

Model Context Protocol (MCP): Do Conceito à Prática

Um guia completo sobre os servidores Model Context Protocol que conectam assistentes de IA às suas ferramentas — do design e desenvolvimento de UI/UX à automação do seu ambiente de trabalho diário.

MCP SERVERS	WORKFLOWS	ECOSSISTEMA	LICENÇA
19	5	UI/UX + Dev	Open-source

FUNDAMENTOS

O que é o Model Context Protocol (MCP)?

O **Model Context Protocol (MCP)** é um protocolo **aberto e padronizado**, criado pela **Anthropic**, que define como modelos de linguagem (LLMs) e agentes de IA se conectam a ferramentas e dados externos. Antes do MCP, cada integração exigia uma ponte sob medida; com o protocolo, um **mesmo servidor MCP** pode ser usado por qualquer assistente de IA compatível.

Na prática, o MCP organiza essa conexão em três camadas:

Papel	Função
Cliente MCP	O assistente ou IDE (Claude Desktop, Cursor, Windsurf, Cline, opencode...) que hospeda o agente e conversa com os servidores.
Servidor MCP	Um processo que expõe as ferramentas (ações), recursos (dados/contexto) e prompts (fluxos prontos) de um sistema — Figma, banco de dados, navegador, Jira etc.
Ferramentas / Recursos / Prompts	O que o servidor oferece ao modelo: executar ações, buscar informações e guiar tarefas dentro da ferramenta conectada.

O resultado é que a IA deixa de ser apenas um **conselheiro passivo** e passa a ser um **participante ativo**: ela pode ler um design, gerar código, testar uma interface e consultar seus sistemas — tudo por meio de servidores MCP.

Por que isso importa aqui: este guia reúne **19 servidores MCP** — 14 do ecossistema de UI/UX e 5 do seu ambiente diário — organizados não por lista técnica, mas por **workflows de trabalho**, para que você saiba **qual servidor usar em cada etapa e como combiná-los**.

“O futuro da engenharia de UI/UX é colaborativo, automatizado e movido por IA — e os servidores MCP são a ponte para esse futuro.”



Brainstorm & Descoberta Visual

O ponto de partida de qualquer produto: explorar ideias, organizar referências e obter feedback visual rápido. Aqui, dois servidores ajudam a transformar conversas em quadros visuais e a “ver” o que está na tela.

MCP-Miro

evalstate

★ 33

Conecta o Claude Desktop (e outros) à plataforma de quadro branco **Miro**.

Ferramenta	Descrição
<code>create_sticky_note</code> / <code>create_shape</code>	Cria post-it / forma em um quadro Miro
<code>bulk_create_items</code>	Cria vários itens de uma vez
<code>get_board_items</code> / <code>get_frames</code>	Recupera itens / frames de um quadro
<code>get_items_in_frame</code>	Obtém itens dentro de um frame específico

APIs: Miro REST API, Node.js, TypeScript, MCP · **Chaves:** token OAuth Miro. Ótimo para brainstorming visual — “tirar foto de post-its e pedir ao Claude para criar um equivalente no Miro”.

Vibe-Eyes

monteslu

★ 22

Atua como os “**olhos**” dos **LLMs** em sessões de vibe coding com jogos e apps baseados em navegador, resolvendo o problema de contexto visual.

Ferramenta	Descrição
<code>getGameDebug</code>	Obtém visualização de canvas e informações de depuração

APIs: @neplex/vectorizer, Socket.IO, Express, SVG0, MCP SDK, Zod · **Chaves:** nenhuma. Cliente de navegador captura o canvas, servidor Node.js vetoriza as imagens e as disponibiliza via MCP — útil para depurar jogos.

Design & Prototipagem

A etapa em que o design toma forma no Figma, Framer, Illustrator e Photoshop. São os servidores que deixam a IA ler, criar e editar seus arquivos de design diretamente.

Cursor Talk to Figma MCP

Sonny Lazuardi

★ 2,4 mil

Integração poderosa que implementa o MCP entre o **Cursor AI** e o **Figma**, permitindo ler e modificar designs via linguagem natural (projetado para Cursor, mas adaptável a outros clientes).

Categoria	Exemplos de ferramentas
Documento & Seleção	<code>get_document_info</code> <code>get_selection</code> <code>get_node_info</code>
Criação de elementos	<code>create_rectangle</code> <code>create_frame</code> <code>create_text</code>
Texto & Estilização	<code>set_text_content</code> <code>set_fill_color</code> <code>set_stroke_color</code> <code>set_corner_radius</code>
Layout & Organização	<code>move_node</code> <code>resize_node</code> <code>delete_node</code>
Componentes & Export	<code>get_styles</code> <code>create_component_instance</code> <code>export_node_as_image</code> <code>execute_figma_code</code>

APIs: Figma Plugin API, WebSocket, Bun, TypeScript, Zod · **Chaves:** nenhuma (exige plugin Figma + Cursor). Conexão via WebSocket entre Cursor AI e o plugin do Figma.

Framelink Figma MCP Server

GLips

★ 4,4 mil

Faz a ponte entre arquivos do Figma e ferramentas de codificação com IA (Cursor, Windsurf e Cline).

Ferramenta	Descrição
<code>get_file</code>	Busca e simplifica os dados de um arquivo do Figma
<code>get_images</code>	Baixa imagens de um arquivo do Figma

APIs: Figma REST API, MCP SDK, Express.js, Zod, Remeda · **Chaves:** token da API Figma. Diferencial: simplifica respostas complexas da API em formatos limpos e legíveis por LLM.

Figma MCP Server com Chunking

archimedescripto

★ 1

Especializado em arquivos Figma **grandes** por meio de gerenciamento avançado de memória (chunking e paginação) – ideal para o **Workflow 2** quando o design scale além do tamanho normal.

Ferramenta	Descrição
<code>get_file_data</code>	Obtém dados do arquivo com chunking e paginação
<code>list_files</code> / <code>get_file_versions</code>	Lista arquivos / histórico de versões em blocos
<code>get_file_comments</code> / <code>get_file_info</code>	Comentários com paginação / info com travessia de nós
<code>get_components</code> / <code>get_styles</code>	Componentes e estilos com suporte a chunking
<code>get_file_nodes</code>	Obtém nós específicos com suporte a chunking

APIs: Figma API, MCP, Node.js, TypeScript, Axios, Docker · **Chaves:** `FIGMA_ACCESS_TOKEN` . Ideal para sistemas de design empresariais com milhares de nós vetoriais.

Framer Plugin MCP

Sheshiyer

★ 7

Criação e gerenciamento de **plugins Framer com recursos web3** (carteiras cripto e contratos inteligentes), integrando protótipos a funcionalidades descentralizadas.

Ferramenta	Descrição
<code>create_plugin</code>	Cria um novo projeto de plugin Framer com recursos web3
<code>build_plugin</code>	Compila o plugin Framer para produção

APIs: MCP SDK, Framer SDK, Web3, Node.js, TypeScript · **Chaves:** nenhuma. Permite Wallet Connect, interação com contratos e componentes de NFTs.

Illustrator MCP Server

Spencer Hubert

★ 14

Permite que IA interaja com o **Adobe Illustrator** via execução de JavaScript e feedback visual — abre caminho para criação automatizada de vetores e prototipagem.

Ferramenta	Descrição
<code>captureIllustrator</code>	Screenshot do workspace atual do Illustrator
<code>runIllustratorScript</code>	Executa código JavaScript no Illustrator

APIs: Python, MCP, AppleScript, Adobe JS API, httpx, Pillow · **Requisitos:** **Só macOS** Adobe Illustrator.

Photoshop Python API MCP Server

loonghao

★ 11

Ponte entre Python e o **Adobe Photoshop** para controle programático e automação de tarefas de edição.

APIs: Python, Poetry, Nox, pytest, GitHub Actions · **Chaves:** nenhuma.

⚠ Em estágio inicial. O autor não recomenda instalar ainda, mas, com contribuições open-source, poderia ser um divisor de águas para automatizar operações do Photoshop.

Design → Código (Handoff)

A ponte entre o design aprovado e o código real: converter layouts em componentes, organizar/documentar a UI e gerar estrutura para apps. São os servidores que **eliminam o atrito do design-to-code**.

MCP Figma → Converter React

studentofjs

★ 4

Converte designs do Figma **diretamente em componentes React**, agilizando o fluxo design→código com acessibilidade de fábrica.

Categoria	Ferramentas
Ferramentas Figma	<code>getFigmaProject</code> <code>getFigmaComponentNodes</code> <code>extractFigmaComponents</code>
Ferramentas React	<code>generateReactComponent</code> <code>generateComponentLibrary</code>

APIs: Figma API, React, Tailwind CSS, TypeScript, Express.js, Zod, Axios, Prettier · Chaves: token API Figma. Acessibilidade Tailwind
Adiciona atributos de acessibilidade automaticamente e converte estilos do Figma em classes Tailwind.

Storybook MCP Server

dannyhw

★ 4

Faz a ponte entre LLMs e o **Storybook**, o famoso framework de desenvolvimento de componentes UI – promissor para **descoberta de componentes, documentação e sugestões de testes**.

Ferramenta	Descrição
<code>get-stories</code>	Recupera uma lista de stories do Storybook

APIs: MCP SDK, Storybook, Bun, TypeScript, Zod · Chaves: nenhuma.

MCP Server Flutter

Anton Malofeev

★ 18

Interage com aplicações **Flutter**, ligando modelos de linguagem ao desenvolvimento de apps – útil para fluxos de desenvolvimento automatizados e testes mobile.

Ferramenta	Descrição
<code>flutter_execute</code>	Executa comandos Flutter em um repositório (create, pub, run, screenshot)

APIs: MCP SDK, Flutter, TypeScript, Node.js, Zod · Requisitos: caminho válido para um repositório Flutter.

Testes & QA de UI

Garantir que a interface funcione e se comporte como o esperado: testar entre navegadores, validar fluxos e checar acessibilidade. Dois servidores dominam essa etapa.

Playwright MCP

Microsoft

★ 7 mil

Extensão do framework **Playwright** que permite a agentes de IA interagir com navegadores de forma estruturada. **Implementação oficial da Microsoft.**

Modo Snapshot	Modo Visão	Uso típico
<code>browser_navigate</code> <code>browser_click</code> <code>browser_type</code> <code>browser_select_option</code> <code>browser_snapshot</code> <code>browser_take_screenshot</code>	<code>browser_screenshot</code> <code>browser_move_mouse</code> <code>browser_click</code> <code>browser_type</code>	Testes web, extração de dados, automação de formulários, pesquisa de UI/UX

APIs: Playwright, MCP, Node.js, Zod, TypeScript · **Chaves:** nenhuma. Abordagem focada em acessibilidade e interações determinísticas. O modo visão cria um ciclo de feedback multimodal.

MCP Selenium Server

Angie Jones

★ 78

Automação de navegador por meio do consagrado framework **Selenium WebDriver** — ideal para testes entre navegadores, automação de fluxo e acessibilidade.

Ferramenta	Descrição
<code>start_browser</code> / <code>navigate</code>	Inicia sessão / navega para uma URL
<code>find_element</code> / <code>click_element</code>	Encontra / clica em um elemento
<code>send_keys</code> / <code>get_element_text</code>	Digita / obtém o texto de um elemento
<code>hover</code> / <code>drag_and_drop</code> / <code>take_screenshot</code>	Hover / arrastar e soltar / screenshot

APIs: Selenium WebDriver, MCP SDK, Node.js, Chrome/Firefox · **Chaves:** nenhuma.

Automação Multi-domínio

Fluxos que cruzam várias ferramentas e domínios em um único agente — da geração de imagem à conversão de Markdown. Aqui a flexibilidade é o superpoder.

MCP-Toolbox

AI-ZeroLab

★ 9

Coleção abrangente que estende as capacidades dos LLMs — a maior da lista, com **mais de 30 ferramentas** para automação multi-ferramenta.

APIs: múltiplas APIs, Python · **Chaves:** várias conforme a ferramenta (Figma, Tavily, etc.) · **Integração:** `Docker` . Ideal para fluxos entre domínios: geração de imagem, conversão Markdown e acoplamento com outros MCPs.



Análise Comparativa

Os **19 servidores** do guia organizados por workflow, com foco e principais casos de uso.

Servidor	Workflow	Foco	Casos de uso
MCP-Miro	1 · Brainstorm	Quadro branco	Brainstorming visual, mapas
Vibe-Eyes	1 · Brainstorm	Contexto visual	Feedback visual, debug de jogos
Cursor Talk to Figma	2 · Design	Integração Figma	Manipulação de design
Framelink Figma	2 · Design	Acesso a dados Figma	Leitura/limpeza de arquivos
Figma c/ Chunking	2 · Design	Figma em arquivos grandes	Sistemas empresariais
Framer Plugin	2 · Design	Plugins Framer c/ web3	Blockchain, NFTs
Illustrator MCP	2 · Design	Controle do Illustrator	Vetores, automação
Photoshop API	2 · Design	Automação do Photoshop	Edição, batch
Figma → React	3 · Handoff	Conversão para código	Automação design-to-code
Storybook MCP	3 · Handoff	Descoberta de componentes	Documentação de UI
Flutter MCP	3 · Handoff	Comandos Flutter	Apps Flutter, teste mobile
Playwright MCP	4 · QA	Automação de navegador	Testes web, pesquisa UI
MCP Selenium	4 · QA	Testes com Selenium	Testes entre navegadores
MCP-Toolbox	5 · Integração	Coleção abrangente	Fluxos entre domínios
ai-memory	Ambiente	Memória persistente	Handoff entre sessões
jira	Ambiente	Integração Jira	Tickets, projetos (leitura)
postgres	Ambiente	Banco PostgreSQL	Consultas SQL, schemas
filesystem	Ambiente	Sistema de arquivos	Leitura/escrita de arquivos
chrome-devtools	Ambiente	Chrome ao vivo	Inspeção, screenshots, PDF

Os 5 MCPs do seu ambiente

Complemento deste guia: os servidores MCP **realmente configurados** no seu ambiente (arquivo `~/.config/opencode/opencode.json`). São o exemplo mais concreto de “MCP na prática” – todos com **enabled: true**.

MCP	Tipo	Comando / URL	Status
ai-memory	remote	http://127.0.0.1:49374/mcp	✓ ativo
jira	local	npx mcp-jira-stdio	✓ ativo
postgres	local	npx mcp-postgres	✓ ativo
filesystem	local	@modelcontextprotocol/server-filesystem	✓ ativo
chrome-devtools	local	npx chrome-devtools-mcp@latest	✓ ativo

1. ai-memory – Memória persistente

servidor remoto local

Servidor de **memória de longo prazo** para agentes. Além do endpoint MCP, há um plugin (`ai-memory.ts`) que intercepta sessões, prompts e execuções de ferramentas, captura o que o agente lê/escreve (com política de captura via marcadores `.ai-memory.toml`) e envia ao servidor em `127.0.0.1:49374`. No início de cada sessão, busca um **handoff** (`/handoff`) e injeta no system prompt o contexto de trabalhos passados – dando memória contínua ao agente entre sessões.

- Memória contínua
- Handoff entre sessões
- Captura de contexto

2. jira – Integração Jira

npx mcp-jira-stdio · freema

Servidor MCP do **Jira**. Expõe `jira_get_visible_projects`, `jira_get_project_info`, `jira_get_issue`, busca e comentários. Config: `JIRA_BASE_URL`, `JIRA_EMAIL`, `JIRA_API_TOKEN` – aponta para `lemontechinfo.atlassian.net`.

⚠ **Segurança:** há uma regra instalada (`jira-safety.md`) que proíbe qualquer escrita – apenas leitura. E o token real da API está exposto em texto puro no config.

3. postgres – Banco PostgreSQL

npx mcp-postgres · kristofer84

Servidor MCP do **PostgreSQL**: `list_tables`, `get_schema`, `query_data` (só SELECT), além de insert/update/delete e DDL. Suporta SSL, auto-config para **AWS RDS** e modo **read-only** (`DB_READ_ONLY=true`). Aponta para `postgresql://viaje-dev:...@localhost:5432/viaje-dev`.

⚠ **Segurança:** a doc recomenda usar role de **menor privilégio** em vez do superusuário; `execute_raw_query` não tem restrição de escrita.

4. filesystem – Sistema de arquivos

@modelcontextprotocol/server-filesystem · oficial

Servidor **oficial de referência**: lê/escreve arquivos, cria/lista/exclui diretórios, move/renomeia, busca e metadados, com **controle de acesso dinâmico via Roots**. Acesso liberado a `/home/clenilsonlopes/lemontech` e à pasta de plugins do Eclipse (`tmp0`).

5. chrome-devtools – Chrome DevTools MCP

npx chrome-devtools-mcp@latest · Google

O **oficial do Google Chrome** (48,7 mil ★): permite ao agente controlar e inspecionar um Chrome ao vivo — navegar, clicar, screenshots, árvore de acessibilidade, DOM, console, rede, traces de performance/memória e **print to PDF**. É o tipo de ferramenta capaz de gerar/recapturar PDFs (como este guia) automaticamente.

COMBINAÇÕES

Usando MCP Servers Juntos

O poder real vem de combinar servidores em fluxos de trabalho de agentes. Três cenários:

Cenário 1 – Pipeline completo design→implementação

- 1. **Cursor Talk to Figma** — explorar/entender a estrutura de um design
- 2. **Figma to React Converter** — gerar o código dos componentes
- 3. **Playwright MCP** ou **MCP Selenium** — testar os componentes
- 4. **Storybook MCP** — documentar os componentes
- 5. **MCP-Miro** — criar apresentação visual para stakeholders

Cenário 2 – Design system multiplataforma

- 1. **Figma MCP com Chunking** — extrair componentes de arquivos grandes
- 2. **Figma → React Converter** — gerar componentes web
- 3. **MCP Server Flutter** — implementações mobile
- 4. **Playwright MCP** — testar entre navegadores
- 5. **MCP-Toolbox** — documentar tudo (conversão Markdown)

Cenário 3 – Criação avançada de ativos visuais

- 1. **MCP-Toolbox** — gerar conceitos iniciais (imagem)
- 2. **Illustrator MCP** — refinar gráficos vetoriais
- 3. **Photoshop Python API** — efeitos de foto (quando pronto)
- 4. **Framer Plugin MCP** — integrar aos protótipos
- 5. **Vibe-Eyes** — feedback visual durante testes

POR QUE USAR

Benefícios e Conclusão

Benefício	Descrição
Unindo Design e Desenvolvimento	Eliminam o atrito entre design e código, com transições perfeitas entre fases criativas e técnicas
Automatizando tarefas repetitivas	Controlam ferramentas programaticamente, liberando de tarefas manuais como design-to-code e testes entre navegadores

Colaboração aprimorada	Criam linguagem comum entre designers, desenvolvedores e assistentes de IA
Acessibilidade	Muitos (como Playwright MCP) têm suporte embutido a acessibilidade, apps mais inclusivos por padrão
Aceleração de fluxo	Acesso direto às ferramentas diárias pode comprimir cronogramas de desenvolvimento

O futuro da engenharia de UI/UX é colaborativo, automatizado e movido por IA — e os servidores MCP são a ponte para esse futuro.