

Lab 076: Microsoft Agent Framework ? De SK para MAF

Nível: L200 Trilha: Todas as trilhas Tempo: ~75 min ? Custo: Gratuito ? Usa dados simulados de migração

O que Você Vai Aprender

Como Semantic Kernel e AutoGen são unificados no Microsoft Agent Framework (MAF) O que são Agent Skills (pacotes portáteis de habilidades em .md) e como eles permitem reutilização Como Graph Workflows (DAG com checkpointing) substituem pipelines lineares Analisar uma matriz de migração comparando 15 funcionalidades entre SK, AutoGen e MAF Identificar níveis de esforço de migração e capacidades exclusivas do MAF

Introdução

O Microsoft Agent Framework (MAF) unifica Semantic Kernel e AutoGen em uma única plataforma coesa para construção de agentes de IA. Lançado como Release Candidate em fevereiro de 2026, o MAF reúne o melhor dos dois mundos:

O sistema de plugins, planejadores e conectores empresariais do Semantic Kernel As conversas multi-agente, execução de código e padrões de chat em grupo do AutoGen

Instalação

[diagram or code sample]

Conceitos-Chave

Conceito | Descrição | Agent Skills | Pacotes portáteis de habilidades em .md que definem capacidades do agente, entradas, saídas e dependências ? compartilháveis entre equipes e projetos | Graph Workflows | Orquestração baseada em DAG com checkpointing, retry e ramificação ? substituindo pipelines lineares | DevUI | Interface de desenvolvimento integrada para depuração de conversas de agentes, inspeção de execução de habilidades e visualização de workflows | API Unificada | Superfície de API única que substitui o Kernel do SK e o AssistantAgent do AutoGen por uma classe Agent comum |

O Cenário

Você é um Engenheiro de Plataforma em uma empresa que construiu agentes usando tanto Semantic Kernel quanto AutoGen. A liderança decidiu migrar para o MAF. Você tem uma matriz de migração (migration_matrix.csv) que mapeia 15 funcionalidades nos três frameworks ? rastreando disponibilidade, esforço de migração e funcionalidades exclusivas do MAF. Seu trabalho: analisar a matriz, identificar ganhos rápidos, sinalizar desafios e construir um plano de migração.

info "Dados Simulados" Este laboratório usa um CSV de matriz de migração simulado. Os dados refletem o mapeamento real de funcionalidades entre Semantic Kernel, AutoGen e MAF conforme documentado nos guias de migração.

Pré-requisitos

Requisito | Por quê | Python 3.10+ | Executar os scripts de análise | Biblioteca pandas | Manipulação de dados |

[diagram or code sample]

tip "Início Rápido com GitHub Codespaces" Open in GitHub Codespaces

Todas as dependências estão pré-instaladas no devcontainer.

? Arquivos de Apoio

note "Baixe estes arquivos antes de iniciar o laboratório" Salve todos os arquivos em uma pasta lab-076/ no seu diretório de trabalho.

Arquivo | Descrição | Download | broken_migration.py | Exercício de correção de bugs (3 bugs + autotestes) | ? Download | migration_matrix.csv | Comparação de 15 funcionalidades: SK vs AutoGen vs MAF | ? Download |

Etapa 1: Entenda a Matriz de Migração

A matriz de migração mapeia 15 funcionalidades em três frameworks. Cada linha representa uma capacidade:

Coluna | Descrição | feature | A capacidade sendo comparada (ex.: plugins, multi_agent_chat) | sk_support | Se Semantic Kernel suporta esta funcionalidade: yes, partial ou no | autogen_support | Se AutoGen suporta esta funcionalidade: yes, partial ou no | maf_support | Se MAF suporta esta funcionalidade: yes, partial ou no | migration_effort | Esforço para migrar de SK/AutoGen para MAF: low, medium ou high | category | Categoria da funcionalidade: core, orchestration, tooling ou integration |

Etapa 2: Carregue e Explore a Matriz

[diagram or code sample]

Saída esperada:

[diagram or code sample]

Etapa 3: Identifique Ganhos Rápidos (Baixo Esforço de Migração)

Funcionalidades com esforço de migração low são seus ganhos rápidos ? comece por aqui:

[diagram or code sample]

tip "Estratégia de Migração" Comece pelas funcionalidades de baixo esforço para construir a confiança da equipe e demonstrar a API unificada do MAF. Estas geralmente possuem equivalentes diretos no SK ou AutoGen, tornando a migração simples.

Etapa 4: Encontre Funcionalidades Exclusivas do MAF

Quais funcionalidades existem no MAF mas não no Semantic Kernel?

[diagram or code sample]

[diagram or code sample]

Etapa 5: Analise as Migrações de Alto Esforço

Estas funcionalidades precisam de mais planejamento:

[diagram or code sample]

warning "Risco de Migração" Funcionalidades de alto esforço frequentemente envolvem mudanças arquiteturais ? por exemplo, substituir orquestração customizada por Graph Workflows ou converter plugins proprietários em Agent Skills. Planeje de 2 a 4 semanas por funcionalidade de alto esforço.

Etapa 6: Construa o Plano de Migração

[diagram or code sample]

? Exercício de Correção de Bugs

O arquivo lab-076/broken_migration.py contém 3 bugs que produzem análise de migração incorreta. Você consegue encontrar e corrigir todos?

Execute os autotestes para ver quais falham:

[diagram or code sample]

Você deverá ver 3 testes falhando. Cada teste corresponde a um bug:

Teste | O que verifica | Dica | Teste 1 | Contagem de funcionalidades de baixo esforço | Deve filtrar migration_effort == "low", não "high" | Teste 2 | Contagem de MAF-only vs SK | Deve verificar sk_support == "no", não "yes" | Teste 3 | Contagem total de funcionalidades | Deve usar len(df), não len(df.columns) |

Corrija todos os 3 bugs e execute novamente. Quando você ver All passed!, está pronto!

? Verificação de Conhecimento

question "Q1 (Múltipla Escolha): O que é o Microsoft Agent Framework (MAF)?"

- A) Uma nova versão do AutoGen com um nome diferente - B) Um framework unificado que combina Semantic Kernel e AutoGen em uma única plataforma - C) Um serviço exclusivo de nuvem para executar agentes no Azure - D) Um substituto para o LangChain

??? success "? Revelar Resposta" Correto: B) Um framework unificado que combina Semantic

Kernel e AutoGen em uma única plataforma

O MAF une os pontos fortes de ambos os frameworks: o sistema de plugins empresariais e planejadores do SK com as conversas multi-agente e execução de código do AutoGen. Ele fornece uma única classe Agent, Agent Skills portáteis e Graph Workflows baseados em DAG.

question "Q2 (Múltipla Escolha): O que são Agent Skills no MAF?"

- A) Funções Python decoradas com @skill - B) Pacotes portáteis de habilidades em .md que definem capacidades, entradas, saídas e dependências - C) Pesos de modelo pré-treinados para tarefas específicas - D) Azure Functions que os agentes podem chamar

??? success "?" Revelar Resposta" Correto: B) Pacotes portáteis de habilidades em .md que definem capacidades, entradas, saídas e dependências

Agent Skills são pacotes baseados em markdown que descrevem o que um agente pode fazer, quais entradas ele precisa, quais saídas ele produz e quais dependências são necessárias. Eles são compartilháveis entre equipes e projetos, permitindo um marketplace de capacidades reutilizáveis de agentes.

question "Q3 (Execute o Laboratório): Quantas funcionalidades têm esforço de migração 'low'?"

Execute a análise da Etapa 3 no ? migration_matrix.csv e conte as funcionalidades de baixo esforço.

??? success "?" Revelar Resposta" 7 funcionalidades

Das 15 funcionalidades na matriz de migração, 7 têm migration_effort = "low". Estes são os ganhos rápidos para a Fase 1 da migração ? geralmente funcionalidades com equivalentes diretos entre SK/AutoGen e MAF.

question "Q4 (Execute o Laboratório): Quantas funcionalidades estão disponíveis no MAF mas NÃO no Semantic Kernel?"

Execute a análise da Etapa 4 para encontrar funcionalidades onde maf_support = "yes" e sk_support = "no".

??? success "?" Revelar Resposta" A contagem de funcionalidades onde MAF tem suporte yes mas SK tem suporte no.

Estas representam capacidades que as equipes ganham ao migrar do SK para o MAF ? como padrões de chat multi-agente, sandboxes de execução de código e outras funcionalidades que originaram no AutoGen e agora estão disponíveis no framework unificado.

question "Q5 (Execute o Laboratório): Quantas funcionalidades no total são rastreadas na matriz de migração?"

Carregue o CSV e verifique a contagem total de linhas.

??? success "?" Revelar Resposta" 15 funcionalidades

A matriz de migração rastreia 15 funcionalidades em 4 categorias: core (5), orchestration (4), tooling (3) e integration (3). Cada funcionalidade é avaliada quanto ao suporte em SK, AutoGen e MAF, junto com o esforço de migração.

Resumo

Tópico | O que Você Aprendeu | Microsoft Agent Framework | Plataforma unificada que une Semantic Kernel e AutoGen (RC Fev 2026) | Agent Skills | Pacotes portáteis de habilidades em .md para capacidades reutilizáveis de agentes | Graph Workflows | Orquestração baseada em DAG com checkpointing, substituindo pipelines lineares | Matriz de Migração | 15 funcionalidades comparadas entre SK, AutoGen e MAF | Estratégia de Migração | Comece pelo baixo esforço (7 funcionalidades), planeje para alto esforço (3 funcionalidades) | DevUI | Interface de desenvolvimento integrada para depuração e visualização de workflows de agentes |

Próximos Passos

Lab 034 ? Multi-Agente com Semantic Kernel (entenda de onde você está migrando) Lab 036 ? Fundamentos do AutoGen (entenda os padrões do AutoGen antes do MAF) Lab 074 ? Foundry Agent Service (opere e monitore frotas de agentes em produção) Lab 073 ? SWE-Bench (avaliar agentes MAF em tarefas de codificação do mundo real)

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 paradas explícitas.

[diagram or code sample]

Microsoft Agent Framework 1.0 GA AutoGen to Agent Framework migration

Phase 3 · Simulação / Caminho Real

Simulação Caminho Real

Preserve the existing free simulation and its fixtures. Estimated cost: US\$0.

Reuse the same acceptance criteria against the documented hosted service. Before running, record a service-specific estimate and a hard spending cap. After validation: Excluir implantações, endpoints, índices, traces e dados de teste após a validação.

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: broken_migration.py

```
/usr/bin/env python3 """Bug-fix: Broken Migration Analyzer - Lab 076.""" import io, pandas as pd def count_low_effort(matrix): return (matrix["migration_effort"] == "high").sum() def features_in_maf_only(matrix): return len(matrix[matrix["semantic_kernel"] == "not supported"]) def migration_coverage(matrix): return matrix["agent_framework"].nunique() / len(matrix) * 100 def run_tests(): csv="feature,semantic_kernel,agent_framework,migration_effort\nPlugins,KernelPlugin,AgentSkills,medium\nA2A,not supported,native A2A,n/a\nStreaming,IAsyncEnumerable,built-in,low\nDevUI,not available,DevUI,low" m=pd.read_csv(io.StringIO(csv));p,f=0,0 n=count_low_effort(m);exp=2 if n==exp:print(f"OK T1: {n}");p+=1 else:print(f"FAIL T1: {n} (exp {exp})");f+=1 a=features_in_maf_only(m);exp=1 if a==exp:print(f"OK T2: {a}");p+=1 else:print(f"FAIL T2: {a} (exp {exp})");f+=1 c=migration_coverage(m);exp=100.0 if abs(c-exp)<0.1:print(f"OK T3: {c:.0f}%");p+=1 else:print(f"FAIL T3: {c:.0f}% (exp {exp:.0f}%)");f+=1 print(f"\n{'All passed!' if f==0 else f'{f} failed'}") if __name__=="__main__":print("Tests:\n");run_tests()
```

Supporting file: migration_matrix.csv

feature,semantic_kernel,autogen,agent_framework,migration_effort Agent Abstraction,ChatCompletionAgent,AssistantAgent,Agent (unified),low Plugin System,KernelPlugin,Tools,AgentSkills (.md files),medium Multi-Agent,AgentGroupChat,GroupChat / Swarm,Graph Workflows,medium Orchestration>manual loop,round_robin / selector,DAG with data flows,high Streaming,IAsyncEnumerable / AsyncGenerator,streaming parameter,built-in with checkpointing,low Memory,SemanticTextMemory,ConversableAgent memory,pluggable memory providers,medium Human-in-the-Loop>manual implementation,human_input_mode,built-in graph nodes,low Observability,OpenTelemetry manual,logging,DevUI + OpenTelemetry built-in,low MCP Integration,via plugin,via tools,native MCP support,low A2A Integration,not supported,not

supported,native A2A support,n/a AG-UI Integration,not supported,not supported,native AG-UI support,n/a Model Support,OpenAI/Azure/Ollama,OpenAI/Azure,OpenAI/Azure/Anthropic/GitHub/Local,low Language Support,.NET and Python,Python (.NET experimental),.NET and Python (stable both),low State Management>manual>manual>checkpointing + time-travel,medium DevUI,not available,AutoGen Studio,DevUI (built-in),low