**MCP - O que é e Como Usar MCP N8N**

**Transcrito por** [**TurboScribe.ai**](https://turboscribe.ai/pt/?ref=docx_export_upsell)**.** [**Atualize para Ilimitado**](https://turboscribe.ai/pt/subscribed?ref=docx_export_upsell) **para remover esta mensagem.**

Nesse vídeo aqui eu vou te explicar sobre MCP, que é um conceito que está revolucionando a criação de agentes e de aplicativos. Eu fiz uma super aula pra você entender na prática o que é o MCP, porque ele é tão poderoso e revolucionário, como que a gente pode diferenciar MCP de APIs, como funciona toda a parte do MCP lá dentro do Netuene e vou te mostrar um projeto aqui pra você conseguir fazer o seu próprio MCP server sem precisar programar. Então bora comigo porque essa aula está bem completa e eu vou te passar todos os detalhes pra você conseguir entender de uma vez por todas o que é o MCP e porque é tão importante você começar a usar.

Bora lá! E pra gente entender sobre MCP, a gente tem que entender sobre as fases de criação e desenvolvimento de agentes de IA. A fase 1 foi quando foi lançado aí os modelos de inteligência artificial, que a gente tem os Large Language Models, por exemplo o ChatGPT, ele é com base na LLM do GPT da OpenAI, então isso daí já foi super revolucionário. Depois disso a gente entra na fase 2, a fase 2 a gente consegue utilizar os modelos de inteligência artificial para ativar ferramentas.

Isso daí foi uma grande evolução porque a gente consegue ativar Tools e fazer Function Calling, é o mesmo nome pra você entender, mas basicamente a partir agora da LLM, eu consigo que a minha LLM consiga ativar outras ferramentas da internet. Dessa forma fica muito mais fácil você conseguir interagir com qualquer outro sistema, por exemplo um banco de dados ou por exemplo um meio de pagamento, mandar um e-mail, criar um calendário ou algo, então agora se consegue criar agentes de IA que basicamente eles conseguem ser integrados em qualquer ferramenta, por isso que essa fase foi tão importante. E na fase 3 a gente tem aí mais uma grande revolução que é a parte do MCP e nada mais é que agora tudo fica mais organizado, mais padronizado, que você vai entender nesse vídeo como é que funciona e quais são as diferenças de cada uma dessas fases.

Bora entender então sobre MCP. Se você não viu, sobre MCP ele basicamente é um protocolo de contexto de modelo criado pelo Cloud, tem até um artigo sobre isso e vários e vários outros artigos que foi lançado alguns meses atrás e agora essa forma de padronização de comunicação das APIs está conseguindo evolucionar o jogo, ainda mais a hora que a gente consegue criar aí nos agentes de IA e utilizar MCP servidores aí de outros locais. Para você entender de uma vez por todas, MCP nada mais é que um padrão estruturado para rolar essa comunicação.

Eu vou te mostrar também quais são os benefícios disso. Bom, primeiramente antes de a gente começar a entender sobre MCP, você tem que entender dois pontos bem importantes para não ficar confuso, que é o MCP Server e o Client-Side, então a gente vai ter dois lados, a gente vai ter o Client-Side e vamos ter o Server-Side. E para ficar claro para você, o Client-Side é quando a gente faz uma requisição, então por exemplo o nosso agente de IA, o nosso modelo, nosso aplicativo está fazendo uma requisição para o servidor MCP, que é onde vai estar tudo padronizado para ativar essas ferramentas.

Então aqui no Client-Side geralmente ele vai fazer envio de uma requisição lá para o sistema MCP, que aí sim ele vai conseguir ativar as ferramentas. Depois o MCP Server-Side ele vai retornar com os dados, então por exemplo, o Google Calendar, por exemplo, ele pode ter um Server-Side, o meu agente de IA vai pedir para fazer uma requisição, uma criação de um evento na agenda, por exemplo. Então ele pede aqui com a linguagem natural, o MCP Server, como é tudo padronizado, não precisa ficar estruturando a API de uma forma bem detalhada, fazer exatamente a chamada, bem organizada, você apenas vai mandar de uma forma aí com texto, com linguagem natural.

Mas chegando no MCP Server, o Google, a parte do servidor do MCP, ele vai entender como fazer aquela chamada e vai retornar aqui com os dados, falando que a agenda foi criada. Isso para qualquer servidor, a gente pode ter um servidor, por exemplo, tem vários tipos de servidores hoje em dia do MCP, que a gente consegue requisitar informações e essa informação volta para a gente. Isso daqui nada mais é que facilita a ativação das ferramentas e das tools que a gente tanto fala aqui, para você criar agentes integrados.

Em resumo sobre a arquitetura do MCP, nada mais é que um grande adaptador para padronizar como os seus projetos de A vão conversar com outras ferramentas. Então, em relação ao servidor de MCP, ele é muito revolucionário, mas cada empresa tem que ter o seu próprio servidor para a gente conseguir consumir o servidor dessa empresa. Por exemplo, o Slack teria que ter o servidor deles, o Gmail teria que ter o servidor, teria que ter o MCP Server.

Para o nosso MCP Client, utilizando como modelo o Cloud AI, o OpenAI e outras LLMs, que geralmente a gente tem uma só agente, ele vai conseguir se comunicar aqui de uma forma padronizada. Isso deixa mais rápido, mais seguro, mais organizado toda essa comunicação de criar agentes que estão se comunicando com os servidores MCP dessas ferramentas. Então, ficou claro para você qual é a diferença entre o MCP Client e o MCP Server e como a gente faz essa conexão? Então, vamos entender quais são os benefícios disso, os benefícios de usar o MCP.

Primeira coisa é que a gente consegue ter a separação da inteligência do agente com a execução. Então, antes, para criar os agentes, a gente tinha que falar para o agente exatamente como ele executar. Eu falava, você é uma pessoa, você é um atendente que consegue fazer reserva nas agendas.

Aí, quando ele entendesse que ele tem que ativar alguma tool, eu teria que falar exatamente qual é o ID da agenda, eu teria que falar qual que é a ação que ele vai fazer, se ele vai conseguir criar um evento na agenda, ou só visualizar, ou só editar, eu teria que falar exatamente qual tipo de ação e eu teria que passar todos os detalhes. Agora, não. Agora, a gente fala o que a gente quer, ou a gente vai entender que ele tem que executar, daí ele vai lá no MCP Server, só que ele manda como uma linguagem natural.

Então, eu quero criar aqui um evento na agenda. Então, vai lá e já consegue configurar. Agora, ele quer usar o Create, Create Event.

Então, vai lá e vai criar o nosso evento. Então, fica muito mais fácil, a gente separa a inteligência do agente e consegue deixar mais organizado. Então, o agente apenas diz o que quer fazer e o executor resolve como, né, o MCP.

O benefício 2, a gente tem uma execução mais segura e controlada. Então, há uma solicitação da execução do servidor do MCP. Isso vai permitir, aí, pelo servidor, um log de auditoria, vai permitir também a gente fazer filtros de permissão, quem é que consegue ter acesso a qual coisa, qual ferramenta lá dentro do meu servidor, restrição de tempo e recursos.

Então, a gente consegue limitar e todo mundo continua usando aquele mesmo servidor. Então, é um baita benefício. Além disso, a gente consegue ter uma orquestração de sistemas mais complexos.

Por exemplo, você pode ter vários MCP servers, cada um de um sistema, né, um MCP de um CRM, de um calendário, de um pagamento e o agente não precisa conhecer a API de cada sistema. Então, antes, para fazer, tinha que criar toda a chamada API detalhada, colocar lá todo o CRUD, né, fazer o Create, fazer o Read, o Update, o Delete. Agora, não.

Agora, você conecta o servidor MCP e fica muito mais fácil. Basicamente, MCP é para agentes o que IA e o HTTP foi para a internet. Então, sem ele, as integrações são mais trabalhosas e, com ele, temos escalabilidade, controle e autonomia.

Então, agora, a gente consegue se comunicar muito melhor nossos agentes com todas as outras sistemas, tá? Então, é uma grande revolução, por isso que, cada vez mais, você vai estar falando e vai estar ouvindo sobre MCP. Então, agora, vamos comparar a parte de APIs com MCP. Então, como eu falei, quando a gente quer fazer uma chamada aqui do jeito antigo, a gente quer que o nosso agente de IA e daí a gente tem que estruturar cada evento, ó. A gente tem que colocar o GetEvents, CreateEvents, UpdateEvents, DeleteEvents, para o nosso agente conseguir fazer todas as interações com a agenda e daí a gente tem que detalhar cada um, o que pode ser um pouquinho mais trabalhoso e também fica mais difícil aí de dar manutenção ou de reutilizar esses mesmos componentes em outros projetos.

Então, agora, ao invés de ativar todas aquelas funções, a gente vai poder fazer apenas uma requisição lá para o servidor MCP do Google, do Calendar, né? Então, aí vai ter todas as ferramentas listadas e como tem essa nova padronização, fica muito mais fácil a comunicação. Eu saio disso daqui para isso. Olha que legal.

Então, trazendo aqui mais alguns exemplos de API e MCP, né? Então, a gente tem uma tarefa aqui, ó, por exemplo, AgendarReunião no Calendar. Na API tradicional, o agente precisa chamar o Google Calendar com autorização, validar e criar o Create, né? Com o MCP, o agente só fala assim, cara, eu quero criar um evento com o nome tal, hora tal e o MCP executa. Então, já vai ter essa padronização que vai ficar mais fácil.

Um link de pagamento do ASAS, por exemplo, o agente tem que executar a chamada API e é necessário enviar todos os parâmetros da forma correta. Com o MCP, ele vai direto lá no MCP, por exemplo, do ASAS. Então, o ASAS, ele tem que ter o próprio MCP para conseguir fazer aí essa comunicação.

Ou, por exemplo, atualizar o CRM de uma ferramenta CRM. Então, a gente tem que fazer uma chamada API, chamada HTTP, por exemplo, de PUT lá na API do CRM. Aqui não.

Ele apenas vai lá no servidor do CRM e consegue fazer essa comunicação, né? Não importa se atualizar, criar, editar, deletar. Já vai ter todo esse padrão de forma de comunicação. Então, para a gente ver mais algumas diferenças entre APIs e MCP.

Então, em relação em executação, o próprio agente chama diretamente a API. No modelo MCP, o agente solicita e a execução é feita pelo MCP Server, tá? Como a gente viu lá na parte do client-side e do server-side. Em relação ao complamento, a gente precisa conhecer cada API e conhecer como ativar cada chamada.

Em relação ao MCP Server, o agente só aciona e ele consegue já organizar de qual chamada vai ser feita. Segurança e controle. Aqui a gente tem um pouco mais dificuldade, né? De conseguir editar e controlar os nossos problemas, os logs aqui do modelo tradicional.

Aqui, no modelo MCP, a gente tem permissões e regras lá dentro. Então, a gente tem maior controle. Facilidade de manutenção, mas é trabalhoso a gente conseguir organizar, escalar, porque vai ter muitas APIs.

Aqui não, vai ficar mais tranquilo porque cada função vai estar se comunicando com o MCP Server aí diretamente. Em relação à governança, não tem uma improvisada, né? A gente não consegue ter essa estrutura. Aqui vai estar tudo organizado e padronizado em relação a esse ponto.

Então, pra você entender sobre MCPs, a gente se pergunta, né? Então, quais ferramentas que já tem o MCP Server? E aqui tem um link, um link do GitHub que a gente consegue ver os servidores MCPs que já tem. Aqui é um ponto importante, como é tudo muito novo, não são todas as ferramentas que estão com o servidor MCP ativo, tá? A gente tem algumas ferramentas grandes com o MCP Servidor oficial rodando, mas tem muitas empresas aí que não construíram o seu próprio servidor. Então, aqui a gente consegue ver aqui integrações oficiais de empresas, né? Tem, por exemplo, o Libabar, a gente tem o Epify, a gente tem aqui a AWS, a Azure.

Então, aqui tem toda a documentação MCP. Se a gente for ver aqui pra baixo, a gente tem também os MCPs da comunidade, tá? Então, a comunidade conseguiu construir os servidores pra ter essa padronização e ficar mais fácil se comunicar com essas ferramentas. Isso é muito útil, só que a gente não consegue garantir aí a qualidade e nem a segurança porque não são servidores oficiais, então é um ponto pra ficar de olho.

Então, vamos falar sobre MCP lá dentro do Unituene. O Unituene é uma ferramenta muito completa e eles estão recebendo muito investimento e já conseguiram dar um passo à frente em relação ao MCP e já estão integrando, né? Então, a gente consegue lá na nossa parte do MCP Client criar a nossa integração pra ele se comunicar com os servidores MCPs externos. Só que além disso, da parte interna, a gente consegue fazer também um próprio MCP Server dentro do Unituene.

Bora entender isso aqui então. Então, como eu falei, no Unituene a gente consegue acessar os servidores MCP. Então, aqui a gente consegue ter o modelo do agente, a memória e aqui a parte de ativar o MCP.

Ele pode ir lá no MCP externo e conseguir ativar isso, tá? Então, ele consegue ativar pela parte do MCP Client. Mas no Unituene também a gente consegue fazer, criar o nosso próprio MCP Server, né? Então, a gente vai ativar o MCP Tool, só que a gente vai ativar pra um servidor que vamos criar diretamente lá dentro do Unituene. Então, esse daí é um grande ponto interessante pro Unituene que a gente consegue fazer tanto a parte do Client-side quanto criar o nosso próprio servidor MCP.

E a gente tem algumas vantagens e desvantagens sobre esses dois pontos. Em relação a consumir MCP Server, é muito bom, realmente, a gente consegue acessar aqui qualquer outro tipo de servidor MCP. Então, como eu falei aqui no último vídeo, a gente tem uma grande revolução da comunicação dos APIs, maior performance dos agentes e maior escabilidade por ter toda essa padronização e organização agora.

Então, show demais! A desvantagem única é que, como é muito novo, há poucos servidores oficiais ainda. Então, não dá pra garantir a segurança e a qualidade de MCP Service da comunidade. Então, isso daí tem que tomar um pouco de cuidado quando você for usar algum servidor MCP.

Olha se é oficial e se realmente tem toda a parte de segurança. Então, excelente usar um MCP Client dentro do Unituene pra ele se comunicar com o MCP Server de outras ferramentas. Show demais! Mas, em relação ao MCP Server aqui dentro do Unituene, como que a gente pode entender sobre isso? Tem algumas vantagens.

A gente consegue maior organização e padronização dos projetos. Então, fica tudo mais organizado ao invés de ficar ativando várias tools. Também, a gente consegue selecionar funções que a gente quer que execute.

Então, eu posso ir aqui no nosso servidor e falar assim, ah, eu quero que esse agente aqui, ele só execute a parte de procurar eventos ou a parte de criar eventos. Então, como você consegue fazer isso? Isso é bem interessante a gente dar essa organização. E, além disso, a gente consegue ter uma centralização de MCP Server pra usar em vários agentes.

Então, vários agentes podem usar sempre os mesmos servidores MCP, tá? Fica mais organizado. Porém, um ponto aí que a maioria das pessoas fica um pouco confusa é que não tem ganhos claros em relação ao MCP Server do Unituene comparando com a parte de ativar por tools. Então, se você ativar as ferramentas chamando outros fluxos, chamando aí as tools, é quase a mesma coisa que chamar os MCP Servers, tá? Não tem uma diferença clara ainda.

Do jeito que está, a gente não tem tantos benefícios de criar nossos próprios servidores MCP, a não ser que uma organização, uma melhor padronização, mas comece a ter a mesma a mesma execução que a gente ativar aí tools, tá? E, também, um ponto negativo é que pode ter um maior consumo da infra da sua hospedagem, tá? Porque a gente está criando um server aqui do zero. Então, essa é uma desvantagem que não é muito legal ali para os nossos projetos. Então, você sempre saiba equilibrar essas partes, tá? Como que você vai usar a parte do MCP Client e a parte do MCP Server.

Então, só pra você ver um exemplo, aqui dentro do Unituene a gente tem aula sobre como você cria um agente de A, que é vendedor, que faz atendimento e consegue fazer vendas aí para a sua empresa. Então, aqui eu tenho toda a parte de organização aqui do cliente e aqui eu consigo fazer a ativação do meu MCP Server, tá? Aqui o meu agente, ele consegue ativar para ir lá no MCP Servidor que eu criei no meu próprio Unituene para fazer a ativação aí da parte de gerir um pedido, né? E conseguir aí enviar um link de pagamento. E lá no MCP Server, ó, eu tenho essas três partes.

Eu tenho a tabela clientes, tem a parte de tabelas pedidos e a parte dos pagamentos A's. Então, o meu MCP Server aqui ele fica apenas esperando alguém acionar ele, né? Então, quando é acionado ele consegue ir lá nas tabelas, atualizar cliente, ler cliente, ler os produtos, criar pedidos, fazer aqui a calculadora, fazer cálculo de criar um pedido, ele já cria o carrinho automático aí, ele cria um link de pagamento personalizado com Pix, boleto, cartão de crédito. Então, esse meu servidor MCP, ele tá funcionando lá esperando o nosso agente utilizar ele para fazer essas criações.

Eu poderia ter outros agentes utilizando esse mesmo servidor, mas a forma que a gente usa é desse jeito aqui, bem tranquilo. Só você vir aqui, criar uma tool e aí você consegue colocar MCP, MCP para ter a sua parte do MCP Client. Dessa forma, você consegue também criar o seu MCP Server, que nada mais é que esse trigger, esse nosso gatilho para ativar outras informações aqui e outros sistemas.

Então, é muito fácil, a gente consegue ter projetos mais organizados dentro do N2N e para construir aí os seus cases. E o que você achou desse vídeo? Você curtiu as informações? Você concorda com a parte de MCP? Já tá usando MCP nos seus projetos? Comenta aqui que esse vídeo aqui deu bastante trabalho para ser construído e organizar as informações e a única coisa que eu peço em troca é que você comente aqui embaixo para saber se você gostou ou não. Se você quer aprender mais sobre como criar agentes, eu vou deixar o link aqui embaixo sobre a nossa formação de agente já que está completinha demais.

Você aprende aí tudo sobre agente já do zero ao avançado, também criar agentes com tools, agentes com MCP. Dá uma olhada que está capturado demais. Fechou galera, vejo você no próximo vídeo.

Até mais!

**Transcrito por** [**TurboScribe.ai**](https://turboscribe.ai/pt/?ref=docx_export_upsell)**.** [**Atualize para Ilimitado**](https://turboscribe.ai/pt/subscribed?ref=docx_export_upsell) **para remover esta mensagem.**