

In this article



Diseño de Figma → Modo de desarrollo → Convertir diseños en código

## Guía del servidor MCP en modo Dev

⚠ Actualmente, el servidor MCP en Dev Mode está en fase beta abierta . Es posible que algunas funciones y configuraciones aún no estén disponibles. La función puede cambiar y podría experimentar errores o problemas de rendimiento durante la versión beta.

### ¿Quién puede utilizar esta función?

- 👤 El servidor MCP en Dev Mode está en fase beta abierta.
- 🔒 Disponible en un puesto Dev o Full en los planes Profesional, Organización o Empresa .
- 👤 Debes usar un editor de código o una aplicación que admita servidores MCP (por ejemplo, VS Code, Cursor, Windsurf, Claude Code).
- 🔒 Solo puedes usar el servidor MCP en Dev Mode desde la aplicación de Figma para escritorio. [Descarga la aplicación Figma para escritorio →](#)

El servidor MCP en Dev Mode integra Figma en tu flujo de trabajo brindando información de diseño esencial y contexto a los agentes de IA que generan código a partir de archivos de diseño de Figma.

Los servidores MCP son parte de una interfaz estandarizada para que los agentes de IA interactúen con fuentes de datos utilizando el Protocolo de contexto de modelo .

Con el servidor habilitado, puedes hacer lo siguiente:

[? Ayuda](#)

producto que desarrollan nuevos flujos o iteran en las funciones de la aplicación.

Extraer el contexto del diseño

Incorpora datos de disposición, componentes y variables directamente en tu IDE. Esto es especialmente útil para los sistemas de diseño y los flujos de trabajo basados en componentes.

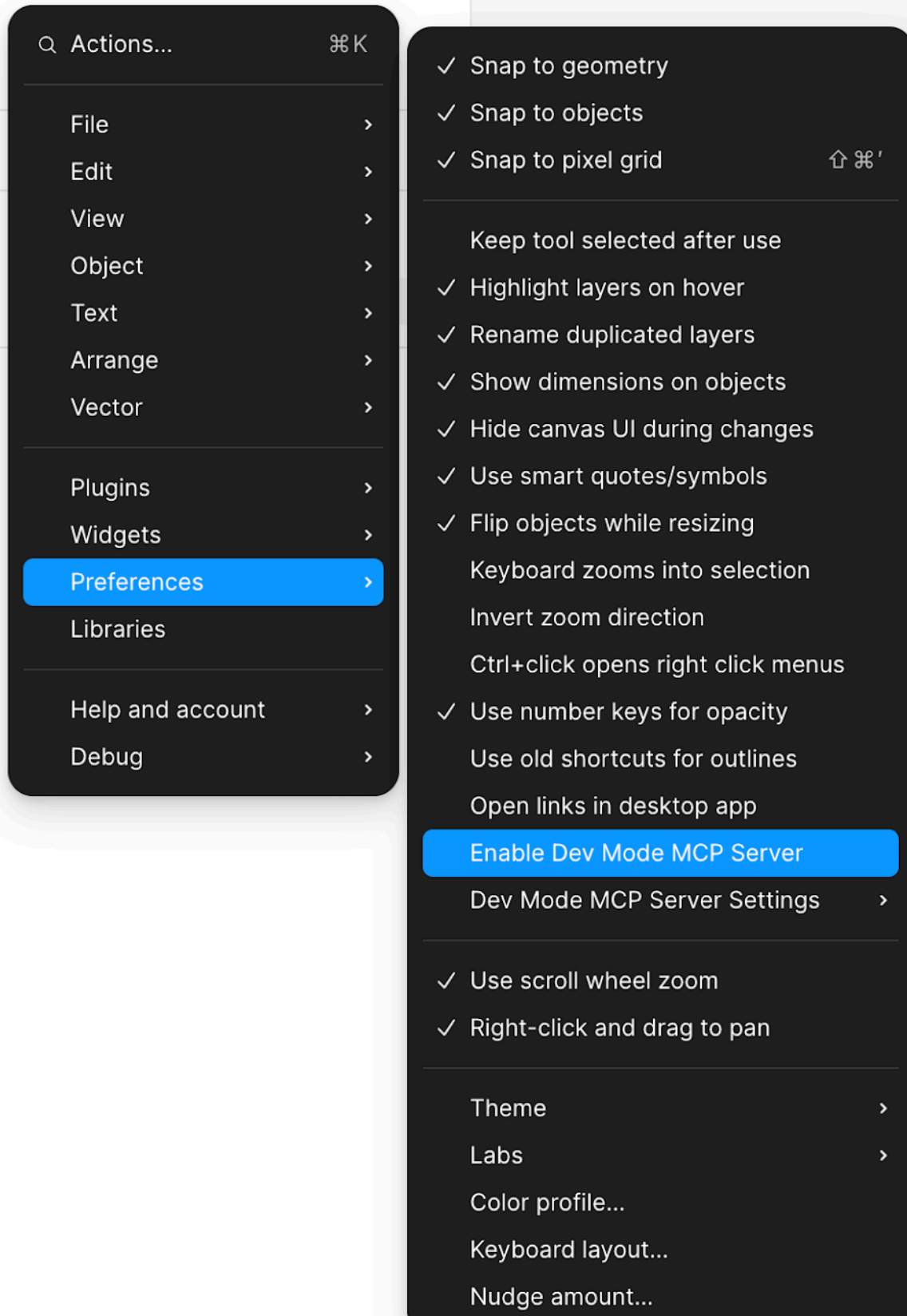
- **Codifica de manera más inteligente con Code Connect**

Mejora la calidad del resultado reutilizando tus componentes actuales. Code Connect mantiene tu código generado consistente con tu base de código.

[Obtener más información sobre Code Connect →](#)

## Paso 1: Habilita el Servidor MCP

1. Abra la [aplicación Figma para escritorio](#) y asegúrese de haber [actualizado a la última versión](#) .
2. Crea o abre un archivo de Figma Design.
3. En la esquina superior izquierda, abre el menú de Figma.
4. En **Preferencias** , seleccione **Habilitar Servidor MCP en Dev Mode** .



El servidor se ejecuta localmente en `http://127.0.0.1:3845/sse` . Ten esta dirección a mano para tu archivo de configuración en el siguiente paso.

## Paso 2: Configura tu cliente MCP

Una vez que el servidor esté funcionando localmente en la aplicación de Figma para escritorio, los clientes MCP podrán conectarse a su servidor. Siga las instrucciones para su cliente específico para agregar el servidor MCP en Dev Mode.

**Código VS**

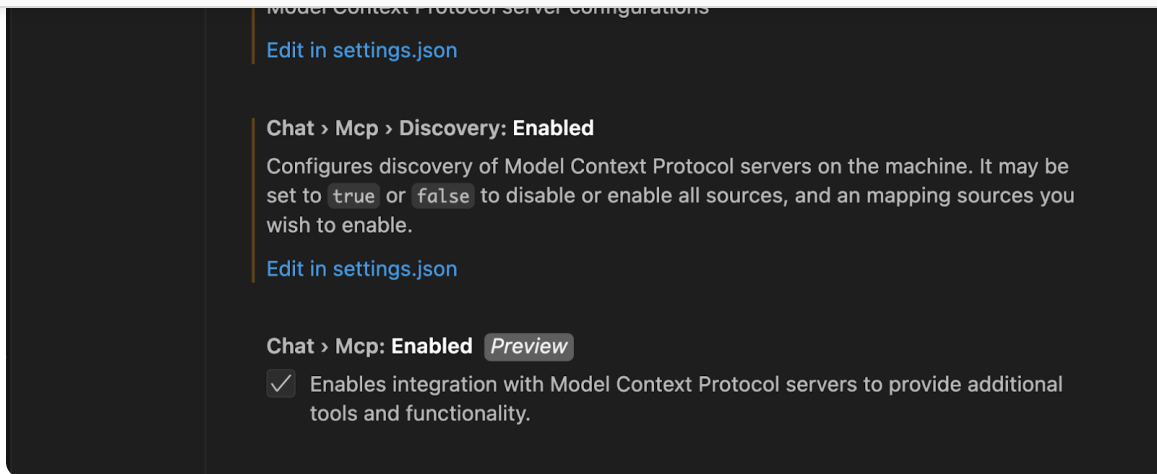
Cursor

Windsurf

Claude Code

Otros editores

1. Ve a **Código** → **Configuración** → **Configuración** o usa el atajo `⌘,`
2. En la barra de búsqueda, escriba "MCP".



3. Seleccione **Editar en settings.json** .

4. Agrega la configuración de MCP en Dev Mode de Figma a tus servidores:



5. Abre la barra de herramientas del chat usando `⌘B` o `⌘I` y cambia al modo **Agente** .

6. Abre el menú de herramientas de selección y busca una sección etiquetada como: **Servidor MCP: MCP en Dev Mode de Figma** . Si no aparecen herramientas, reinicie la aplicación de Figma para escritorio y VS Code.

Una vez configurado, actualice o inicie el servidor. Deberías ver una conexión exitosa y las herramientas disponibles. Si la conexión falló o no ves ninguna herramienta, verifica que el servidor esté activo en la aplicación de Figma para escritorio.

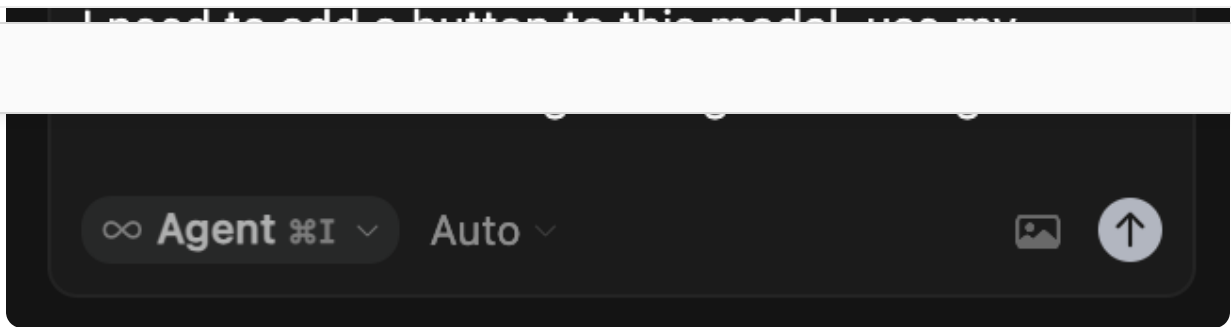
## Paso 3: Indica a tu cliente MCP

El servidor MCP en Dev Mode presenta un conjunto de herramientas que ayudan a los LLM a traducir diseños en Figma. Una vez conectado, puedes indicarle a tu cliente MCP que accede a un nodo de diseño específico.

Hay dos maneras de proporcionar contexto de diseño de Figma a tu cliente de IA:

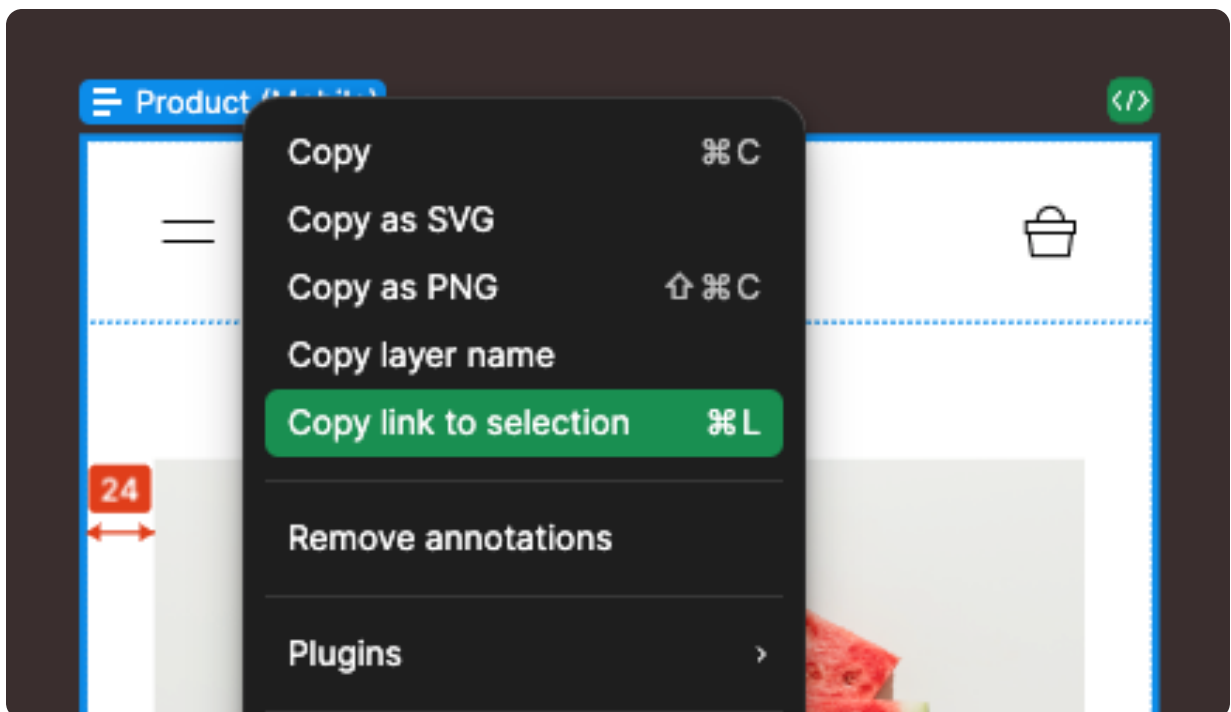
### Basado en la selección

1. Selecciona un marco o una capa dentro de Figma usando la aplicación de escritorio.
2. Solicita a tu cliente que te ayude a implementar tu selección actual.



## Basado en los enlaces

1. Copia el enlace a un marco o capa en Figma.



2. Solicita a tu cliente que te ayude a implementar el diseño en la URL seleccionada.

**Nota:** Tu cliente no podrá navegar a la URL seleccionada, pero extraerá el ID de nodo que el servidor MCP necesita para identificar sobre qué objeto debe devolver información.

`get_code`

Usa esto para generar un código para tu selección de Figma con el servidor MCP. La salida predeterminada es **React + Tailwind** , pero puedes personalizarla con tus indicaciones:

- Modifica el marco
  - "Genera mi selección de Figma en Vue".
  - "Genera mi selección de Figma en HTML simple + CSS".
  - "Genera mi selección de Figma en iOS".
- Usa tus componentes
  - "Genera mi selección de Figma usando componentes de src/components/u"
  - "Genera mi selección de Figma usando componentes de src/ui y estilo con Tailwind"

Puedes pegar enlaces o seleccionar el marco o componente en Figma antes de las indicaciones.

[Te mostramos cómo configurar Code Connect para optimizar la reutilización de los componentes →](#)

`get_variable_defs`

Devuelve las variables y estilos utilizados en tu selección, como colores, espaciado y tipografía.

- Enumera todos los tokens empleados
  - "Obtén las variables usadas en mi selección de Figma".
- Concéntrate en un tipo específico



Enumera los nombres de las variables y sus valores usados en la selección de

### `get_code_connect_map`

Recupera una asignación entre los ID de nodo de Figma y sus componentes de código correspondientes en tu código base. Específicamente, devuelve un objeto en que cada clave es un ID de nodo de Figma, y el valor contiene:

- `codeConnectSrc` : La ubicación del componente en tu código base (por ejemplo, una URL o ruta de archivo).
- `codeConnectName` : El nombre del componente en tu código base.

Esta asignación se utiliza para conectar los elementos de diseño de Figma con sus implementaciones en React (u otro marco), lo que permite flujos de trabajo de diseño a código sin interrupciones y asegura que se utilicen los componentes correctos para cada parte del diseño. Si un nodo de Figma está conectado a un componente de código, esta función te ayuda a identificar y utilizar el componente exacto en tu proyecto.

### `get_image`

Para usar esta herramienta, vaya a **Preferencias > Configuración del servidor MCP en Modo Dev > Activar herramienta `get_image`**

Esto toma una captura de pantalla de su selección para conservar la fidelidad de la disposición. Mantén esto activado, a menos que estés gestionando los límites de tokens.

## Configuración del servidor MCP en modo Dev

Estas son configuraciones adicionales que puedes activar o desactivar en Preferencias y usar con el cliente MCP

de posición genéricos en su lugar, lo que es útil si prefieres intercambiarlos

### Habilitar Code Connect

Incluye las asignaciones de Code Connect en la respuesta, de modo que el código generado pueda reutilizar los componentes de tu código base conectado siempre que sea posible.

A medida que utiliza el servidor MCP en Dev Mode, es posible que veas una ventana emergente dentro de Figma que te solicita comentarios. Para brindarlo, [usa este formulario](#).

## Prácticas recomendadas de MCP

La calidad del código generado depende de varios factores. Algunos controlados por ti y otros por las herramientas que estás utilizando. A continuación, encontrarás algunas sugerencias para obtener resultados organizados y consistentes.

### Organiza tu archivo de Figma para optimizar el código

Proporciona el mejor contexto para tu intención de diseño, de modo que el MCP y tu asistente de IA puedan generar un código que sea claro y consistente, y esté alineado con tu sistema.

- **Usa componentes** para cualquier cosa reutilizada (botones, tarjetas, entradas, etc.)
- **Vincula componentes a tu código base** a través de Code Connect. Esta es la mejor manera de lograr un uso consistente de componentes en el código. Sin eso, el

- Utilice la Disposición automática para comunicar la intención de diseño adaptable.

Consejo: Cambie el tamaño del marco en Figma para comprobar que se comporta como se espera antes de generar el código.

- Usa anotaciones y recursos de desarrollo para transmitir la intención del diseño que es difícil de capturar solo a partir de los elementos visuales, como el comportamiento, la alineación o la respuesta esperada.

## Escribe indicaciones efectivas para guiar a la IA

El MCP brinda a tu asistente de IA datos estructurados de Figma, pero tu indicación determina el resultado. Las buenas indicaciones pueden hacer lo siguiente:

- Alinear el resultado con tu marco o sistema de diseño.
- Siga las convenciones de nomenclatura y la estructura de los archivos.
- Agregar código a rutas específicas (por ejemplo, `src/components/ui` )
- Añadir o modificar el código en los archivos existentes en lugar de crear nuevos.
- Seguir sistemas de disposición específica (por ejemplo, cuadrícula, flexbox, absoluto).

### Ejemplos:

- "Genera código SwiftUI para iOS desde este marco"
- "Usa la UI de Chakra para esta disposición"
- "Usa componentes de `src/components/ui` "
- "Añade esto a `src/components/marketing/PricingCard.tsx` "

intención clara conduce a mejores resultados

## Activa herramientas específicas cuando sea necesario

El MCP es compatible con diferentes herramientas, y cada una brinda a tu asistente de IA un tipo diferente de contexto estructurado. A veces, el asistente no elige el correcto automáticamente, en especial porque hay más herramientas disponibles. Si los resultados no son correctos, intenta ser explícito en tu indicación.

- `get_code` brinda una representación estructurada de **React + Tailwind** de tu selección en Figma. Este es un punto de partida que tu asistente de IA puede traducir a cualquier marco o estilo de código, dependiendo de tu indicación.
- `get_variable_defs` extrae las variables y estilos utilizados en tu selección (color, espacio, tipografía, etc.). Esto ayuda al modelo a referenciar tus tokens en el código generado.

Por ejemplo, si estás recibiendo código sin procesar en lugar de tokens, intenta algo como:

- "Obtén los nombres y valores de las variables utilizadas en este marco".

## Agrega reglas personalizadas

Establece directrices a nivel de proyecto para mantener la consistencia del resultado, tal como las notas de incorporación para un nuevo desarrollador. Estas son cosas como:

- Primitivas de disposición preferidas
- Organización de archivos
- Patrones de nomenclatura
- Qué no se debe codificar

## Ejemplos:

Cursor

Código Claude

Reglas generales

Copiar

```
---
descripción: Reglas del MCP en Dev Mode de Figma
globs:
alwaysApply: true
---
- El servidor MCP en Dev Mode de Figma proporciona un extremo
- IMPORTANTE: Si el servidor MCP en Dev Mode de Figma devuelve
- IMPORTANTE: NO importes o agregues nuevos paquetes de iconos
- IMPORTANTE: NO uses ni crees marcadores de posición si se p
```

Agregar estos una vez puede reducir drásticamente la necesidad de indicaciones repetitivas y garantiza que tus compañeros de equipo o agentes sigan las mismas expectativas de forma consistente.

Asegúrate de revisar la documentación de tu IDE o cliente MCP para saber cómo estructurar las reglas y experimenta para encontrar qué funciona mejor para tu equipo. Una guía clara y consistente a menudo produce un código mejor y más reutilizable con menos idas y vueltas.

## Divide las selecciones grandes

Las selecciones grandes pueden ralentizar las herramientas, causar errores o provocar

modelo lo procese. En cambio:

1. Genera código para secciones más pequeñas o componentes individuales (por ejemplo, Tarjeta, Encabezado, Barra lateral).
2. Si te parece lento o se ataca, reduce el tamaño de la selección.

Esto ayuda a que el contexto sea manejable y los resultados, más previsibles, tanto para ti como para el modelo.

Si algo en el resultado no se ve del todo bien, generalmente, es útil revisar lo básico: cómo está estructurado el archivo de Figma, cómo está redactada la indicación y qué contexto se está enviando. Seguir las prácticas recomendadas mencionadas puede marcar una gran diferencia y, a menudo, conducir a un código más consistente y reutilizable.

¿Te resultó útil este artículo?

Si

No

Figma



Español (Latinoamérica)

|                      |   |                                 |                             |
|----------------------|---|---------------------------------|-----------------------------|
| usuario              | Características de creación de prototipos | practicar Apoyo Desarrolladores | Estudio Invision Enmarcador |
| Diseño UX            | Características de los sistemas de diseño | Aprende diseño Descargas        | Diseño en Windows Miró      |
| Prototipado          | Funciones de colaboración                 | Qué hay de nuevo Lanzamientos   |                             |
| Diseño gráfico       | FigJam                                    | Carreras                        |                             |
| Estructura alámbrica | Precios                                   | Sobre nosotros                  |                             |
| Reunión creativa     | Empresa                                   | Socios de la agencia            |                             |
| Plantillas           | Estudiantes y educadores                  | Privacidad                      |                             |
| Diseño remoto        | Cientes                                   | Estado                          |                             |
|                      | Seguridad                                 |                                 |                             |
|                      | Integraciones                             |                                 |                             |
|                      | Contacto                                  |                                 |                             |