

Lab 088: Copilot CLI: o Agente no Terminal

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

Instalar e autenticar o Copilot CLI atual Usar fluxos interativos e programáticos

Configurar agentes, hooks, MCP e aprovações Aplicar verificações de segurança e política

Alinhamento GH-300

Use Copilot features (25?30%). 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: Instale o CLI atual

Use o pacote oficial (npm install -g @github/copilot), WinGet, Homebrew ou o binário. A extensão aposentada gh copilot não é este produto.

Etapa 2: Comece com um prompt somente leitura

Peça ao CLI para explicar o repositório e propor um comando de validação. Inspecione o plano e cada permissão solicitada.

Etapa 3: Adicione automação controlada

Crie um agente de qualidade documental, um hook que bloqueie comandos que imprimem segredos e uma ferramenta MCP somente leitura.

Etapa 4: Exercite o modo programático

Produza saída legível por máquina para uma tarefa limitada. Falhe se houver escrita não aprovada ou comando destrutivo.

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: .github/agents/cli.agent.md

Plan briefly, make the smallest complete change, run the relevant tests, and summarize evidence.

Supporting file: .github/hooks/pre-command.json

```
{ "hook": "pre-command", "deny": [ "rm -rf", "git reset --hard" ], "require_approval": [ "deploy", "delete" ] }
```

Supporting file: commands.csv

command	purpose	approval	copilot -p	bounded prompt	not required	copilot --allow-tool	tool execution	required
---------	---------	----------	------------	----------------	--------------	----------------------	----------------	----------

Supporting file: scripts/session_summary.ps1

```
param([string]$Session = "workshop") Write-Output "Session: $Session"
```