

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 003: Escolhendo a Ferramenta Certa

Nível: L50 Trilha: Todas as trilhas Tempo: ~15 min ? Custo: Gratuito ? Nenhuma conta necessária

O Que Você Vai Aprender

Um framework prático de decisão para escolher sua ferramenta de agente de IA Compreensão dos principais trade-offs (controle vs. simplicidade, custo vs. poder) Rotas de aprendizado sugeridas com base no seu papel e objetivos

Introdução

Após revisar o panorama no Lab 002, a pergunta natural é: por onde devo começar?

Use o fluxograma de decisão e os guias por perfil abaixo para encontrar seu caminho.

Fluxograma de Decisão

Fluxograma de Decisão

question "? Verifique Seu Entendimento" De acordo com o fluxograma de decisão, qual ferramenta você deve usar se seu objetivo principal é conectar um banco de dados ou API existente a agentes de IA?

??? success "Resposta" Você deve construir um MCP Server. O MCP (Model Context Protocol) fornece um padrão de conector universal para que qualquer agente de IA compatível com MCP possa acessar sua ferramenta ou fonte de dados por meio de uma interface comum.

Por Perfil

? Analista de Negócios / Usuário Avançado Objetivo: Automatizar fluxos de trabalho, criar agentes sem escrever código.

Trilha recomendada: 1. Lab 001 ? Lab 011 ? Lab 069 ? Lab 075

Ferramentas: Copilot Studio, Declarative Agents, Power BI Copilot, M365 Copilot

??? Desenvolvedor (Python / C#) Objetivo: Escrever agentes em código, integrar com sistemas existentes.

Trilha recomendada: 1. Lab 013 ? Lab 076 ? Lab 020 ? Lab 082 ? Lab 084

Ferramentas: Agent Framework (SK), MCP, Guardrails, GitHub Models

? Engenheiro de Integração / Plataforma Objetivo: Expor sistemas existentes (bancos de dados, APIs) para agentes de IA.

Trilha recomendada: 1. Lab 012 ? Lab 020 ? Lab 031 ? Lab 054 ? Lab 064

Ferramentas: MCP, A2A Protocol, pgvector, Azure API Management

?? Arquiteto de Soluções Objetivo: Projetar sistemas multiagentes em produção com governança e observabilidade.

Trilha recomendada: 1. Lab 076 ? Lab 049 ? Lab 050 ? Lab 074 ? Lab 084

Ferramentas: Agent Framework, Foundry Agent Service, OpenTelemetry, A2A + MCP

? Engenheiro / Analista de Dados Objetivo: Construir análises impulsionadas por IA, agentes de dados e pipelines de enriquecimento.

Trilha recomendada: 1. Lab 047 ? Lab 052 ? Lab 053 ? Lab 067 ? Lab 075

Ferramentas: Fabric IQ, Work IQ, GraphRAG, Power BI Copilot

? Administrador Corporativo / Governança de TI Objetivo: Governar, proteger e monitorar implantações de agentes de IA em toda a organização.

Trilha recomendada: 1. Lab 063 ? Lab 065 ? Lab 066 ? Lab 064 ? Lab 046

Ferramentas: Entra ID, Purview DSPM, Copilot Studio Governance, APIM, Agent 365

? Estudante / Aprendiz Objetivo: Entender agentes de IA e construir algo real,

gratuitamente.

Trilha recomendada: 1. Lab 001 ? Lab 004 ? Lab 013 ? Lab 078 ? Lab 076 ? Lab 022

Ferramentas: GitHub Models, Foundry Local, Agent Framework ? tudo gratuito!

question "? Verifique Seu Entendimento" Um arquiteto de soluções precisa projetar um sistema multiagentes em produção com observabilidade e governança. Qual combinação de ferramentas este lab recomenda?

??? success "Resposta" Foundry, Semantic Kernel, AutoGen e App Insights. A trilha de aprendizado recomendada é: Foundry Agent MCP ? Agent Observability ? Multi-Agent SK ? AutoGen Multi-Agent. Isso cobre runtime gerenciado, lógica de agentes, orquestração multiagentes e monitoramento.

question "? Verifique Seu Entendimento" O que significa "mais controle = mais responsabilidade" no trade-off controle vs. simplicidade?

??? success "Resposta" Ferramentas pro-code como AutoGen e Semantic Kernel oferecem total flexibilidade sobre a lógica do agente, mas você precisa lidar com mais coisas ? tratamento de erros, implantação, segurança, escalabilidade. Ferramentas no-code como Copilot Studio são mais rápidas de construir, mas menos personalizáveis. A escolha certa depende das habilidades e requisitos da sua equipe.

Os Dois Principais Trade-offs

Controle vs Simplicidade, Gratuito vs Pago

Mais controle = mais flexibilidade + mais responsabilidade. Mais simplicidade = mais rápido de construir + menos personalizável.

question "? Verifique Seu Entendimento" Um estudante sem assinatura do Azure e sem orçamento ainda pode construir um agente de IA funcional usando as ferramentas deste hub?

??? success "Resposta" Sim! GitHub Models e Semantic Kernel são completamente gratuitos. Os labs conceituais L50 e os labs L100?L200 usando GitHub Models não exigem assinatura do Azure. Estudantes podem construir agentes reais, executar MCP servers localmente e aprender todo o ciclo de desenvolvimento de agentes a custo zero.

2. Gratuito vs. Pago

O SVG acima inclui a comparação completa entre Gratuito vs. Pago. Comece gratuito ? adicione o Azure apenas quando precisar de recursos de produção.

? Teste de Conhecimento

question "Q1 (Múltipla Escolha): Um desenvolvedor deseja construir uma extensão do VS Code que responda a @mybot no GitHub Copilot Chat. Qual ferramenta/API ele deve usar?"

- A) Copilot Studio - B) VS Code Chat Participant API (Lab 025) - C) Microsoft Foundry Agent Service - D) Azure Bot Service

??? success "? Revelar Resposta" Correta: B ? VS Code Chat Participant API

A Chat Participant API registra um participante @yourextension diretamente na interface do Copilot Chat do VS Code. Ela roda inteiramente dentro do VS Code ? sem assinatura do Azure, sem servidor necessário. O Copilot Studio é para agentes sem código no Teams/M365. O Foundry é para agentes hospedados no lado do servidor com escala completa na nuvem.

question "Q2 (Múltipla Escolha): Qual fator é o MAIS importante ao escolher entre Copilot Studio e Semantic Kernel?"

- A) A linguagem de programação que você prefere (Python vs C#) - B) Se você precisa de implantação na nuvem ou implantação local - C) Seu perfil e quanto controle de código você precisa ? desenvolvedor cidadão vs. desenvolvedor profissional - D) O provedor de LLM (OpenAI vs Anthropic)

??? success "? Revelar Resposta" Correta: C

O eixo de decisão principal é controle de código vs. velocidade. O Copilot Studio é voltado

para desenvolvedores cidadãos e profissionais de TI que precisam de um agente funcional rapidamente, sem código. O Semantic Kernel é voltado para desenvolvedores profissionais que precisam de controle total sobre lógica, esquemas de ferramentas, padrões de memória e comportamento em produção. Ambos suportam múltiplos LLMs e implantação na nuvem.

question "Q3 (Múltipla Escolha): O princípio do 'menor privilégio' diz que seu agente deve ter acesso exatamente ao que precisa ? nada mais. Qual destas opções viola o menor privilégio?"

- A) Um agente de busca de produtos que pode chamar `search_products()` e `get_product_details()` - B) Um agente de atendimento ao cliente com acesso somente leitura ao banco de dados - C) Um agente de status de pedido com credenciais de administrador completas para o banco de dados de pedidos - D) Um agente de clima que só pode chamar a API pública de clima

??? success ""? Revelar Resposta" Correta: C ? Credenciais de administrador completas violam o menor privilégio

Um agente de status de pedido só precisa *ler* registros de pedidos. Dar a ele credenciais de administrador significa que um ataque de injeção de prompt ou erro de lógica poderia excluir pedidos, modificar preços ou acessar todos os dados de clientes. A configuração correta é um usuário de banco de dados somente leitura com escopo nas tabelas específicas que o agente precisa. As opções A, B e D seguem o menor privilégio corretamente.

Resumo

Não existe uma única ferramenta "certa" ? depende do seu perfil, objetivos e restrições. A boa notícia: tudo neste hub começa gratuito, e você sempre pode evoluir. O framework de decisão acima aponta o caminho mais eficiente para você.

Próximos Passos

Escolha sua trilha e mergulhe!

Rota sem código: ? Lab 011 ? Copilot Studio Rota para desenvolvedores (gratuita): ? Lab 013 ? GitHub Models Rota MCP: ? Lab 012 ? What is MCP? Rota Azure completa: ? Lab 030 ? Foundry + MCP Quer entender LLMs primeiro? ? Lab 004 ? How LLMs Work

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 003: Escolhendo a Ferramenta Certa

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 003: Escolhendo a Ferramenta Certa

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