

Servidor MCP em .NET

Como construir e consumir

Tiago Aguiar

Desenvolvedor .NET Sênior





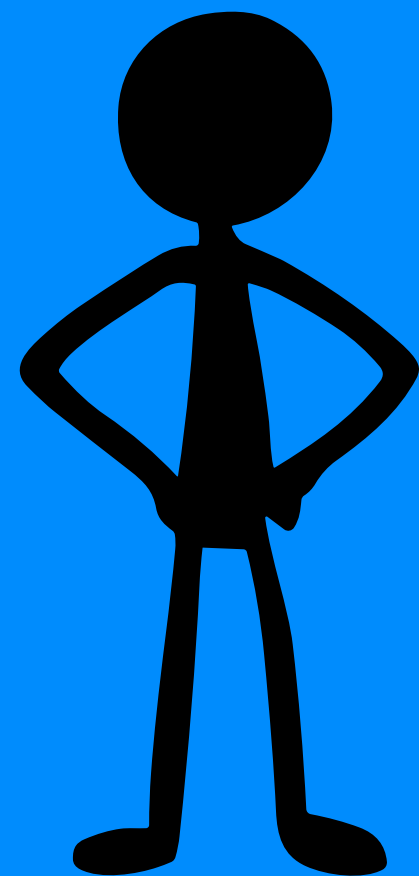
- +14 Anos na Área de Tecnologia
- Desenvolvedor .NET
- Back-end
- Canal AguiarDev no YouTube
- Podcast AguiarDev Talks
- Comunidade no Discord



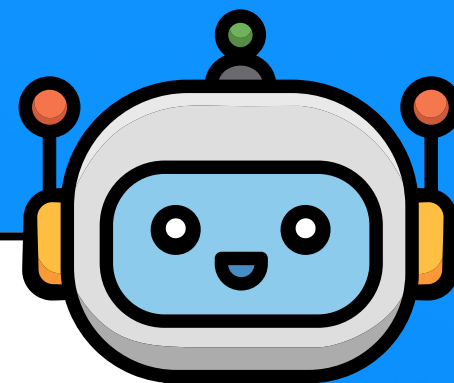
Evolução da Interação Humano x IA

Estágio 1

Chat

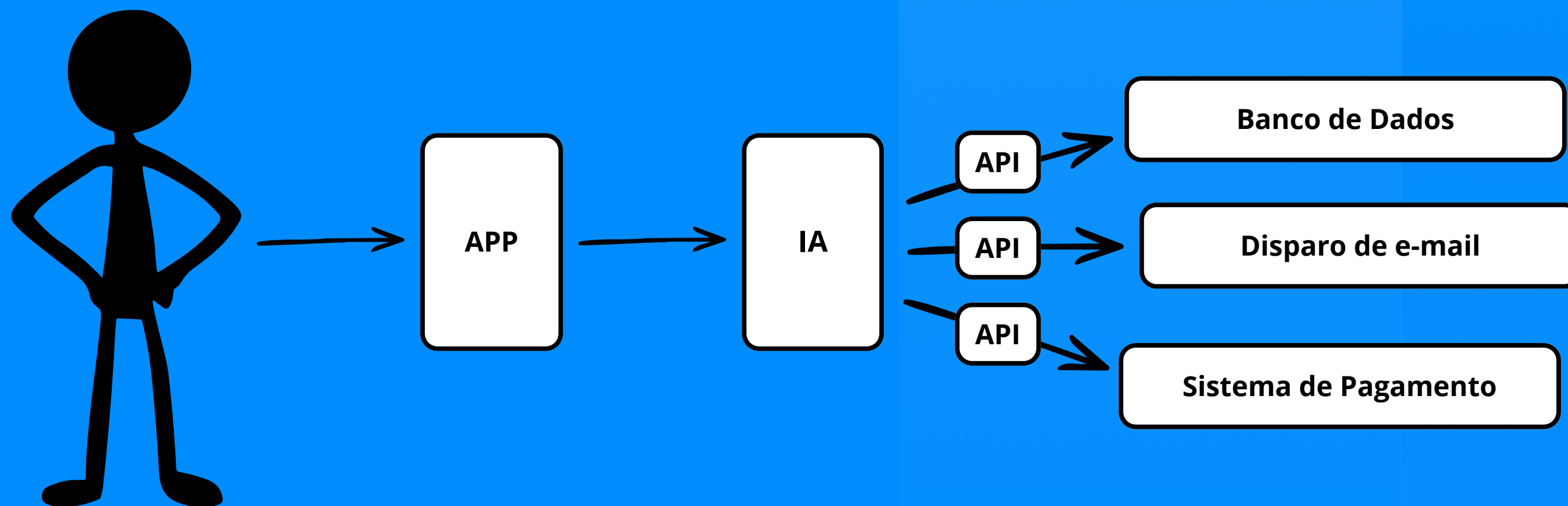


- Copilot
- ChatGPT
- Claude
- Gemini



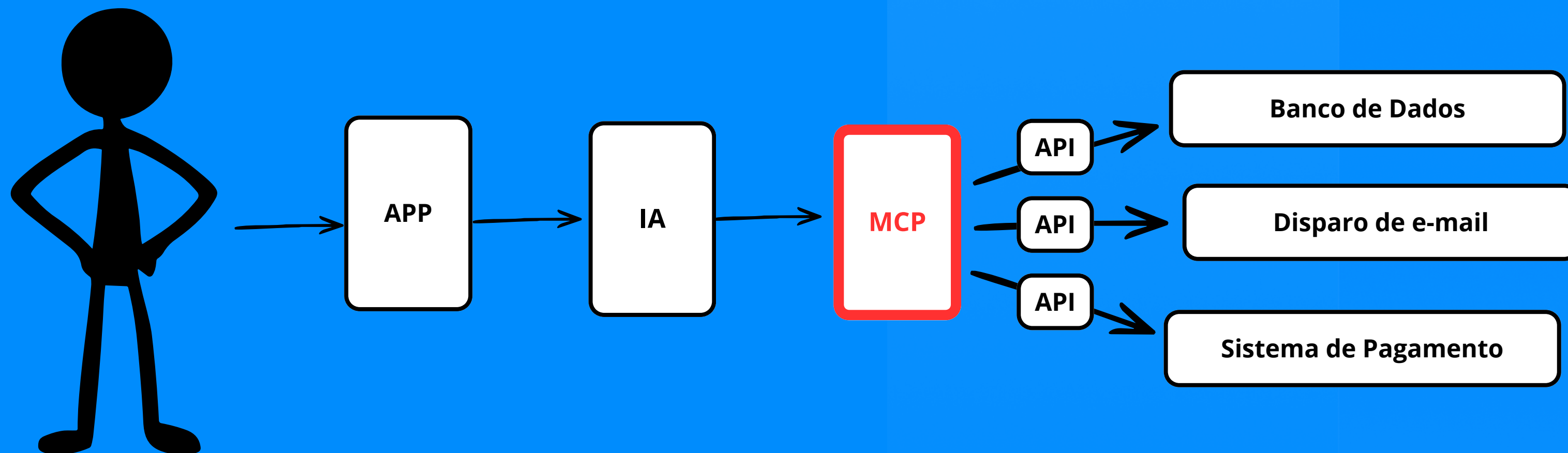
Estágio 2

API



Estágio 3

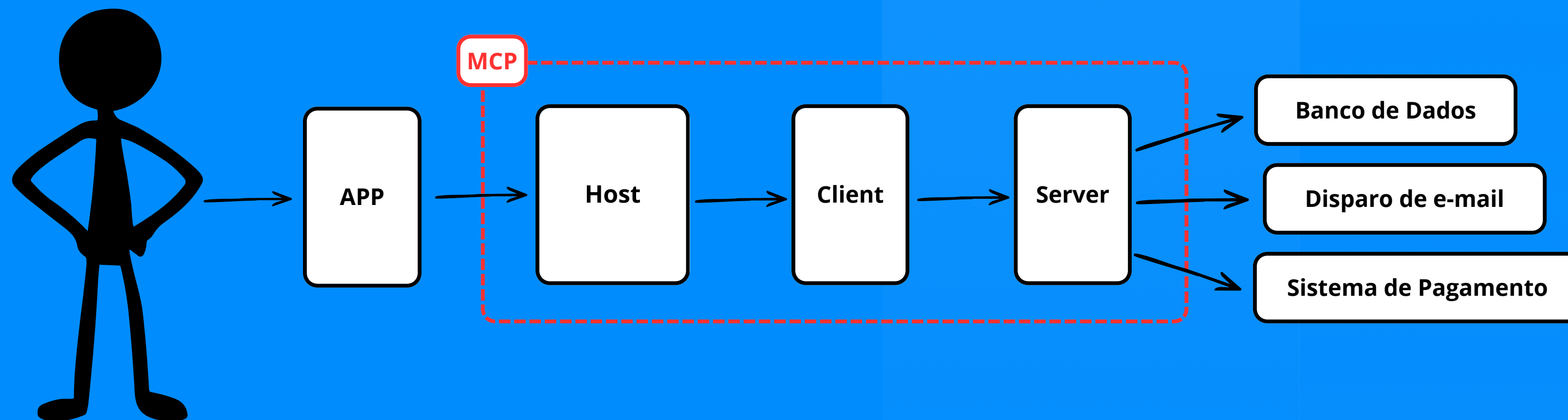
MCP



O que é MCP

- Protocolo aberto para conectar IA a ferramentas, dados e sistemas
- Anthropic, nov/2024 — hoje adotado por OpenAI, Google, Microsoft, IDEs...
- "USB-C para IA": um conector padrão no lugar de um cabo por dispositivo
- Arquitetura cliente-servidor sobre JSON-RPC 2.0
- Padroniza: inicialização · descoberta de capacidades · invocação

Arquitetura MCP



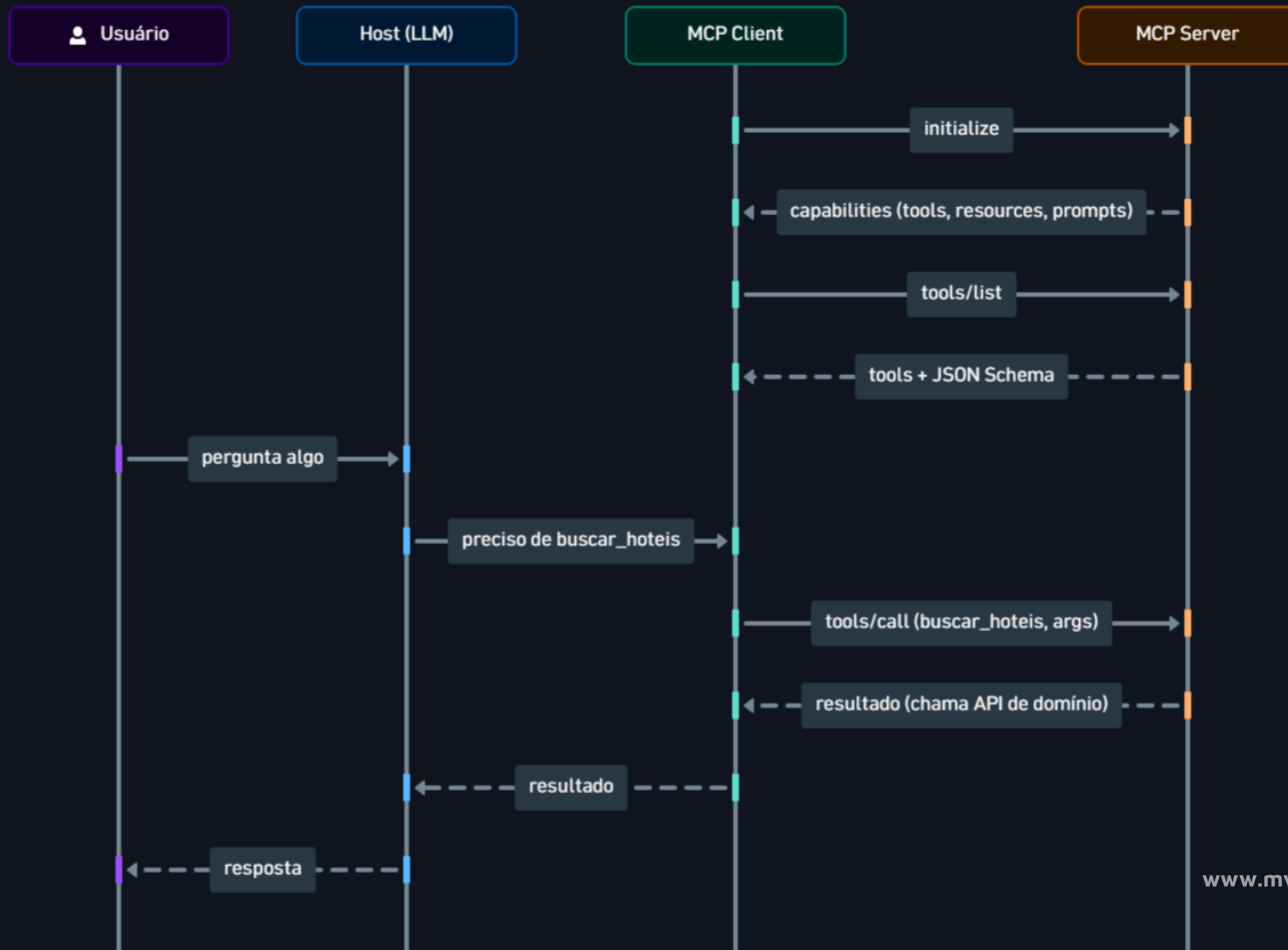
Quem é quem

Papel	Exemplos concretos
Host	Claude Desktop - Claude Code - seu sistema com LLM
Client	embutido no host — você não escreve *
Server	o que você constrói — .NET, Node, Python...

- "Host" ≠ servidor de infra — é o app de IA
- Transporte: stdio (local) ou Streamable HTTP (rede)

Primitivas

- Tools
- Resources
- Prompts



Menos de 10 linhas

```
var builder = WebApplication.CreateBuilder(args);

builder.Services
    .AddMcpServer()
    .WithHttpTransport(o => o.Stateless = true)
    .WithToolsFromAssembly();

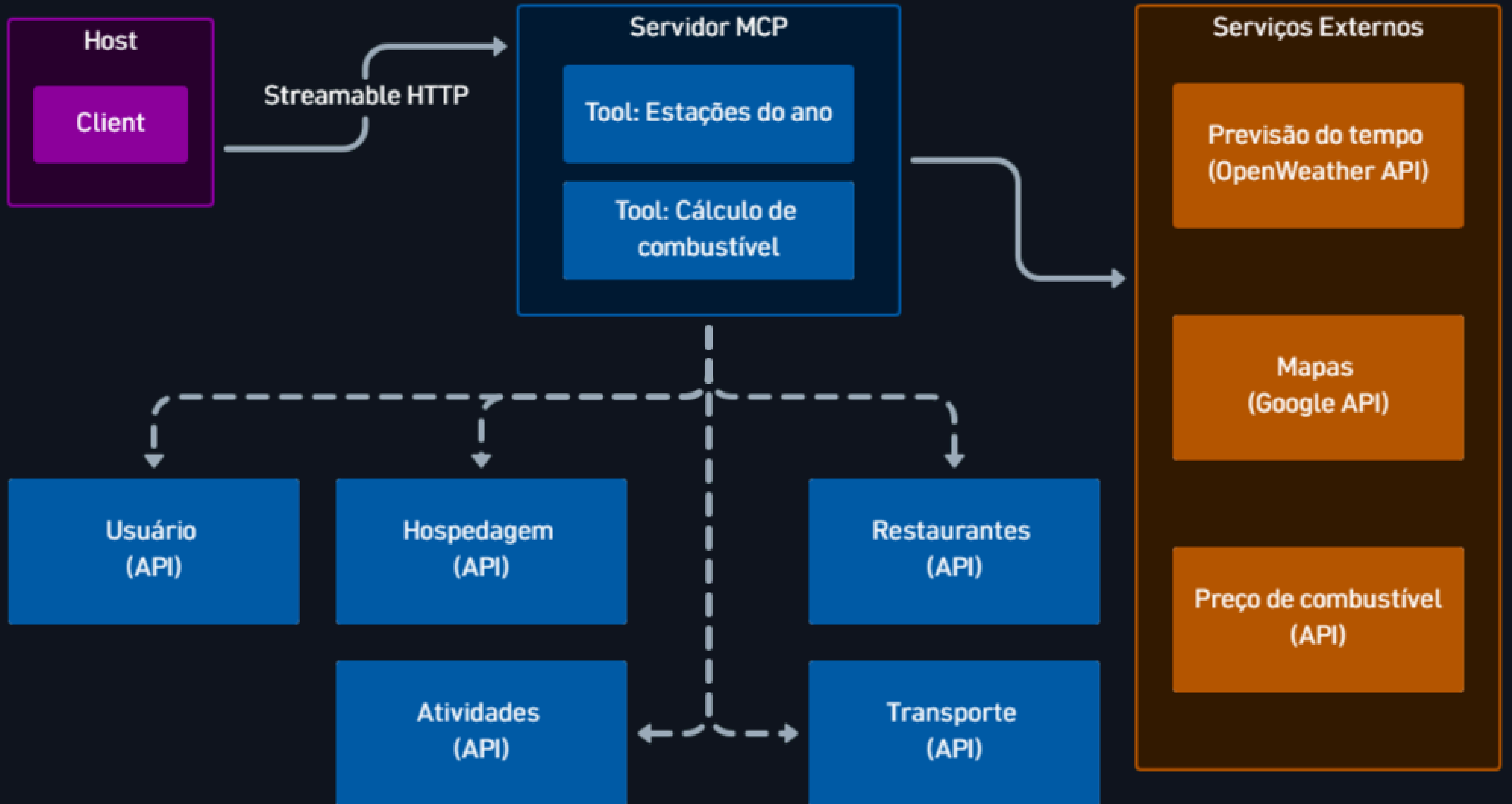
var app = builder.Build();
app.MapMcp();
app.Run();
```

Anatomia de uma tool

```
[McpServerToolType]                                // ← a classe agrupa tools
public static class HospedagemTools
{
    [McpServerTool,                                  // ← expõe o método como tool
     Description("Busca hotéis com quartos disponíveis, " +
                 "filtrando por estado e período.")] // ← contrato com a LLM
    public static async Task<List<HotelResumoResponse>> BuscarHoteis(
        HospedagemClient client,                    // ← injeção de dependência
        [Description("Sigla do estado (UF), ex: SP.")] string? estado = null,
        [Description("Check-in no formato AAAA-MM-DD.")] DateOnly? checkIn = null)
    {
        var query = QueryStringBuilder.Build(("estado", estado), ...);
        var response = await client.HttpClient.GetAsync($"/hoteis{query}");
        await response.EnsureSuccessOrThrowProblemDetailsAsync();
        return await response.Content.ReadFromJsonAsync<...>() ?? [];
    }
}
```

Demonstração

- Comprar passagem + hotel + passeios + restaurante = vários sistemas, várias telas
- A proposta: um assistente que resolve a viagem inteira conversando
- Ele concilia transporte, hospedagem, atividades e restaurantes - considerando clima, deslocamento e quem viaja
- Por baixo: um MCP server conectando a LLM a todos esses sistemas



Para produção

- Autenticação
- Observabilidade
- Contrato com a LLM

Antes de sair construindo

- Quantas tools por server?
- Descrição é interface.
- Do outro lado tem o client — hoje mostramos hosts prontos.

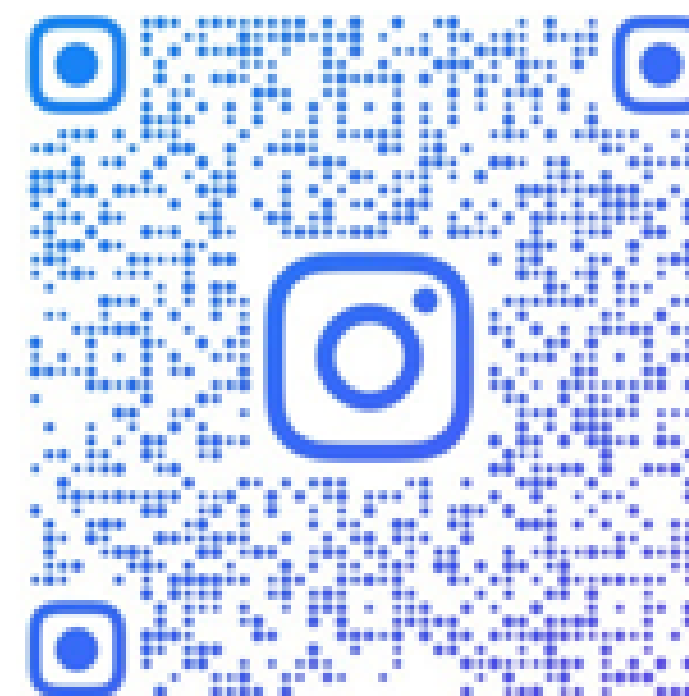
Recapitulando

- MCP é um protocolo aberto
- Host - Client - Server
- Uma tool = método static + [McpServerTool] + descrição + tipos
- Teste no Inspector antes da LLM
- Resources (a app injeta) e Prompts (o usuário aciona) além das tools

Siga MVP Conf nas redes sociais



LINKEDIN



@MVPCONF

Obrigado!