

warning "Atualização de backend (outubro de 2026)" A inferência do GitHub Models foi descontinuada em 30 de julho de 2026. As referências abaixo são mantidas apenas como contexto histórico enquanto este lab é migrado. Use o Lab 013 ? Backends de LLM gratuitos para configurar o Foundry Local ou o Ollama antes de continuar.

Lab 002: Paisagem do Agente de IA 2025

Nível: L50 Caminho: Todos os caminhos Tempo: ~20 min ? Custo: Gratuito ? Nenhuma conta necessária

O Que Você Vai Aprender

O ecossistema de agentes de IA da Microsoft em um relance Quando usar cada plataforma: Copilot Studio, Microsoft Foundry, Semantic Kernel, Teams AI Library, AutoGen Como o MCP se encaixa em todos eles O espectro de no-code a pro-code

Introdução

O ecossistema da Microsoft oferece várias maneiras sobrepostas de construir agentes de IA. Isso pode ser confuso ? você deve usar o Copilot Studio ou o Foundry? Semantic Kernel ou LangChain? MCP ou chamadas diretas de API?

Este laboratório fornece um mapa da paisagem para que você possa tomar decisões informadas.

O Espectro: No-Code ? Pro-Code

Espectro de No-Code a Pro-Code

Não há um "melhor" extremo ? depende do seu caso de uso, habilidades da equipe e requisitos de governança.

Comparação de Plataformas

? GitHub Copilot

|| O que é | Assistente de codificação de IA integrado ao seu IDE e GitHub | Melhor para | Desenvolvedores individuais, produtividade em codificação | Capacidade do agente | Copilot Chat, GitHub Models, Copilot Extensions | Habilidade necessária | Baixa (chat) a Alta (extensões) | Custo | Camada gratuita disponível |

? Copilot Studio (Low-Code)

|| O que é | Construtor de agentes sem código/baixo código da Microsoft | Melhor para | Analistas de negócios, usuários do M365, agentes do Teams | Capacidade do agente | Fluxos de tópicos, conectores, ações personalizadas, Azure OpenAI | Habilidade necessária | Baixa ? nenhuma codificação necessária | Custo | Incluído em muitas licenças do M365; teste gratuito disponível |

? Microsoft Foundry Agent Service

|| O que é | Tempo de execução de agente gerenciado no Azure | Melhor para | Agentes de produção, escala empresarial | Capacidade do agente | Chamada de ferramentas, Code Interpreter, servidores MCP, avaliação | Habilidade necessária | Média ? SDK em Python ou C# | Custo | Assinatura do Azure (camada gratuita para prototipagem) |

? Semantic Kernel

|| O que é | SDK de agente de código aberto (Python / C# / Java) | Melhor para | Desenvolvedores que desejam controle de código com a pilha da Microsoft | Capacidade do agente | Plugins, memória vetorial, planejadores, multi-agente | Habilidade necessária | Média ? experiência em Python ou C# | Custo | Gratuito (código aberto); custos de LLM dependem do backend |

?? AutoGen

|| O que é | Framework multi-agente de código aberto da Microsoft Research | Melhor para | Fluxos de trabalho complexos de múltiplos agentes, pesquisa, orquestração | Capacidade do agente | Conversas aninhadas, humano no loop, execução de código | Habilidade necessária | Alta ? Python, conceitos avançados de agentes | Custo | Gratuito (código aberto); custos de

LLM |

? Teams AI Library

| | O que é | SDK para construir bots do Teams com IA | Melhor para | Aplicativos nativos do Teams, colaboração empresarial | Capacidade do agente | IA conversacional dentro dos canais do Teams, acesso a dados do M365 | Habilidade necessária | Média ? Node.js ou C# | Custo | SDK gratuito; requer locatário do M365 |

question "? Verifique Sua Compreensão" Qual é a principal diferença entre Copilot Studio e Semantic Kernel em termos de quem deve usá-los?

??? success "Resposta" Copilot Studio é projetado para desenvolvedores cidadãos e analistas de negócios que precisam de construção de agentes sem código/baixo código. Semantic Kernel é projetado para desenvolvedores profissionais (Python/C#) que desejam controle total sobre a lógica do agente, plugins e memória.

Onde o MCP Se Encaixa?

Model Context Protocol (MCP) não é uma plataforma ? é um padrão de conector. Pense nisso como o USB-C das ferramentas de IA: uma interface padrão que qualquer agente pode usar para se conectar a qualquer fonte de dados ou ferramenta.

Onde o MCP Se Encaixa

O MCP funciona com todas as plataformas acima ? e também com Claude Desktop, OpenAI e qualquer outro host compatível com MCP.

question "? Verifique Sua Compreensão" O MCP é descrito como "USB-C para ferramentas de IA." Que problema específico essa analogia destaca que o MCP resolve?

??? success "Resposta" O MCP resolve o problema de integração N×M. Sem o MCP, conectar 5 agentes a 5 ferramentas requer 25 integrações personalizadas. Com o MCP como um padrão universal, cada ferramenta publica um servidor MCP e cada agente compatível com MCP pode se conectar a ele ? reduzindo as integrações para N+M.

Folha de Dicas de Decisão

Situação | Ferramenta recomendada | "Quero um agente no Teams para minha equipe, sem codificação" | Copilot Studio | "Quero adicionar IA à minha extensão do VS Code" | VS Code Chat Participant API | "Quero um agente de produção suportado pelo Azure, com monitoramento" | Microsoft Foundry Agent Service | "Quero escrever código em Python/C# para construir um agente sofisticado" | Semantic Kernel | "Quero múltiplos agentes de IA colaborando em tarefas complexas" | AutoGen | "Quero conectar minha ferramenta/API existente a qualquer agente de IA" | Construir um Servidor MCP | "Só quero experimentar com LLMs gratuitamente" | GitHub Models |

question "? Verifique Sua Compreensão" Um desenvolvedor quer construir um sistema onde um agente "pesquisador", um agente "escritor" e um agente "revisor" colaborem na produção de um relatório. Qual ferramenta da Microsoft é mais adequada para isso?

??? success "Resposta" AutoGen. É especificamente projetado para orquestrar múltiplos agentes especializados que colaboram em tarefas complexas por meio de conversas aninhadas. O Semantic Kernel se destaca na construção de agentes sofisticados únicos, enquanto o AutoGen se destaca na coordenação de múltiplos agentes.

? Verificação de Conhecimento

question "Q1 (Múltipla Escolha): Você é um desenvolvedor cidadão sem experiência em codificação. Você precisa construir um chatbot do Teams que responda a perguntas sobre políticas de RH do SharePoint. Qual ferramenta você deve escolher?"

- A) AutoGen - B) Semantic Kernel - C) Copilot Studio - D) Microsoft Foundry Agent Service

??? success "? Revelar Resposta" Correto: C ? Copilot Studio

O Copilot Studio é a opção sem código/baixo código projetada para desenvolvedores cidadãos

e profissionais de TI. Ele se integra nativamente ao Teams e ao Microsoft 365, pode apontar para o SharePoint como uma fonte de conhecimento e não requer código. AutoGen e Semantic Kernel exigem habilidades de desenvolvimento em Python/C#. O Foundry é para desenvolvedores que constroem backends personalizados.

question "Q2 (Múltipla Escolha): O que o MCP (Model Context Protocol) resolve no ecossistema de agentes de IA?"

- A) Ele fornece um construtor GUI para agentes sem codificação - B) Ele otimiza o uso de tokens de LLM para reduzir custos de API - C) Ele define um padrão universal para que qualquer agente possa se conectar a qualquer ferramenta/fonte de dados através de uma interface comum - D) Ele gerencia autenticação e controle de acesso baseado em funções para agentes

??? success ""? Revelar Resposta" Correto: C

O MCP é descrito como "USB-C para ferramentas de IA" ? um padrão de plug universal. Sem o MCP, conectar 5 agentes a 5 ferramentas requer 25 integrações personalizadas. Com o MCP, cada ferramenta publica um servidor MCP e cada agente compatível com MCP pode usá-lo. Ele resolve o problema de integração N×M em todo o ecossistema.

question "Q3 (Múltipla Escolha): Qual é a principal diferença entre Semantic Kernel e AutoGen?"

- A) Semantic Kernel é de código aberto; AutoGen é proprietário da Microsoft - B) Semantic Kernel constrói agentes sofisticados únicos com plugins; AutoGen orquestra múltiplos agentes especializados colaborando em tarefas complexas - C) AutoGen funciona apenas com GPT-4o; Semantic Kernel suporta qualquer LLM - D) Semantic Kernel é apenas para Python; AutoGen suporta Python e C#

??? success ""? Revelar Resposta" Correto: B

Semantic Kernel se destaca na construção de um agente profundamente capaz ? com plugins, memória, planejadores e uso estruturado de ferramentas. AutoGen se destaca na orquestração de \*múltiplos\* agentes ? um agente pesquisador, um agente escritor, um agente revisor ? cada um realizando uma subtarefa especializada e passando resultados entre eles. Ambos são de código aberto e suportam múltiplos LLMs.

Resumo

O ecossistema da Microsoft possui ferramentas para todos os níveis de habilidade e casos de uso ? desde o Copilot Studio sem código até o AutoGen pro-code. O MCP é o conector universal que funciona em todas elas. No próximo laboratório, ajudaremos você a escolher a ferramenta certa para sua situação específica.

Próximos Passos

? Lab 003: Escolhendo a Ferramenta Certa

Modernização da Fase 3

warning "Verified 2026-10-27" Use Microsoft Agent Framework 1.0 em novos agentes. Semantic Kernel fica como origem legada de migração; AutoGen está orientado a manutenção. Remova planners obsoletos e use workflows, ferramentas, memória, checkpoints e parada explícitos.

[diagram or code sample]

Microsoft Agent Framework 1.0 GA AutoGen to Agent Framework migration

Kit de apoio

O build do site gera downloads estáveis em PDF e ZIP. O ZIP contém o guia do lab e todos os arquivos de apoio declarados para este lab.

Baixar PDF gerado Baixar ZIP gerado

Supporting file: CHEATSHEET.md

Arquivos de apoio: Lab 002: Paisagem do Agente de IA 2025

Comece pelo caminho local ou gratuito documentado. Registre premissas e evidências de validação. Mantenha serviços hospedados opcionais, salvo quando o lab os exigir.

Supporting file: README.md

Lab 002: Paisagem do Agente de IA 2025

Use estes arquivos com o guia do lab e o kit gerado.