
- Estudei essa temática sob a ótica de Stuart Russell, Peter Norvig e Michael Wooldridge.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana, pretendo:

- Buscar e analisar outros algoritmos de busca evolucionária;
- Buscar e analisar trabalhos otimização de agentes inteligentes com busca evolucionária.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

Material Complementar para a Pesquisa de Otimização de Agentes Inteligentes

O Trabalho de John H. Holland e os Algoritmos Genéticos

John H. Holland é reconhecido como o pai dos Algoritmos Genéticos (AGs). Seu trabalho principal, que fundou o campo, é o livro "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence" (1975).

A definição de Holland para um AG é a de **um método de busca que mimetiza a evolução biológica para entender o processo de adaptação**. A estrutura que ele propôs é baseada em três pilares:

1. **População:** Em vez de trabalhar com uma única solução, o AG opera sobre um conjunto de soluções candidatas, chamadas de "população". Cada solução é representada por um "cromossomo".
2. **Seleção por Aptidão:** Cada cromossomo é avaliado por uma "função de aptidão" (*fitness*) que mede quão boa é aquela solução. Simulando a "sobrevivência do mais apto", os cromossomos com maior aptidão têm mais chances de serem selecionados para se reproduzir.
3. **Operadores Genéticos:** São os mecanismos que criam a próxima geração de soluções:
 - **Crossover (Cruzamento):** O operador principal na visão de Holland. Ele combina partes de dois cromossomos "pais" para criar um novo "filho", que herda características de ambos.
 - **Mutação:** Introduz pequenas alterações aleatórias nos cromossomos. Sua função é garantir a diversidade e evitar que a busca fique estagnada em uma solução subótima.

A grande contribuição teórica de Holland foi a **Hipótese dos Blocos de Construção**. Ele postulou que os AGs funcionam de forma tão eficaz porque descobrem, propagam e recombinaem pequenos padrões de alta performance (chamados *schemata*), usando-os como "blocos de construção" para formar soluções cada vez melhores.

A Definição de Agentes Inteligentes

Existem duas definições principais e complementares no campo.

A Visão Abrangente de Russell e Norvig

Em sua obra canônica, "Artificial Intelligence: A Modern Approach" (1995), Stuart Russell e Peter Norvig definem um agente como **"qualquer coisa que pode ser vista como percebendo seu ambiente através de sensores e agindo sobre esse ambiente através de atuadores"**. O foco deles é no

agente racional, que é aquele que age para maximizar seu desempenho esperado.

Eles classificam os agentes em quatro tipos, em ordem crescente de complexidade :

1. **Agentes Reativos Simples:** agem apenas com base na percepção atual (regras "se-então").
2. **Agentes Baseados em Modelos:** mantêm um estado interno (um "modelo" do mundo) para lidar com informações que não são imediatamente perceptíveis.
3. **Agentes Baseados em Objetivos:** possuem metas explícitas e escolhem ações que os levarão a atingir esses objetivos.
4. **Agentes Baseados em Utilidade:** usam uma função de "utilidade" para escolher entre diferentes objetivos, permitindo decisões mais refinadas quando há metas conflitantes ou incerteza.

A Visão Prática de Michael Wooldridge

Michael Wooldridge, uma autoridade em sistemas multiagentes, consolida sua visão em sua principal obra, "An Introduction to MultiAgent Systems" (2002). Segundo ele, **"um agente é um sistema computacional que está situado em algum ambiente e que é capaz de ação autônoma nesse ambiente a fim de cumprir seus objetivos de projeto"**.

A partir disso, um agente inteligente é também aquele que exibe quatro propriedades essenciais:

- **Autonomia:** opera sem a intervenção direta de humanos e tem controle sobre suas próprias ações.
- **Reatividade:** percebe seu ambiente e responde de forma oportuna às mudanças.
- **Proatividade:** não apenas reage, mas toma a iniciativa para atingir seus objetivos.
- **Habilidade Social:** é capaz de interagir com outros agentes para cumprir suas metas.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 11 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana, realizei as seguintes atividades relacionadas à revisão bibliográfica sistemática:

- Estudei fontes sobre o processo de Revisão Bibliográfica Sistemática;
- A partir dos achados, elaborei e documentei um planejamento detalhado da revisão);
- No planejamento, defini:
 - Questão de pesquisa;
 - Palavras-chave e sinônimos;
 - Strings e fontes de busca;
 - Critérios de inclusão e exclusão;
 - Etapas de seleção, leitura e análise dos artigos;
 - Tópicos para comparação de metodologias e resultados.
- Estruturei o fluxo de trabalho em um diagrama para facilitar o acompanhamento;
- Testei previamente a plataforma Parsifal para validar a viabilidade da aplicação prática do planejamento.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana, pretendo:

- Iniciar busca seguindo o planejamento descrito acima;
- Utilizar a plataforma Parsifal para aplicar o planejamento.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go! ▾](#)

Material Complementar: Planejamento de Revisão Bibliográfica Sistemática

Questão de Pesquisa

“Como algoritmos evolucionários têm sido aplicados na otimização de agentes inteligentes, quais desafios têm sido enfrentados e quais avanços têm sido reportados?”

Palavras-chave e sinônimos

- **Evolutionary algorithms** (synonyms: genetic algorithms, evolution strategies, differential evolution, co-evolutionary algorithms, evolutionary computation).
- **Intelligent agents** (synonyms: autonomous agents, software agents, multi-agent systems, artificial agents).
- **Optimization** (synonyms: agent optimization, adaptive optimization, evolutionary optimization).

String de busca

("open-ended evolution" OR "open ended learning" OR "unbounded evolution")

AND ("self-improving agents" OR "self-adaptive systems" OR "autonomous agents" OR "continual learning agents"))

OR

("evolutionary algorithms" OR "genetic algorithms" OR "evolution strategies" OR "differential evolution" OR "evolutionary computation")

AND ("intelligent agents" OR "autonomous agents" OR "multi-agent systems")

AND (optimization OR "agent optimization" OR "evolutionary optimization"))

Fontes de pesquisa

- ACM Digital Library;
- IEEE Xplore;
- ScienceDirect;
SpringerLink;
- Scopus;
- Web of Science.

Critérios de Inclusão

- Artigos que tratem explicitamente do uso de algoritmos evolucionários na otimização de agentes inteligentes ou evolução *open-ended* ou auto-melhoria de agentes inteligentes ;
- Estudos que apresentem aplicações práticas, simulações ou análises comparativas de desempenho;
- Publicações revisadas por pares, em inglês;
- Revisões sistemáticas ou surveys relevantes para o tema.

Critérios de Exclusão

- Trabalhos que abordem algoritmos evolucionários ou evolução *open-ended* sem relação com agentes inteligentes ou vice-versa;
- Estudos de agentes inteligentes otimizados por métodos que não sejam evolucionários (ex.: apenas RL clássico, redes neurais, algoritmos determinísticos);
- Publicações fora do escopo acadêmico (blogs, relatórios técnicos não revisados);
- Documentos duplicados ou versões preliminares de artigos já publicados em periódicos.

Seleção Final dos Artigos

- Obtenção da lista definitiva de artigos após a aplicação dos critérios de inclusão e exclusão.

Leitura Completa dos Artigos

- Leitura e análise detalhada de cada artigo selecionado, com foco nos objetivos, métodos e resultados.

Comparação das Metodologias

- Análise comparativa das metodologias utilizadas, registrando em tabela:
 - Semelhanças e diferenças técnicas;
 - Justificativas para as escolhas metodológicas adotadas em cada trabalho;
 - Tipos de agentes e algoritmos evolucionários utilizados;
 - Ambientes ou cenários de aplicação.

Comparação dos Resultados

- Análise comparativa dos resultados obtidos, também com registro em tabela:
 - Principais avanços ou limitações relatados;
 - Se os resultados são diretamente comparáveis entre os trabalhos;
 - Destaques positivos e negativos por estudo.

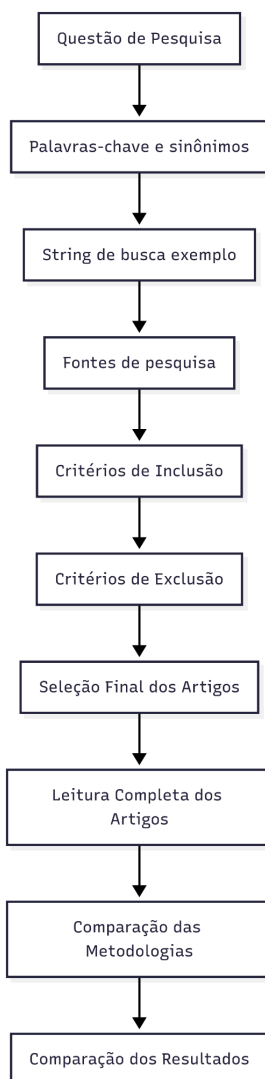


Figura 1: Diagrama de revisão bibliográfica sistemática

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 18 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta Semana, realizei as seguintes atividades:

- Apliquei o processo de busca sistemática, utilizando o Gemini como motor de busca pelo Google Scholar;
- Realizei a triagem inicial com apoio de IA, seguida da deduplicação de artigos usando ferramenta automatizada;
- De 3.500 artigos encontrados inicialmente, 28 foi a quantidade final encontrada pelo modelo generativo. Eu adicionei mais 3 artigos, que encontrei paralelamente;
- Realizei manualmente a última filtragem desse total, selecionando 5 artigos entre os anteriores;
- Desses artigos, finalizei a leitura do trabalho "Darwin Godel Machine: Open-Ended Evolution of Self-Improving Agents";
- A partir dos créditos do Google Cloud, oferecidos pelo professor Sávio e divulgados pelo professor Federson (obrigado aos dois), pude iniciar o processo de reprodução do trabalho Darwin Godel Machine como parte do estudo. Porém, por conta dos grandes tamanhos de contexto, o modelo Gemini 2.5 Flash está enfrentando dificuldades no processo de geração.
- Registre todo o fluxo de pesquisa, da busca inicial à análise final, garantindo rastreabilidade e clareza do procedimento.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana, pretendo:

- Resolver o problema com o Gemini 2.5 Flash ligado ao tamanho do contexto reproduzir o Darwin Godel Machine;
- Iniciar uma leitura do artigo *The Confluence of Evolutionary Computation and Multi-Agent Systems: A Survey*.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

Material Complementar: Processo de Pesquisa e Seleção de Artigos

Este texto descreve o processo para a seleção de artigos científicos focados na evolução de agentes inteligentes, com ênfase em sistemas de "open-endedness" e algoritmos evolucionários.

1. Passo a Passo do Processo

O processo foi estruturado da seguinte forma:

1. **Busca Sistemática:** Execução das buscas no Google Scholar.
2. **Filtragem com IA:** Aplicação dos critérios de inclusão e exclusão definidos anteriores com o auxílio do Gemini para uma triagem inicial.
3. **Desduplicação:** Remoção de artigos duplicados com o uso de uma ferramenta automatizada.
4. **Análise e Clusterização:** Identificação de grupos temáticos (clusters) nos artigos selecionados.
5. **Seleção Final e Resumo:** Escolha dos artigos mais aderentes ao tema e elaboração dos resumos.

2. Detalhes da Busca no Google Scholar

A busca foi realizada no Google Scholar com o auxílio do Gemini para otimizar a seleção de palavras-chave e a aplicação dos critérios de inclusão e exclusão.

- **Prompt Geral para o Gemini:**

"Atividade: Pesquisa Acadêmica sobre Agentes Inteligentes e Algoritmos Evolucionários. Realize uma busca no Google Scholar utilizando a seguinte string de busca: (("open-ended evolution" OR "open ended learning" OR "unbounded evolution") AND ("self-improving agents" OR "self-adaptive systems" OR "autonomous agents" OR "continual learning agents")) OR (("evolutionary algorithms" OR "genetic algorithms" OR "evolution

strategies" OR "differential evolution" OR "evolutionary computation") AND ("intelligent agents" OR "autonomous agents" OR "multi-agent systems") AND (optimization OR "agent optimization" OR "evolutionary optimization"). Os critérios de inclusão são: artigos que tratem explicitamente do uso de algoritmos evolucionários na otimização de agentes inteligentes ou da evolução open-ended ou auto-melhoria de agentes inteligentes; estudos que apresentem aplicações práticas, simulações ou análises comparativas de desempenho; publicações revisadas por pares, escritas em inglês; revisões sistemáticas ou surveys relevantes para o tema. Os critérios de exclusão são: trabalhos que abordem algoritmos evolucionários ou evolução open-ended sem relação com agentes inteligentes, ou o contrário; estudos com agentes inteligentes otimizados por métodos que não sejam evolucionários, como aprendizado por reforço clássico, redes neurais ou algoritmos determinísticos; publicações fora do escopo acadêmico, como blogs ou relatórios técnicos não revisados; documentos duplicados ou versões preliminares de artigos já publicados em periódicos. Na entrega, escreva um parágrafo explicando como a busca foi feita, quais filtros foram aplicados e como a seleção foi realizada, com uma linguagem simples como se você estivesse relatando a atividade. Informe quantos artigos foram encontrados inicialmente e quantos restaram após a filtragem.”

- **Strings de Busca Utilizadas e Justificativa:**

- ("intelligent agents" OR "multi-agent systems") AND ("evolutionary algorithms" OR "genetic algorithms"): Para capturar trabalhos que aplicam algoritmos evolucionários a agentes.
- "open-ended evolution" AND "intelligent agents": Para focar em artigos que tratam diretamente da evolução aberta de agentes.
- "open-endedness" AND "agent-based models": Para incluir modelos baseados em agentes que exploram o conceito de "open-endedness".

- **CrITÉrios de Inclusão e Exclusão e Justificativa:**

- **Inclusão:**

- Artigos que explicitamente conectam "open-endedness" com a evolução de agentes. (Garantir o foco do estudo);
- Trabalhos que aplicam algoritmos evolucionários para desenvolver comportamentos complexos em agentes. (Ampliar o escopo para a evolução de agentes em geral).
- **Exclusão:**
 - Artigos que discutem "open-endedness" em um contexto puramente teórico ou filosófico, sem aplicação a agentes. (Evitar trabalhos fora do escopo);
 - Trabalhos que mencionam agentes inteligentes de forma superficial, sem focar em sua evolução. (Manter a relevância);
 - Artigos sem revisão de pares.
- **Desduplicação:** A desduplicação foi realizada com uma ferramenta automatizada de gerenciamento de referências bibliográficas, que identifica e remove artigos duplicados com base em título, autores e DOI.

3. Identificação e Análise de Clusters

Após a seleção inicial, os artigos foram agrupados em dois clusters principais com base na similaridade de palavras-chave e resumos. Essa clusterização foi feita com o auxílio de algoritmos de aprendizado de máquina não supervisionado (como o K-Means), utilizando bibliotecas como a Scikit-Learn em Python para processamento de linguagem natural e agrupamento.

- **Cluster 1: "Teoria e Conceitos de Open-Endedness em Agentes"**
 - **Cara do Cluster:** Este grupo de artigos foca nos fundamentos teóricos, desafios e métricas para alcançar a verdadeira "open-endedness". Discute-se o que significa um sistema ser "aberto", como medir a inovação e a complexidade crescente, e as implicações para a inteligência artificial geral (AGI).
- **Cluster 2: "Aplicações de Algoritmos Evolucionários em Agentes"**
 - **Cara do Cluster:** Este cluster é mais prático e aplicado. Os artigos aqui presentes detalham a implementação de algoritmos evolucionários (genéticos, de enxame, etc.) para a evolução de comportamentos específicos em agentes, como locomoção, cooperação, competição e resolução de problemas.

4. Resultados Estatísticos da Seleção

Estágio de Triagem	Número de Artigos	Razão para Exclusão
Resultados da Consulta Inicial	3.500	
Após Revisão de Título/Resumo	120	Não relacionado a agentes inteligentes e algoritmos evolutivos ou evolução <i>open-ended</i> simultaneamente; Métodos não evolucionários; Publicação não revisada por pares.
Após Revisão do Texto Completo	28	Escopo desalinhado após análise detalhada; Versões duplicadas/preliminares.
Estudos Finais Incluídos	28	-

5. Destaques de Artigos Selecionados

A seguir, resumos sucintos de alguns dos principais artigos selecionados que tratam da evolução de agentes:

- **"Darwin Godel Machine: Open-Ended Evolution of Self-Improving Agents"**: Propõe um sistema, o DGM, que modifica seu próprio código de forma iterativa, inspirado na evolução darwiniana e na pesquisa de "open-endedness". O sistema demonstra uma melhoria autônoma na capacidade de codificação.
- **"Agent-based evolutionary algorithms: Emerging paradigm or buzzwords?"**: Analisa a sinergia entre algoritmos evolucionários e sistemas multi-agente. Discute como as propriedades dos agentes (autonomia, comunicação, aprendizado) podem ser usadas para aprimorar o desempenho dos algoritmos evolucionários.

5. Lista Completa de Artigos Selecionados para Análise

Abaixo está a lista completa dos artigos que foram selecionados para a análise final, focados na evolução de agentes:

Apêndice A: A Coleção Curada: Títulos de Todos os Artigos Selecionados

1. *Open-Ended Evolution: Perspectives from the OEE Workshop in York*
2. *Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents*
3. *Progress and challenges in adaptive robotics*
4. *Formal Definitions of Unbounded Evolution and Innovation Reveal Universal Mechanisms for Open-Ended Evolution in Dynamical Systems*
5. *Towards Lifelong Open-Ended Learning Autonomy in Robotics: A Cognitive Architecture Approach*
6. *Reward-Driven Self-Assembly in a Population of Agents for Open-Ended Learning*
7. *Evolutionary Algorithms: A Critical Review and Future Directions*
8. *The Confluence of Evolutionary Computation and Multi-Agent Systems: A Survey*
9. *Evolutionary Reinforcement Learning: A Systematic Review and Future Directions*
10. *Neuroevolution: A Different Kind of Deep Learning*
11. *A Systematic Literature Review of the Successors of “NeuroEvolution of Augmenting Topologies”*
12. *Why We Do Not Evolve Software? Analysis of Evolutionary Algorithms*
13. *Continual Learning in an Open and Dynamic World*
14. *A Survey on Evolutionary Computation for Multi-Agent Systems*
15. *Multi-agent Optimization using Evolutionary Algorithms: A Survey*
16. *Evolutionary Game Theory and Multi-Agent Systems*
17. *Co-evolutionary Algorithms for Multi-Agent Learning*
18. *Evolving Neural Networks for Autonomous Agents in Complex Environments*
19. *Differential Evolution for Agent-Based Model Calibration*
20. *Open-Ended Learning in Symmetric Zero-Sum Games*
21. *Continual Learning for Autonomous Agents: A Survey*
22. *Self-Adaptive Systems: A Survey of Current Trends*
23. *Genetic Algorithms for the Optimization of Agent Behavior in Simulations*
24. *Evolution Strategies as a Scalable Alternative to Reinforcement Learning*
25. *POET: Endlessly Generating Increasingly Complex and Diverse Learning Environments and their Solutions*

-
26. *Unbounded Discovery of Stepping-Stone Skills*
 27. *Quality-Diversity: A New Frontier for Evolutionary Computation*
 28. *Evolving Plastic Neural Networks for Continual Learning*
 29. *AgentBreeder: Mitigating the AI Safety Impact of Multi-Agent Scaffolds via Self-Improvement*
 30. *Automated Design of Agentic Systems. (ADAS)*
 31. *Evolutionary multi-agent systems*

6. Lista Final de Artigos Seleccionados por mim

1. *Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents*
2. *The Confluence of Evolutionary Computation and Multi-Agent Systems: A Survey*
3. *AgentBreeder: Mitigating the AI Safety Impact of Multi-Agent Scaffolds via Self-Improvement*
4. *Automated Design of Agentic Systems. (ADAS)*
5. *Evolutionary multi-agent systems*

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 24 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Tema: Otimização de agentes inteligentes por meio da computação evolucionária.

Nessa semana da Residência, realizei as seguintes atividades:

- Realizei a leitura do artigo “The Confluence of Evolutionary Computation and Multi-Agent Systems: A Survey” ([link](#)).
- Neste trabalho, os autores definem duas principais classes de confluência:
 - **Agent-Based Evolutionary Computation:** Incorpora características dos *multi-agent systems* (MAS) (autonomia, percepção local, interação) nos algoritmos de *evolutionary computation* (EC) para, por exemplo:
 - Melhorar a paralelização;
 - Evitar convergência prematura;
 - Aumentar a diversidade de soluções.
 - **EC-Assisted Multi-Agent Systems:** utiliza EC para resolver problemas de otimização dentro de MAS para, por exemplo:
 - Otimização de estratégias de equipes;
 - Alocação de tarefas;
 - Planejamento de caminhos para múltiplos robôs;
 - Jogos com múltiplos agentes (multi-player games).
- Realizei uma reprodução reduzida (10 iterações) do “Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents” ([repositório original](#)), utilizando o modelo Gemini 2.5 Pro com recursos do Google Cloud Platform, compartilhados pelo professor Sávio.
 - **Resultados:** em um conjunto de 10 tarefas simples do Swe-bench, o *conding agent* evoluiu os acertos de 20% para 30% ([árvore de evolução](#)), o que é uma melhoria moderada, mas condiz com o cenário em que houve 10 iterações de auto-melhoria. O artigo original realiza 80 iterações e chega ao melhor agente com 50% de acerto (incluindo tasks mais desafiadoras no cálculo de score).

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana, pretendo:

- Analisar fragilidades do “Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents” (DGM) e estudar alternativas de melhoria;
 - (Potencialmente) Focar na questão da alta demanda computacional do DGM e analisar como equilibrar custo computacional e precisão.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 1 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Tema: Otimização de agentes inteligentes por meio da computação evolucionária.

Nessa Semana da Residência, eu fiz uma análise profunda do trabalho (artigo e código) “Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents” (DGM) com foco em buscar espaços de melhoria do método proposto. Descubri, a partir da experiência de reprodução do código da semana passada e estudo do trabalho, que existem, no mínimo, os seguintes prováveis espaços de melhoria:

- **Nicho/função:** o DGM evolui única e exclusivamente *coding agents*.
Pergunta: podemos evoluir também agentes matemáticos, químicos, de conhecimento geral, de vendas, atendimento ao consumidor ou qualquer outra função?
Resposta: conceitualmente, sim, uma vez que o pipeline do DGM, por mais que foque em *coding agents* e estabeleça seu código assim, não é intrínseco a estes.
- **Foundation Models - LLM:** o modelo a ser utilizado pelo *coding agent* e sua face de proposição de autmelhorias é definido pelo programador.
Pergunta: podemos tornar a escolha do foundation model parte do processo evolutivo?
Resposta: na configuração proposta pelo DGM, não, uma vez que a escolha é fixa e não há um caminho óbvio de como inserir essa dinamicidade de escolha dos modelos, dado que, para o agente propor a melhoria, ele necessariamente já precisa de um modelo configurado. Porém, é possível propor um novo pipeline que aplique essa possibilidade.
- **Custo computacional:** 22 mil dólares de LLM API em apenas experimentos de *benchmark* (SWE-bench) é uma quantia significativa.
Pergunta: podemos reduzir esse custo?
Resposta: na configuração atual do DGM, é possível utilizar modelos mais leves e menos custosos, mas isso depende da configuração feita pelo programador. Uma ótima possibilidade é transformar a quantidade de tokens do repositório do agente em parte dos critérios de escolha do agente que será automodificado.
- **Agente único:** o DGM trabalha com apenas um agente, que dispõe de ferramentas, não havendo, portanto, a abordagem multiagêntica.
Pergunta: podemos tornar a evolução multiagêntica?

Resposta: na configuração atual do DGM, não, porém, em um pipeline que parta de uma indicação de configuração multiagêntica, o agente que autoevolui pode incorporar seu sistema em um sistema multiagente.

Em virtude desses pontos levantados, propus as seguintes hipóteses de soluções:

- Em vez de focar em um *coding agent*, a evolução pode focar no que um usuário x solicitou como sistema agêntico, passando a ter, então, um *agent of agents* ou *meta-agent*.

Exemplo: “Quero um sistema agêntico especialista em solucionar dúvidas sobre direito penal com as seguintes condições...”.

Nesse cenário, o *agent of agents* ou *meta-agent* poderia dispor, inicialmente, da possibilidade de criar *subagents* especialistas em um sistema multiagente e utilizar *tools* pré-configuradas, que possibilitem, via API, acessar *benchmarks* diversos, para uma avaliação centrada no seu objetivo principal. No caso do exemplo (tirar dúvidas sobre direito penal);

- Além disso, o tamanho do repositório de um agente x, calculado por *tokens* ou caracteres, poderia minimizar a chance de sua escolha para reprodução da árvore evolutiva, dado que o baixo custo computacional deveria ser um “gene” a ser perpetuado pelas gerações;
- Sobre a dinamização da escolha dos *foundation models*, ainda preciso explorar mais as possibilidades de aplicação, considerando que não quero perder o caráter “open-ended” e “self-improving” do trabalho.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana, pretendo:

- Iniciar a implementação das hipóteses de soluções propostas, a começar pela adição do fator “custo computacional” como parte da função de escolha do agente que irá passar por auto-melhoria.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 9 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Tema: Otimização de agentes inteligentes por meio da computação evolucionária.

Na Semana passada, propus: “Iniciar a implementação das hipóteses de soluções propostas, a começar pela adição do fator “custo computacional” como parte da função de escolha do agente que irá passar por auto-melhoria.”

Nessa Semana, ajustei a fórmula de seleção dos pais do “Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents (DGM)” para inclusão de um **bônus de brevidade:**

- **Ajuste na Fórmula:**
 - **Desempenho**
 $a_i = \text{performance}(a_i)$
Avalia quão bem o agente executa uma tarefa ou benchmark proposto.
 - **Contagem de Descendentes Funcionais**
 $n_i = \text{functioning_children_count}(a_i)$
Indica quantos descendentes gerados por esse agente obtiveram sucesso, sendo utilizados como sinal de exploração ou saturação.
 - **Tamanho do Código**
 $c_i = \text{character_count}(a_i)$
Mede o número de caracteres no código do agente, servindo como proxy de complexidade ou concisão.
 - **Desempenho Escalonado**
 s_i
Representa uma versão normalizada do desempenho, ajustada para um intervalo comum (como 0 a 1), facilitando comparações.
 - **Fator de Novidade**
 $h_i = 1 / (1 + n_i)$

Dá preferência a agentes com poucos descendentes funcionais, incentivando a diversidade na população.

- **Fator de Brevidade**

Valoriza agentes com implementações mais concisas, atenuando o impacto do tamanho por meio de uma função logarítmica.

- **Pontuação Composta para Seleção**

$$w_i = s_i \cdot h_i \cdot b_i$$

Combina os três fatores principais, desempenho, novidade e brevidade, em uma única medida que orienta a escolha dos agentes mais promissores como pais.

- **Implementação em código:**

- Função Adicionada: `get_dir_size_in_chars(path)`

- O que faz: Calcula o total de caracteres no diretório de um agente.

- Função Modificada: `choose_selfimproves()`

- O que faz: No método '`score_child_prop`', a probabilidade de seleção de um agente agora é o produto de 3 fatores: performance, novidade (nº de filhos) e brevidade (tamanho do código).

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana pretendo:

- Validar a nova versão implementada e avaliar de a seleção dos pais considerando o bônus de brevidade está afetando o desempenho;
- Implementar uma métrica de cálculo do uso de *tokens* para avaliar se o custo computacional está sendo reduzido;
- Iniciar implementação de melhoria de “**Nicho/função**”: Em vez de focar em um *coding agent*, a evolução pode focar no que um usuário x solicitou como sistema agêntico, passando a ter, então, um *agent of agents* ou *meta-agent*.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

Material Complementar: Ajuste na Seleção dos Pais

Melhoria: dado o alto custo computacional do DGM em relação a chamada dos modelos, identifiquei um novo espaço de melhoria que pode auxiliar na redução de custos como parte dinâmica da própria evolução *open-ended*:

- **Inclusão de um bônus de brevidade**, ou seja, bônus para agentes que possuem repositórios mais sucintos.
 - Fórmula de Seleção de Pais (Ajustada)
 - (performance) $\alpha_i = \text{performance}(\text{ait})$
 - Mede o quão bem o agente funciona em um benchmark
 - (contagem de filhos) $n_i = \text{functioning_children_count}(\text{ait})$
 - Conta quantos descendentes funcionais o agente já gerou.
 - (tamanho do código) $c_i = \text{character_count}(\text{ait})$
 - Mede o tamanho do código do agente em caracteres.
 - (performance escalonada)

$$s_i = \frac{1}{1 + \exp(-\lambda(\alpha_i - \alpha_0))}$$

-
- Converte a performance bruta para uma escala de 0 a 1.
- (bônus de novidade) $h_i = 1/(1+n_i)$
 - Favorece agentes com menos filhos funcionais, incentivando a exploração.
- **(bônus de brevidade)**

$$b_i = \frac{1}{1 + \gamma \cdot \ln(c_i)}$$

-
- Favorece agentes com código menor, usando o logaritmo para normalizar o impacto do tamanho.
- (performance final) $w_i = s_i \cdot h_i \cdot b_i$

- Combina performance, novidade e brevidade em uma única pontuação.

- **Implementação em código:**
 - Função Adicionada: `get_dir_size_in_chars(path)`
 - O que faz: Calcula o total de caracteres no diretório de um agente.
 - Função Modificada: `choose_selfimproves()`
 - O que faz: No método '`score_child_prop`', a probabilidade de seleção de um agente agora é o produto de 3 fatores: performance, novidade (nº de filhos) e brevidade (tamanho do código).

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 16 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Tema: Otimização de Agentes Inteligentes por meio da Computação Evolucionária.

Na Semana passada, propus validar a nova versão de seleção dos pais considerando o bônus de brevidade; implementar uma métrica de cálculo do uso de *tokens*; iniciar implementação de melhoria de “nicho/função”.

Portanto, nesta Semana:

- Realizei uma nova execução reduzida (10 iterações) do DGM, com uso de um modelo local (Qwen 3 Coder Instruct) a fim de “excluir” a dependência de modelos e gastos com API;
- Realizei a execução reduzida do código com o bônus de brevidade implementado;
- Adaptei o código para registrar a quantidade de *tokens* de cada agente e, portanto, avaliar se o bônus surtiu efeito positivo;
- A partir dos experimentos, observei que ambas as versões evoluíram de 20% para 30% de acurácia no *benchmark* Swe-bench, demonstrando progresso na performance geral. A versão modificada se destacou por utilizar, em média, menos tokens ao longo da evolução, indicando uma melhora significativa em eficiência computacional;
- Registrei as árvores e evolução e resultados no material complementar.
- Após isso, iniciei a implementação da melhoria de “nicho/função”, rascunhando um novo pipeline a ser implementado (este ainda está em construção).

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana pretendo:

- Dar sequência na implementação de um pipeline de evolução open-ended de agentes de automelhoramento próprio.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

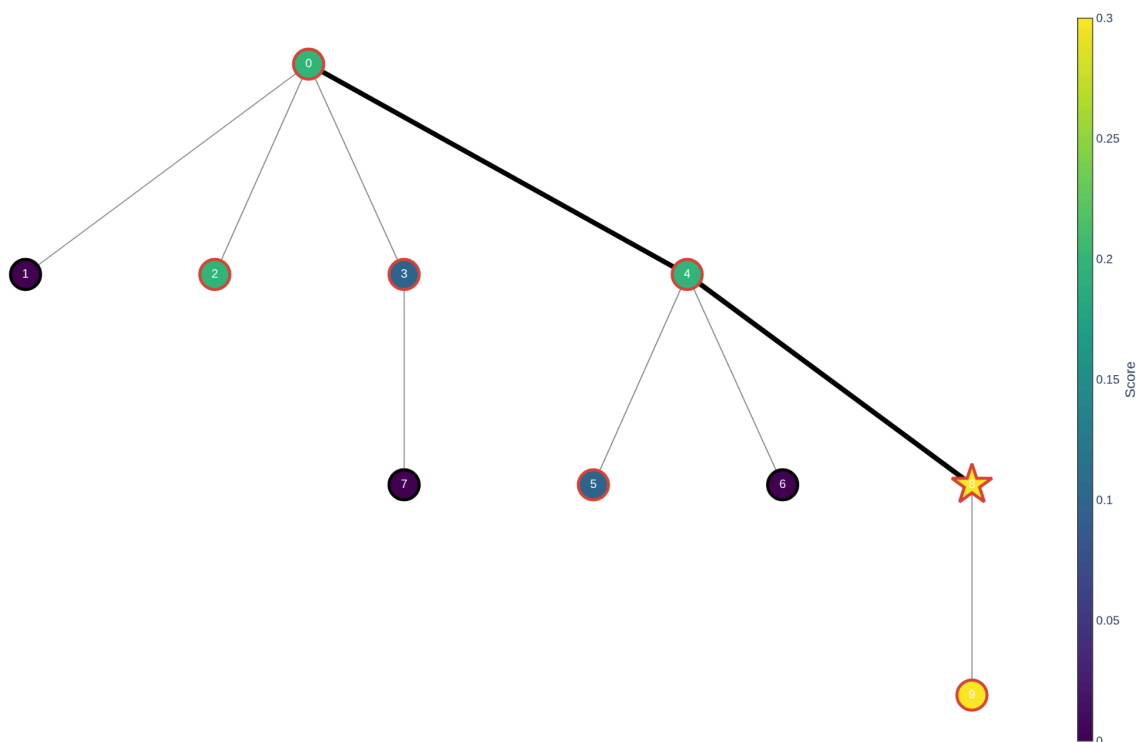
ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

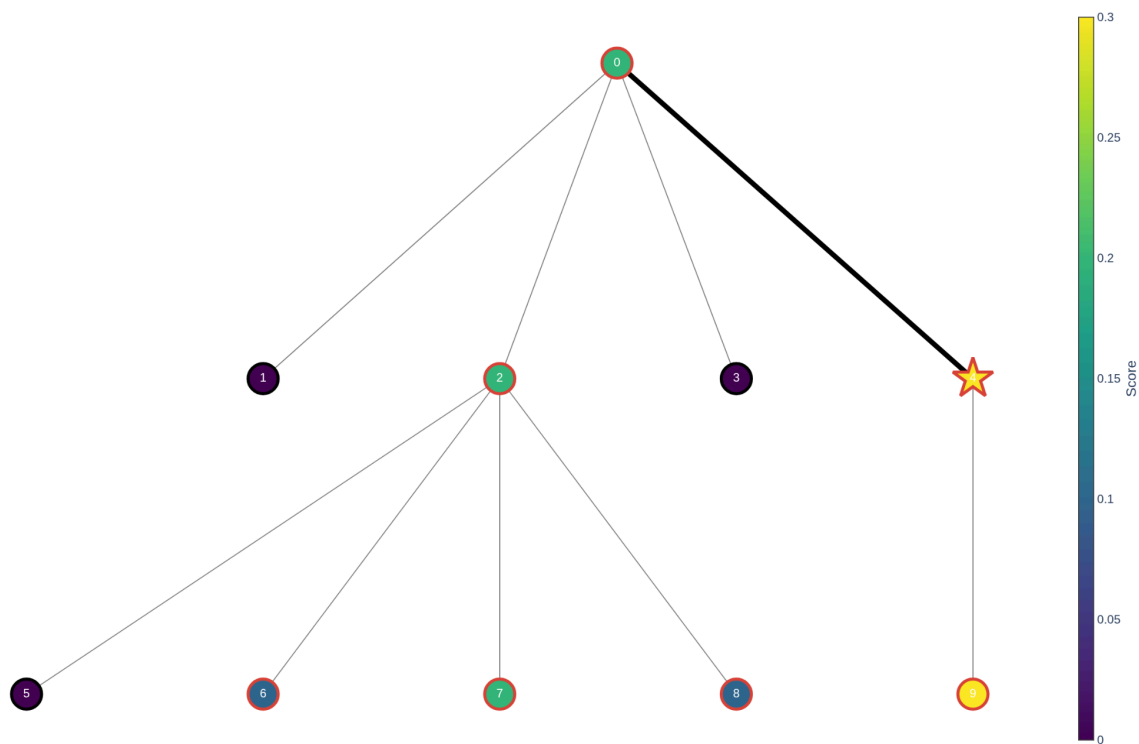
Material Complementar: Resultados da Evolução

Resultados

- **DGM versão original (10 iterações)**



- **DGM versão com bônus de brevidade (10 iterações)**



TOKENS

- **DGM versão original (10 iterações):** 104.872 tokens em média
- **DGM versão com bônus de brevidade (10 iterações):** 98.651 tokens em média

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 23 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Tema: Otimização de Agentes Inteligentes por meio da Computação Evolucionária.

Na Semana passada, propus desenvolver um *framework* de evolução *open-ended* de agentes de automelhoramento próprio, sem restrição ao domínio de código.

Então, nessa Semana eu:

- Estruturei e registrei o *framework* de evolução de agentes criados a partir da demanda e requisitos do usuário, com liberdade para escolha de domínio e objetivos;
- Entendi que não se tratava de *self-improving agents*, já que a aplicação da melhoria em nível de código, nesse cenário, demandava um módulo separado do principal em evolução;
- Criei e defini o *pipeline* evolutivo e os componentes: Agente-Alvo, Agente Criador, Planejador de Otimização, Agente Desenvolvedor e Avaliação e Seleção;
- Criei um diagrama para a representação visual do passo a passo;
- Criei uma POC em código para validação inicial do *framework*;
- Registrei todos os materiais complementares.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Na próxima Semana, pretendo:

- Evoluir a POC a um MVP;
- Avaliar o desempenho no processo de otimização do agente considerando no mínimo um dos requisitos a seguir: *score*, alinhamento ao objetivo, segurança, uso de *tokens*.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

Material Complementar: Framework de Evolução de Agentes (EA)

Proponho um método para a criação e otimização automatizada de agentes inteligentes baseados em LLM, inspirado na evolução darwiniana. O processo é iniciado a partir de um único prompt do usuário, que descreve o agente desejado (o **Agente-Alvo**).

O framework opera da seguinte forma:

1. **Criação Inicial:** Um **Agente Criador** interpreta o prompt do usuário e gera a versão inicial do Agente-Alvo, incluindo seu código, prompts e fluxo de trabalho (workflow).
2. **Avaliação e Aptidão:** O Agente-Alvo é executado em um conjunto de datasets ou benchmarks de referência. Seu desempenho é mensurado e convertido em um "score de aptidão" (fitness score).
3. **Ciclo Evolutivo (Seleção e Mutação):** O processo entra em um ciclo iterativo de evolução. A cada iteração, uma versão do Agente-Alvo é selecionada da população, com maior probabilidade de seleção para aquelas com maior score de aptidão.
4. **Otimização (Mutação):** O agente selecionado passa por um processo de mutação automatizada:
 - Um **Planejador de Otimização** (nó LLM) avalia o desempenho, os erros e a cadeia de pensamento (chain of thought) do agente. Com base nessa análise, ele propõe uma sugestão de melhoria ou correção.
 - Um **Agente Desenvolvedor** recebe essa sugestão e a implementa diretamente no código-fonte do Agente-Alvo, gerando uma nova versão (mutante).
5. **Reavaliação:** A nova versão do agente é então reavaliada no benchmark, recebendo um novo score de aptidão, e é reintroduzida na população para competir no próximo ciclo de seleção.

Este processo iterativo permite que o agente evolua continuamente, adaptando e melhorando suas habilidades para maximizar o desempenho na tarefa solicitada pelo usuário.

Detalhes da Proposta

Para executar o framework, são definidos os seguintes componentes principais:

- **Agente-Alvo:** É a entidade central do processo, representando o agente que o usuário deseja construir. É composto por seu código-fonte, prompts de sistema, ferramentas (tools) que pode utilizar e o fluxo de trabalho (workflow) que define sua lógica operacional. Este agente é o "indivíduo" que sofre a evolução.
- **Agente Criador:** É um agente especializado, responsável pela "gênese" do processo. Ele recebe o prompt inicial do usuário (descrevendo o objetivo do Agente-Alvo) e gera a primeira versão funcional (v0) do seu código e artefatos necessários.
- **Planejador de Otimização:** Atua como o cérebro analítico do processo de mutação. Este nó (baseado em LLM) é invocado após uma avaliação de desempenho. Ele analisa os resultados (ex: logs de erro, pontuações) e a cadeia de pensamento (chain of thought) do Agente-Alvo para identificar falhas ou ineficiências. Seu *output* é uma diretriz clara e acionável de melhoria.
- **Agente Desenvolvedor:** É o "mutador" do sistema. Este agente recebe a diretriz de melhoria do Planejador de Otimização e atua diretamente no código-fonte do Agente-Alvo. Ele edita, refatora ou adiciona código para implementar a sugestão, criando assim uma nova versão (mutante) do agente.
- **Avaliação e Seleção (Datasets & Evolution):** A evolução é guiada por dados. Um ou mais datasets/benchmarks são essenciais para avaliar objetivamente o desempenho de cada versão do Agente-Alvo, gerando um *score* de aptidão. Inspirado na seleção natural, os agentes com maiores *scores* têm maior probabilidade de serem selecionados para o ciclo de "mutação" (passar pelo Planejador e Desenvolvedor), garantindo que as características de sucesso sejam propagadas e refinadas.

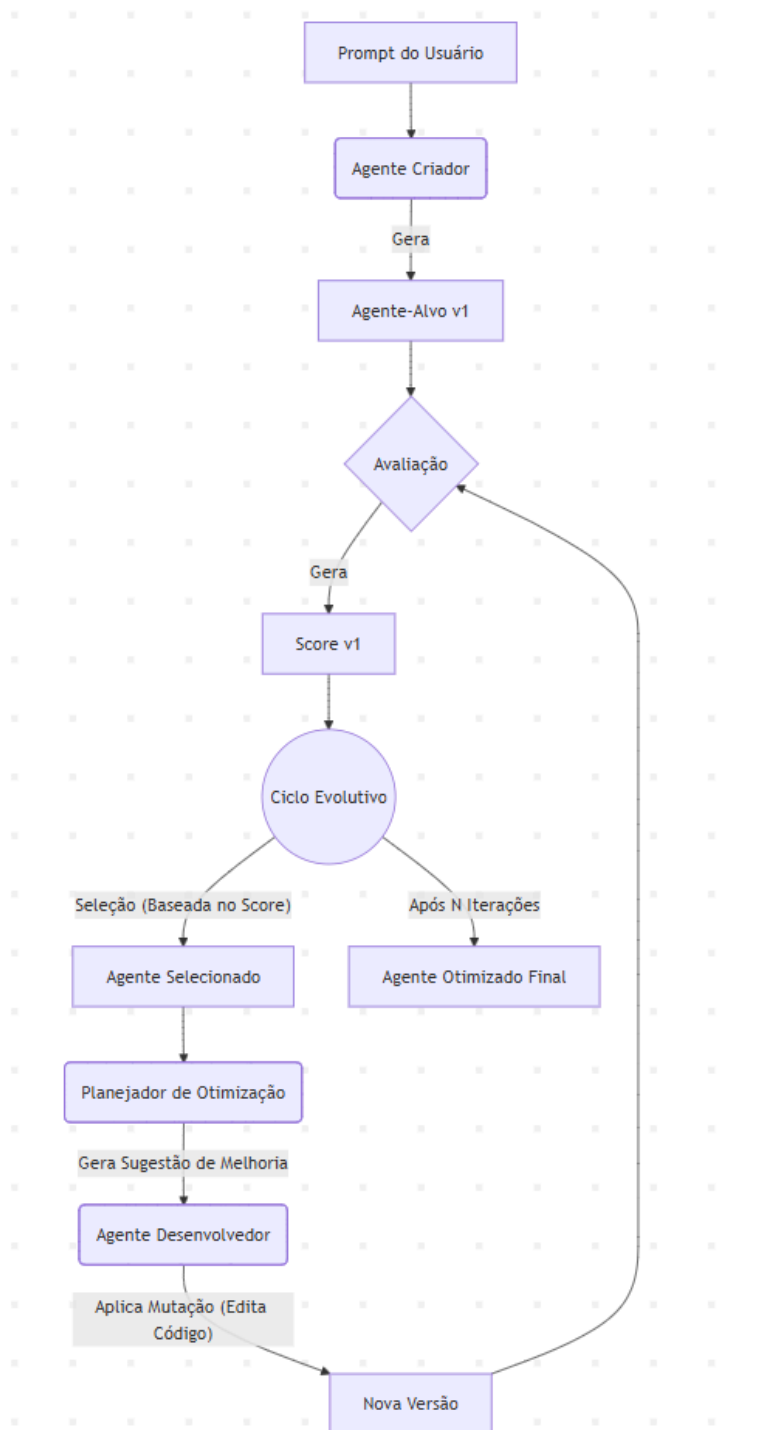



Figura 1: Framework de Evolução de Agentes (EA).

POC (CÓDIGO):  [RESIDÊNCIA]POC.ipynb

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“Gate”) de aprovação: 12 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

André Cerqueira Castro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Tema: Evolução de Agentes Inteligentes

Na Semana passada, propus “evoluir o sistema para, inspirado na computação evolutiva, selecionar nós para evolução a partir da sua aptidão.”

Então, nessa Semana eu:

- Apliquei uma estratégia evolutiva inspirada na seleção natural darwiniana, alcançando um agente funcional com desempenho satisfatório, validando a eficácia da abordagem;
- Nomeei o sistema de **Specialist Agent Scaffold-System (SASS)**, pois ele cria uma base funcional (scaffold) para agentes especialistas. A partir de um objetivo claro e dados de qualidade, o sistema gera e evolui um agente inicial, pronto para ajustes finos conforme micro-objetivos específicos;
 - **Exemplo:** para desenvolver um agente tutor de matemática, o sistema usa um dataset de problemas matemáticos e gera um agente especialista com bom desempenho em resolução. O desenvolvedor então só precisa ajustá-lo ao tom e perfil desejado, otimizando tempo e esforço.
- O sistema foi modularizado e documentado em repositório aberto para facilitar testes, contribuições e avanços colaborativos:
<https://github.com/andrecqcastro/Specialist-Agent-Scaffold-System-SASS->
- O repositório é **parametrizável**, permitindo a troca de datasets, modelos e estratégias de evolução, conforme as necessidades do usuário.
- Para evitar vazamentos e avaliações enganosas (problema comum em abordagens como [Darwin-Godel Machine](#)), implementei divisão entre treino, validação e teste:
 - **Treino:** gera exemplos de acertos/erros para orientar a evolução.
 - **Validação:** calcula scores e seleciona agentes para mutação.
 - **Teste:** avalia o desempenho do melhor agente final em dados inéditos.
- Adicionei um controle adaptativo do parâmetro **lambda**, que ajusta o peso dos melhores scores na seleção de agentes:
 - Se o número de iterações for baixo (ex.: por limitação de recursos), o lambda aumenta para tornar a

seleção mais "gulosa", maximizando ganhos em poucas gerações.

- Usando o dataset **GSM8K** com 250 amostras divididas em treino, validação e teste, e executando **10 iterações evolutivas**, a acurácia do agente subiu de **0.1333 para 0.9333**, utilizando modelos OpenAI distintos para execução e evolução.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Pretendo seguir aprimorando essa solução, tornando-a cada vez mais robusta e eficaz. Meu objetivo é evoluí-la a ponto de superar benchmarks de artigos de referência e compartilhar esses avanços com a comunidade científica.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)