

Lab 087: Copilot + MCP: GitHub MCP Server, Playwright e Seu Próprio Servidor

Level: L200 Jornada: copilot Tempo: ~60 min Custo: GitHub Free

info "Desenho gratuito primeiro" Conclua o exercício com Copilot Free ou com a simulação incluída. Recursos pagos são opcionais e marcados claramente.

tip "Impacto em AI Credits" Baixo no exercício guiado; médio/alto em execuções agênticas ou repetidas. Prefira seleção automática de modelo, planeje antes de agir, mantenha o contexto focado e evite sessões paralelas desnecessárias.

O que Você Aprenderá

Conectar ferramentas MCP aprovadas ao Copilot Comparar servidores GitHub, Playwright e local customizado Aplicar allowlists e indicações de somente leitura Projetar um contrato mínimo de ferramenta

Alinhamento GH-300

Use Copilot features (25?30%) · Data and architecture (10?15%). Este mapeamento segue o recorte curricular GH-300 de 07/08/2026. O escopo oficial pode manter termos antigos como ?Copilot Edits?; este lab ensina a experiência atual e sinaliza a diferença.

Mapa Conceitual

[diagram or code sample]

Exercício Prático

Etapa 1: Inventarie os servidores

Liste servidor, transporte, credenciais, limite de dados e ferramentas antes de conectar.

Prefira o GitHub MCP Registry ou um servidor cujo código você possa inspecionar.

Etapa 2: Execute uma tarefa GitHub somente leitura

Peça ao Copilot para listar issues ou inspecionar metadados sem alterar estado. Confirme cada invocação antes de ampliar permissões.

Etapa 3: Execute uma verificação Playwright local

Use Playwright MCP somente em localhost. Registre uma observação de acessibilidade e nunca envie dados privados a servidor não revisado.

Etapa 4: Reutilize o Lab 020

Exponha uma ferramenta local de consulta OutdoorGear. Adicione schema estrito, saída limitada, erros explícitos e allowlist.

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: mcp.json

```
{ "servers": { "outdoorgear": { "command": "python", "args": [ "server.py" ] } } }
```

Supporting file: mcp_allowlist.json

```
{ "allowed_servers": [ "outdoorgear" ], "allowed_tools": [ "catalog_search" ] }
```

Supporting file: threat_checklist.csv

threat,control,evidence untrusted-tool-output,schema validation,negative test
excess-permission,allowlist,approved server list