



Guía académica para conectar Zotero local con Claude Desktop mediante zotero-mcp

Danny Murillo González

Investigador

Universidad Tecnológica de Panamá - CIDITIC

Universidad de Granada, España - Doctorando

CC BY-NC-SA 4.0

Atribución/Reconocimiento-NoComercial-Compartir Igual 4.0 Internacional

08/07/2026

1. Resumen

El documento es una **guía académica y funcional** para conectar una biblioteca local de Zotero con Claude Desktop mediante `zotero-mcp`. Su propósito principal es que investigadores, docentes, estudiantes, bibliotecólogos, documentalistas, técnicos y usuarios no técnicos puedan convertir Zotero en una fuente activa de consulta académica asistida por inteligencia artificial. La guía usa como ejemplo temático el **Procesamiento de Lenguaje Natural (NLP)**, aunque aclara que el procedimiento puede aplicarse a cualquier área del conocimiento.

La guía plantea como objetivo general conectar Zotero local con Claude Desktop para consultar documentos académicos, notas, metadatos, anotaciones y referencias bibliográficas desde una interfaz conversacional. Sus objetivos específicos incluyen explicar conceptos clave como Zotero, Claude Desktop, MCP, `zotero-mcp`, `uv`, `uvx`, Python, API local y base semántica; instalar las herramientas necesarias; activar la comunicación local de Zotero; crear una base semántica; configurar Claude Desktop; probar consultas académicas y orientar el uso de la integración para revisión de literatura, organización temática, generación de conocimiento y trazabilidad bibliográfica.

El documento está dirigido a varios perfiles: usuario investigador, técnico, no técnico, intermedio y novel. Esta clasificación es importante porque la guía no se limita a comandos, sino que también intenta explicar el valor metodológico de la integración. Para investigadores, la utilidad está en explorar literatura, preparar revisiones, organizar documentos y detectar vacíos de investigación; para usuarios técnicos, el énfasis está en rutas, comandos, archivos JSON y depuración; mientras que para usuarios noveles o no técnicos se ofrecen instrucciones paso a paso y capturas de pantalla.

En el marco conceptual, la guía explica que **Zotero** funciona como gestor bibliográfico y fuente documental; **Claude Desktop** como interfaz conversacional; **MCP** como protocolo que permite conectar modelos de IA con herramientas externas; y `zotero-mcp` como servidor especializado que conecta Claude con Zotero. También explica que `uv` permite instalar herramientas Python de forma sencilla y que `uvx` permite ejecutar `zotero-mcp` desde Claude.

2. Objetivo general

Conectar una biblioteca local de Zotero con Claude Desktop mediante `zotero-mcp`, para que el usuario pueda consultar documentos académicos, notas, metadatos, anotaciones y referencias bibliográficas desde una interfaz conversacional asistida por inteligencia artificial.

3. Objetivos específicos

1. Explicar los conceptos básicos necesarios: Zotero, Claude Desktop, MCP, `zotero-mcp`, `uv`, `uvx`, Python, API local y base semántica.
2. Instalar las herramientas necesarias para ejecutar `zotero-mcp`.
3. Activar y verificar la comunicación local de Zotero.
4. Crear una base semántica local para mejorar la búsqueda por significado.

5. Configurar Claude Desktop para que reconozca el servidor MCP de Zotero.
6. Probar consultas académicas sobre un tema abierto, usando como ejemplo Procesamiento de Lenguaje Natural.
7. Orientar el uso de la integración para revisión de literatura, organización temática, generación de conocimiento y trazabilidad bibliográfica.

4. Público objetivo

Esta guía está diseñada para varios perfiles de usuario.

- **Usuario investigador**
 - Es quien desea usar Zotero y Claude para explorar literatura científica, preparar revisiones, organizar documentos, construir marcos teóricos o identificar vacíos de investigación.
 - Necesita comprender no solo los comandos, sino también el valor metodológico de la integración.
- **Usuario técnico**
 - Es quien puede instalar herramientas, usar PowerShell, revisar rutas, configurar archivos JSON y depurar errores.
 - Necesita comandos claros, rutas, verificaciones y salidas esperadas.
- **Usuario no técnico**
 - Es quien usa Zotero como gestor bibliográfico, pero no está familiarizado con consola, Python, MCP ni archivos de configuración.
 - Necesita explicaciones conceptuales, instrucciones paso a paso y capturas de pantalla.
- **Usuario intermedio**
 - Es quien ya ha usado Zotero, Claude o herramientas de IA, pero necesita un procedimiento reproducible, verificable y adaptable a su tema de investigación.
- **Usuario novel**
 - Es quien solo tiene Zotero instalado y quiere conectar su biblioteca con Claude por primera vez.
 - Necesita saber qué instalar, dónde hacer clic, qué copiar, qué pegar y cómo verificar si todo funciona.

5. ¿Qué deseamos lograr con la guía?

Al terminar la guía, el usuario debería tener:

- Zotero Desktop abierto y configurado para comunicación local.
- uv y uvx instalados.
- zotero-mcp instalado.
- Base semántica creada o verificada.
- Claude Desktop configurado con un servidor MCP.
- Claude Desktop capaz de consultar a base de conocimiento de Zotero.

- Integración de Claude Desktop con Zotero o bien una respuesta en la que Claude indique que puede usar herramientas de Zotero para buscar documentos, notas, anotaciones o referencias.

6. Marco conceptual

6.1. ¿Qué es Zotero?

Zotero es un gestor bibliográfico que permite guardar, organizar, citar y recuperar referencias académicas almacenadas tanto en la versión de escritorio u online, si existe sincronización de las cuentas.

En Zotero se pueden almacenar:

- Artículos científicos
- libros
- capítulos
- tesis
- informes
- PDF
- notas
- anotaciones
- tags
- colecciones
- metadatos bibliográficos

En la práctica, deseamos utilizar una colección de Zotero sobre el tema de **Procesamiento de Lenguaje Natural**, el cual puede contener artículos sobre Modelos BERT, transformers, embeddings, clasificación de textos, análisis semántico, recuperación de información y modelos multilingües, por lo que podremos hacer búsqueda no solo del concepto principal sino de palabras semánticamente vinculadas a NLP.

6.2. ¿Qué es Claude Desktop?

Desktop Claude AI es es una **aplicación nativa de Anthropic** para macOS y Windows que lleva los modelos de inteligencia artificial a su computadora. Esta versión permite conectar a Claude con fuentes de datos, bases de datos y herramientas externas directamente en tu equipo utilizando **protocolo MCP (Model Context Protocol)**, además de la capacidad de interactuar con los recursos del sistema operativo, interpretar la pantalla, usar el teclado y ejecutar acciones en nombre del usuario dentro de los permisos configurados.

En esta integración:

- Claude Desktop actúa como interfaz conversacional.
- zotero-mcp actúa como puente.
- Zotero actúa como fuente bibliográfica.

6.3. ¿Qué es MCP?

MCP significa **Model Context Protocol**, es un estándar abierto creado por **Anthropic** en 2024 para conectar los modelos de inteligencia artificial con datos y herramientas externas. Funciona de manera similar a un puerto USB-C para la IA:¿, es decir, ofrece una conexión universal para que un asistente o modelo de lenguaje (como Claude o ChatGPT) pueda leer archivos, consultar bases de datos, archivos o usar aplicaciones sin necesidad de programar una integración única para cada sistema.

En esta guía, utilizaremos MCP a través de **zotero-mcp**, este permite que Claude Desktop se conecte con Zotero local, siendo **MCP** el mecanismo de comunicación para que Claude use herramientas externas.

6.4. ¿Qué es zotero-mcp?

El componente **zotero-mcp** es un servidor MCP especializado en Zotero, donde su función es permitir que Claude consulte la biblioteca de Zotero.

Con **zotero-mcp**, Claude puede acceder a información de Zotero, como:

- Títulos
- autores
- años
- DOI
- resúmenes
- tags
- notas
- anotaciones
- colecciones
- metadatos

En términos prácticos:

- Zotero guarda la biblioteca.
- zotero-mcp permite que Claude consulte la biblioteca.
- Claude recibe e interpreta la pregunta del usuario.
- Claude devuelve una respuesta según datos de zotero-mcp.

6.5. ¿Qué es uv?

uv es una herramienta para instalar y administrar paquetes de Python.

Se utiliza en esta guía porque **zotero-mcp** está desarrollado en Python y puede instalarse de forma sencilla mediante **uv**.

Con **uv**, el usuario puede instalar herramientas sin tener que crear manualmente un entorno Python complejo.

6.6. ¿Qué es uvx?

uvx es una utilidad incluida con **uv**. Sirve para ejecutar herramientas Python directamente. En esta guía, Claude puede usar **uvx** para ejecutar **zotero-mcp**.

Ejemplo conceptual:

```
Claude ejecuta uvx.
uvx ejecuta zotero-mcp.
zotero-mcp consulta Zotero.
Claude recibe los resultados.
```

6.7. ¿Es necesario instalar Python?

En esta guía no es necesario. La ruta recomendada de esta guía usa **uv**, por lo que no es obligatorio instalar Python manualmente al inicio.

Sin embargo, puede ser necesario instalar Python si:

- uv no se instala correctamente.
- El usuario quiere ejecutar scripts adicionales.
- Se desea desarrollar una aplicación propia.
- Se quiere trabajar con RAG, embeddings o procesamiento de datos fuera de Claude.

Para usuarios noveles, se recomienda intentar primero con **uv**.

Para usuarios técnicos o intermedios, se puede instalar Python.

6.8. ¿Qué es la API local de Zotero?

Zotero Desktop puede exponer una API local en la computadora.

Normalmente se encuentra en:

```
http://localhost:23119/api/
```

Esta API permite que otras aplicaciones del mismo equipo consulten la biblioteca de Zotero. Para que funcione:

- Zotero debe estar abierto.
- La comunicación local debe estar activada.
- El puerto local debe responder correctamente.

6.9. ¿Qué es la API Web de Zotero?

La API Web de Zotero permite acceder a la biblioteca sincronizada en zotero.org.

Puede requerir:

```
ZOTERO_API_KEY
ZOTERO_LIBRARY_ID
ZOTERO_LIBRARY_TYPE
```

En esta guía, la API Web no se utiliza, utilizaremos todas las colecciones de la Biblioteca de Zotero en modo local.

La API Web se deja como opción avanzada para usuarios que necesiten sincronización web, acceso remoto o modo híbrido.

6.10. ¿Qué es una base semántica?

Una **base de datos semántica** (o base de datos vectorial) es un sistema que almacena información basada en el significado y el contexto de los datos, en lugar de usar solo palabras clave exactas.

Significado (Embeddings): Convierte textos, código o fragmentos de información en números (vectores) que representan conceptos.

Búsqueda por similitud: Permite encontrar información relacionada por el sentido o la intención de la consulta, no por coincidencias literales de letras.

En esta guía, **zotero-mcp** puede crear una base semántica a partir de los metadatos de documentos integrados a la biblioteca de Zotero.

Normalmente al crear esta **base semántica**, se guarda en una ruta similar a:

```
C:\Users\USUARIO\.config\zotero-mcp\chroma_db
```

Se crea para:

- Mejorar la búsqueda temática
- Encontrar documentos relacionados aunque usen palabras distintas
- Recuperar literatura por significado
- Facilitar consultas académicas más flexibles
- Apoyar revisiones de literatura
- Conectar preguntas del usuario con documentos relevantes.

Ejemplo aplicado a Procesamiento de Lenguaje Natural:

Si el usuario pregunta:

```
Busca documentos sobre representación semántica de textos.
```

La búsqueda semántica puede recuperar documentos sobre:

- embeddings

- sentence transformers
- BERT
- representación contextual
- semantic similarity
- text classification
- information retrieval

Aunque el título del documento no contenga exactamente la frase **representación semántica de textos**.

7. Enfoque metodológico de la guía

Esta guía propone una forma de trabajo académico con la integración **Zotero + Claude + zotero-mcp** puede apoyar procesos como:

- Exploración de literatura
- Organización temática de documentos
- Identificación de conceptos
- Comparación de enfoques
- Elaboración de estados del arte
- Búsqueda de autores y documentos clave
- Detección de vacíos de investigación
- Generación de preguntas de investigación
- Síntesis preliminar de evidencia
- Trazabilidad bibliográfica.

8. Tema de ejemplo de la guía

Para la guía, se usará como ejemplo contenidos en la biblioteca de Zotero relacionado con el tema, **Procesamiento de Lenguaje Natural**.

Ejemplos de preguntas:

- Busca documentos sobre modelos BERT y clasificación de textos.
- Agrupa mi literatura de Zotero sobre NLP en temas como embeddings, transformers, recuperación de información y análisis semántico.
- ¿Qué documentos de mi biblioteca explican la diferencia entre BERT, SciBERT y Sentence-BERT?
- Identifica documentos sobre embeddings aplicados a literatura científica.

El usuario puede reemplazar este tema por cualquier otro basado en su colección.

Parte II. Instalación y configuración paso a paso

Requisitos previos

Antes de iniciar, el usuario debe verificar:

- Windows 10 o Windows 11
- Cuenta de Zotero y Claude
- Zotero Desktop instalado
- Claude Desktop instalado
- PowerShell
- Conexión a internet
- Biblioteca de Zotero con referencias.

Opcional:

- Python instalado
- Cuenta sincronizada (opcional)
- API key de Zotero

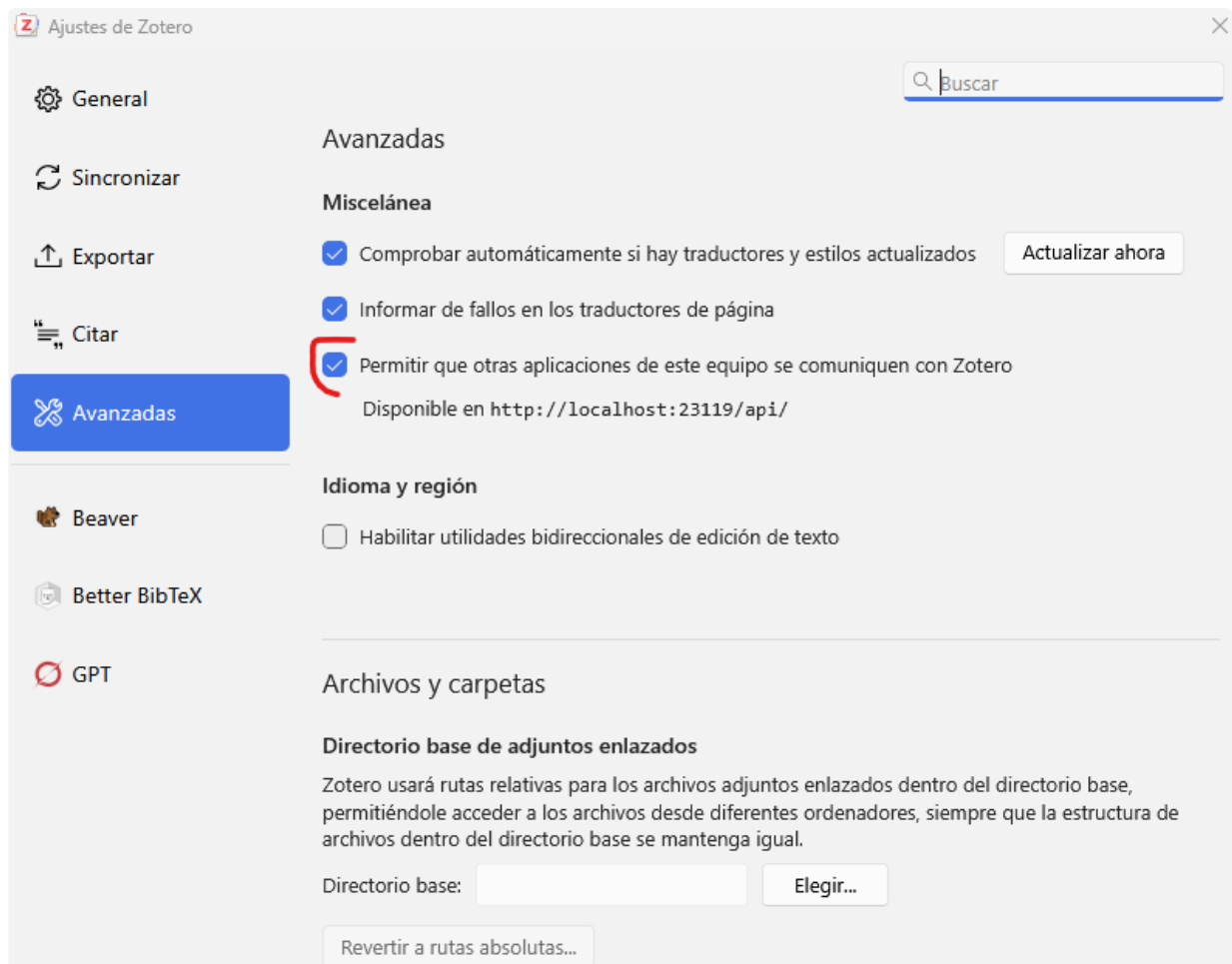
Paso 1: activar la comunicación local en Zotero

Abrir la aplicación Zotero Desktop.

Ir al menú:

Editar → Ajustes → Avanzadas

En la pestaña de **Avanzadas**, seleccione en **Misceláneas**, *Permitir que otras aplicaciones de este equipo se comuniquen con Zotero*.



Esta opción permite que la librería de Phyon que vamos a instalar, **zotero-mcp**, consulte la biblioteca local de Zotero y los documentos que se encuentran en todas las colecciones.

Paso 2: Verificar que Zotero responde localmente

Para verificar que Zotero puede responder localmente a otras aplicaciones debemos verificar si la API de Zotero esta activa o **corriendo localmente**.

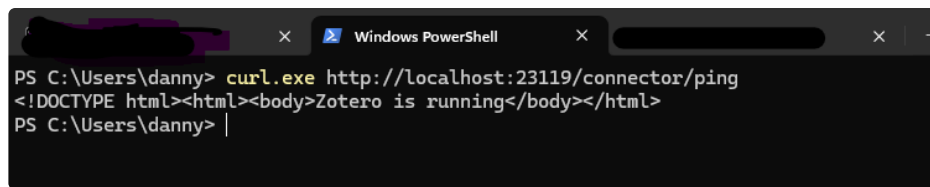
Abrir la aplicación **Windows PowerShell** de su computador.



Ejecutar:

```
curl.exe http://localhost:23119/connector/ping
```

Respuesta:



```
PS C:\Users\danny> curl.exe http://localhost:23119/connector/ping
<!DOCTYPE html><html><body>Zotero is running</body></html>
PS C:\Users\danny> |
```

Si al ejecutar el código anterior, el resultado que se muestra esta relacionado con **Failed...**, puede indicar dos cosas:

1. No se habilitó la opción de configuración del paso 10
2. Zotero no esta corriendo, Zotero debe estar abierto.

```
curl: (7) Failed to connect to localhost:23119 after 2245 ms: Could not connect to server
```

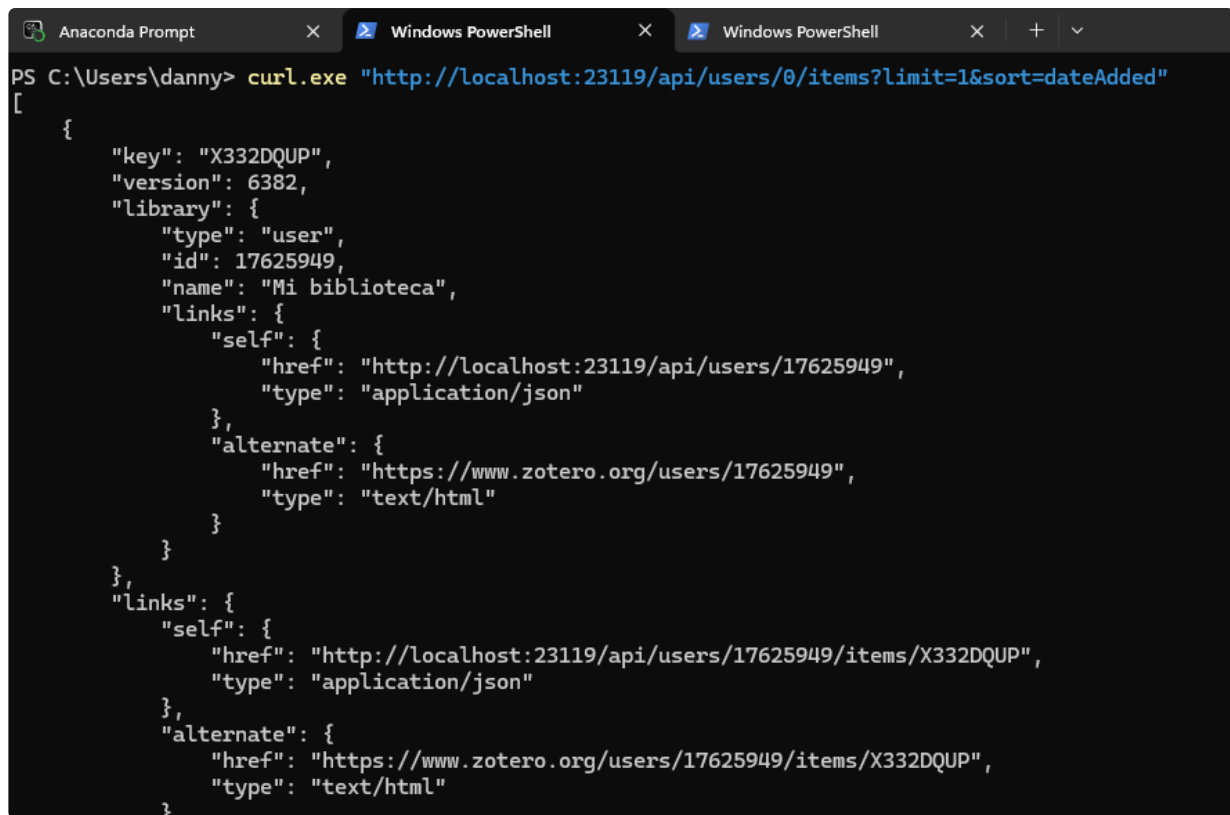
Paso 3: probar la API local de Zotero

En la aplicación **Windows PowerShell**, ejecutar:

```
curl.exe "http://localhost:23119/api/users/0/items?limit=1&sort=dateAdded"
```

Resultado:

Se debe mostrar una estructura en formato JSON con datos de una referencia. Esto indica que la API local funciona y confirma que Zotero no solo responde, sino que puede devolver elementos bibliográficos.



```

PS C:\Users\danny> curl.exe "http://localhost:23119/api/users/0/items?limit=1&sort=dateAdded"
[
  {
    "key": "X332DQUP",
    "version": 6382,
    "library": {
      "type": "user",
      "id": 17625949,
      "name": "Mi biblioteca",
      "links": {
        "self": {
          "href": "http://localhost:23119/api/users/17625949",
          "type": "application/json"
        },
        "alternate": {
          "href": "https://www.zotero.org/users/17625949",
          "type": "text/html"
        }
      }
    },
    "links": {
      "self": {
        "href": "http://localhost:23119/api/users/17625949/items/X332DQUP",
        "type": "application/json"
      },
      "alternate": {
        "href": "https://www.zotero.org/users/17625949/items/X332DQUP",
        "type": "text/html"
      }
    }
  }
]

```

Paso 4: instalar uv

En la aplicación **Windows PowerShell**, ejecutar:

```
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

Cerrar PowerShell y abrirlo nuevamente.

Verificar la instalación escribiendo:

```
where.exe uv
where.exe uvx
```

Resultado:

```

C:\Users\danny>where.exe uv
C:\Users\danny\.local\bin\uv.exe

C:\Users\danny>where.exe uvx
C:\Users\danny\.local\bin\uvx.exe

```

Sino se muestra la ruta de de **uv** y **uvx**, es porque PowerShell está cargando una variable de entorno diferente, se puede verificar o corregir, pero como el problema es solo la ruta (PATH), podemos utilizar este código para la verificación:

```
$env:Path = "$env:USERPROFILE\.local\bin;$env:Path"
```

Verificar:

```
where.exe uv
where.exe uvx

o

uv --version
ux --version
```

Resultado esperado:

```
PS C:\Users\danny> where.exe uv
C:\Users\danny\uv
C:\Users\danny\.local\bin\uv.exe
PS C:\Users\danny> uv --version
uv 0.11.28 (ebf0f43d7 2026-07-07 x86_64-pc-windows-msvc)
PS C:\Users\danny> uvx --version
uvx 0.11.28 (ebf0f43d7 2026-07-07 x86_64-pc-windows-msvc)
PS C:\Users\danny> where.exe uvx
C:\Users\danny\.local\bin\uvx.exe
PS C:\Users\danny> |
```

Paso 5: instalar zotero-mcp

Instalar el componentes **zotero-mcp** con soporte semántico para la comunicación entre Zotero y Claude.

En la aplicación **Windows PowerShell**, ejecutar:

```
uv tool install "zotero-mcp-server[semantic]"
```

Resultado:

```
PS C:\Users\danny> uv tool install "zotero-mcp-server[semantic]"
Resolved 166 packages in 4.00s
Uninstalled 3 packages in 121ms
- ebooklib==0.20
- lxml==6.1.1
- pymupdf==1.27.2.3
Installed 2 executables: zotero-cli, zotero-mcp
```

Verificar:

```
where.exe zotero-mcp
```

Resultado:

```
C:\Users\USUARIO\.local\bin\zotero-mcp.exe
```

Probar:

```
zotero-mcp --help
```

Se debe mostrar:

```
PS C:\Users\danny> zotero-mcp --help
usage: zotero-mcp [-h] {serve,setup,update-db,openai-batch-status,openai-batch-import,db-status,db-inspect,update}

Zotero Model Context Protocol server

positional arguments:
  {serve,setup,update-db,openai-batch-status,openai-batch-import,db-status,db-inspect,update,version,setup-info}
    Command to run
  serve                Run the MCP server
  setup                Configure zotero-mcp (Claude Desktop or standalone)
  update-db            Update semantic search database
  openai-batch-status  Show OpenAI Batch API status
  openai-batch-import  Import completed OpenAI batch embeddings
  db-status            Show semantic search database status
  db-inspect           Inspect indexed documents or show aggregate stats for the semantic DB
  update              Update zotero-mcp to the latest version
  version              Print version information
  setup-info           Show installation path and configuration info for MCP clients

options:
  -h, --help            show this help message and exit
```

Consultar estado de base semántica

Al ejecutar el siguiente código, sirve para **consultar el estado de la base semántica local** que usa **zotero-mcp** y cuántos documentos tiene registrados en esa base de la colección de Zotero. En la aplicación **Windows PowerShell**, ejecutar:

```
zotero-mcp db-status
```

La primera vez la respuesta va a ser la siguiente, significa que la base existe, pero todavía **no tiene documentos indexados**.

```
Document count: 0
```

Paso 6: Crear la base semántica

Con Zotero abierto, ejecutar primero una prueba con pocos documentos.
significado del código:

zotero-mcp servicio a ejecutar

update-db crear o actualizar base de datos semántica

--limit 100 número de documentos a evaluar, valor mayor de 1.

En la aplicación **Windows PowerShell**, ejecutar:

```
zotero-mcp update-db --limit 100
```

Este comando crea una base semántica inicial con una muestra de documentos.

Se crea para que Claude pueda buscar en Zotero por significado y no solo por palabras exactas.

Resultado:

```
PS C:\Users\danny> zotero-mcp update-db --limit 100
Using configuration: C:\Users\danny\.config\zotero-mcp\config.json
Starting database update...

Fetching fulltext for 100 items via web API...
Fulltext: 100/100 items checked, 85 with text

Indexing 100 items...

Done: 100 indexed, 0 skipped, 0 errors

Database update completed:
- Total items: 100
- Processed: 100
- Added: 40
- Updated: 60
- Skipped: 0
- Errors: 0
- Duration: 0:09:59.887136
```

Significado:

Revisó **100 documentos** e intentó obtener texto completo o contenido textual asociado. De esos 100, encontró texto utilizable en **85 documentos**.

Esto significa que no todos los registros tienen texto completo disponible. Algunos pueden tener solo metadatos, título, autores, DOI o resumen.

Indexó correctamente los 100 ítems y no omitió ninguno tampoco hubo errores. Además en este ejemplo se muestra, **Added: 40**, quiere decir que agregó 40 ítems nuevos a la base semántica, esto se debe ha que en la práctica ya yo había generado una base semántica.

En la aplicación **Windows PowerShell**, ejecutar, para revisar resultados:

```
zotero-mcp db-status
```

Si aparece, 100, la indexación inicial funcionó:

```
Document count: 100
```

Resultados:

Se muestra, **Document count: 365**, porque en mi caso, ya la base semántica se había generado

pero el resultado en pantalla debe ser similar, solo que varía el número de documentos.

```
PS C:\Users\danny> zotero-mcp db-status
=== Semantic Search Database Status ===
Collection: zotero_library
Document count: 365
Embedding model: default
Database path: C:\Users\danny\.config\zotero-mcp\chroma_db

Update configuration:
- Auto update: True
- Frequency: startup
- Last update: 2026-07-14T13:41:40.669124
- Should update: True
- OpenAI Batch API: inactive
```

Indexar toda la biblioteca de Zotero

Como hemos verificado que si existen al menos 100 documentos en su colección en Zotero, procederemos a indexar todos los documentos de la Biblioteca de Zotero, utilizando el siguiente comando:

En la aplicación **Windows PowerShell**, ejecutar:

```
zotero-mcp update-db --force-rebuild
```

Luego verificar:

```
zotero-mcp db-status
```

Resultado:

Se muestra la ruta de la base de datos semántica generada.

Database path: **C:\Users\danny.config\zotero-mcp\chroma_db**

```
PS C:\Users\danny> zotero-mcp db-status
=== Semantic Search Database Status ===
Collection: zotero_library
Document count: 0
Embedding model: default
Database path: C:\Users\danny\.config\zotero-mcp\chroma_db

Update configuration:
- Auto update: True
- Frequency: startup
- Last update: 2026-07-14T13:41:40.669124
- Should update: True
- OpenAI Batch API: inactive
```

Explicación de la creación de la base semántica

Crear la base semántica significa transformar la información bibliográfica de Zotero en representaciones vectoriales o computacionales que facilitan la búsqueda conceptual.

En lugar de depender solo de coincidencias textuales exactas, la base semántica ayuda a recuperar documentos cercanos en significado.

Ejemplo:

Pregunta: documentos sobre modelos contextuales en NLP.

La base semántica puede recuperar documentos no solo que contenga la palabra de búsqueda principal NLP, sino de temas relacionados con este término, como:
BERT, transformers, contextual embeddings, language models, SciBERT, RoBERTa.

Paso 7: instalar Python (si es necesario)

Este paso es **opcional**, si lo anterior funcionó al instalar **zotero-mcp**, se omite.

Descargar Python desde el sitio oficial.

Durante la instalación, marcar:

```
Add python.exe to PATH
```

Después de instalar, cerrar PowerShell y abrirlo nuevamente.

Verificar:

```
python --version
```

También:

```
where.exe python
```

Paso 8: Configurar Claude Desktop

Claude Desktop debe saber cómo ejecutar **zotero-mcp**, para configurarlo debes **cerrar Claude completamente**.

En **Windows PowerShell** ejecutar:

```
taskkill /IM Claude.exe /F
```

Al ejecutar el código puedes obtener como respuesta:

```
ERROR: no se encontró el proceso "Claude.exe".
```

El mensaje puede mostrarse porque no se instaló **CLAUDE Desktop** o que el proceso o aplicación **CLAUDE** ya estaba cerrada. Si es el segundo, omita el error.

Buscar el archivo de configuración de **CLAUDE Desktop**.

En **Windows PowerShell** ejecutar:

```
Get-ChildItem "$env:APPDATA\Claude", "$env:LOCALAPPDATA\Packages" -Recurse -Filter
"claude_desktop_config.json" -ErrorAction SilentlyContinue |
Select-Object FullName, LastWriteTime, Length
```

Puede aparecer una ruta o dos.

FullName	LastWriteTime
C:\Users\danny\AppData\Roaming\Claude\claude_desktop_config.json	06/28/2026 ...
C:\Users\danny\AppData\Local\Packages\Claude_pzs8srxjxfjfc\LocalCache\Roaming\Claude\claude_desktop_config.json	06/29/2026 ...

Esta es la ruta posible del archivo de configuración **.json**. *Si aparecen dos rutas, conviene configurar ambas con el mismo contenido.*

```
C:\Users\danny\AppData\Local\Packages\Claude_pzs8srxjxfjfc\LocalCache\Roaming\Claude\claude
_desktop_config.json
```

Paso 9: escribir la configuración JSON de Claude

Abrir el primer archivo en **Windows PowerShell** ejecutar:

```
notepad "$env:APPDATA\Claude\claude_desktop_config.json"
```

Colocar este JSON, cambiando **USUARIO** por el usuario real de Windows en su computador:

```
{
  "mcpServers": {
    "zotero": {
      "command": "C:\\Users\\USUARIO\\.local\\bin\\uvx.exe",
      "args": [
        "zotero-mcp"
      ],
      "env": {
        "ZOTERO_LOCAL": "true",
        "NO_PROXY": "localhost,127.0.0.1,::1"
      }
    }
  }
}
```

El contenido del archivo debe mostrarse como:

```
{
  "mcpServers": {
    "zotero": {
      "command": "C:\\Users\\danny\\.local\\bin\\uvx.exe",
      "args": [
        "zotero-mcp"
      ],
      "env": {
        "ZOTERO_LOCAL": "true",
        "NO_PROXY": "localhost,127.0.0.1,::1"
      }
    }
  }
}
```

Validar el JSON:

```
Get-Content "$env:APPDATA\Claude\claude_desktop_config.json" -Raw | ConvertFrom-Json | Out-Null
```

Si no aparece ningún error, el JSON es válido.

Abrir y modificar el segundo archivo (solo necesario si hay dos archivos .json):

```
notepad
"$env:LOCALAPPDATA\Packages\Claude_pzs8srxrjxfjjc\LocalCache\Roaming\Claude\claude_desktop_config.json"
```

Este es el archivo de configuración completo de **CLAUDE**, por lo que tendrá código de su configuración, adicional al .JSON anterior. El código a insertar debe estar al inicio, verificar los paréntesis y la coma al final para darle continuidad al código .JSON de configuración.

```
{
  "mcpServers": {
    "zotero": {
      "command": "C:\\Users\\danny\\.local\\bin\\uvx.exe",
      "args": [
        "zotero-mcp"
      ],
      "env": {
        "ZOTERO_LOCAL": "true",
        "NO_PROXY": "localhost,127.0.0.1,::1"
      }
    }
  },
  "preferences": {
    "remoteToolsDeviceName": "dannymu",
    "coworkHipaaRestricted": false,
    "coworkWebSearchEnabled": true,
    "coworkScheduledTasksEnabled": false,
    "ccdScheduledTasksEnabled": false,
    "epitaxyPrefs": {
      "starred-local-code-sessions": [],
      "starred-cowork-spaces": [],
      "starred-session-groups": [],
      "dframe-local-slice": {
        "pinnedOrder": [],
        "customGroupAssignments": {},
        "customGroupOrder": {}
      },
      "ccd-sessions-filter": {
        "state": {
          "selectedProjects": []
        },
        "version": 0
      },
      "desktop-frame.paneStore.v1": {
        "state": {
          "extraPanelsByMode": {},
          "colWeightsByMode": {}
        }
      }
    }
  }
}
```

Validar el JSON

Get-Content

"\$env:LOCALAPPDATA\Packages\Claude_pzs8sxnjxfjjc\LocalCache\Roaming\Claude\claude_desktop_config.json" -Raw | ConvertFrom-Json | Out-Null

Si no aparece ningún error, el JSON es válido.

Paso 10: Configurar dos rutas de Claude si existen (opcional)

Solo se realiza si:

- se detectaron dos archivos **claude_desktop_config.json**
- sino se logro la configuración del paso 9

- si se quiere escribir la misma configuración en ambos archivos.

En **Windows PowerShell** ejecutar:

```
taskkill /IM Claude.exe /F

$cfg1 = "$env:APPDATA\Claude\claude_desktop_config.json"
$cfg2 =
"$env:LOCALAPPDATA\Packages\Claude_pzs8sxnjxfjfc\LocalCache\Roaming\Claude\claude_desktop_c
onfig.json"

$json = @'
{
  "mcpServers": {
    "zotero": {
      "command": "C:\\Users\\USUARIO\\.local\\bin\\uvx.exe",
      "args": [
        "zotero-mcp"
      ],
      "env": {
        "ZOTERO_LOCAL": "true",
        "NO_PROXY": "localhost,127.0.0.1,::1"
      }
    }
  }
}
'@

foreach ($cfg in @($cfg1, $cfg2)) {
  New-Item -ItemType Directory -Force (Split-Path $cfg -Parent) | Out-Null
  Copy-Item $cfg "$cfg.backup_$(Get-Date -Format yyyyMMdd_HH:mm:ss)" -Force -ErrorAction
  SilentlyContinue
  [System.IO.File]::WriteAllText($cfg, $json, [System.Text.UTF8Encoding]::new($false))
  Write-Host "Actualizado: $cfg"
}
```

Cambiar:

```
USUARIO en la ruta
"command": "C:\\Users\\USUARIO\\.local\\bin\\uvx.exe"

según la ruta real de su equipo.
```

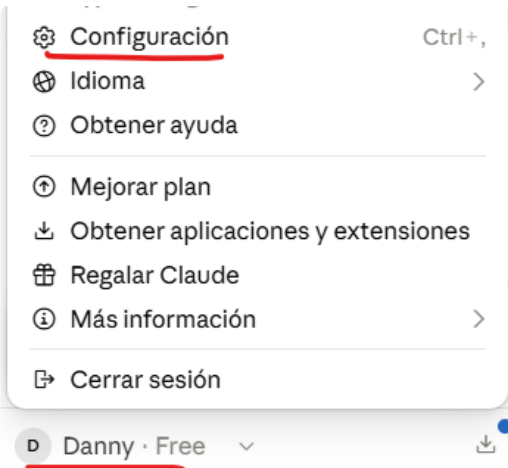
Paso 11: abrir Claude y confirmar el servidor MCP

Antes de abrir Claude, confirmar que Zotero sigue respondiendo:

```
curl.exe http://localhost:23119/connector/ping
```

Abrir **Claude Desktop**.

Nombre de usuario (parte inferior derecha) > **Configuración**



Se abrirá una ventana

Aplicación de Escritorio > Desarrollador > Servidores MCP locales

Debe aparecer en la configuración, **Zotero**, argumento: **zotero-mcp**:



También se puede revisar configuración desde:

Nombre de usuario (parte inferior derecha) > **Configuración**

Personalizar > Conectores

Se debe mostrar, **zotero**.

Configuración

General

Cuenta

Privacidad

Facturación

Capacidades

Claude Code

Aplicación de escritorio

General

Extensiones

Desarrollador

Personalizar

Habilidades

Conectores

Plugins

Todo

Conectado

No conectado

POPULAR

Gmail

Conectar

Google Drive

Conectar

Slack

Conectar

Conector	Tipo	Estado
Integración de GitHub	Web	✓
z zotero	Escritorio Dev local	✓
Google Drive	Web	Conectar

Paso 12: Probar Claude con Zotero

Hasta aquí, son los pasos técnicos, necesario para la integración de estas herramientas.

Abrir **CLAUDE Desktop**

Crear un Nuevo **Proyecto**

nombre: **mcp-claude-zotero**

descripción: Integración de documentos de Zotero en CLAUDE utilizando MCP

Abrir un **chat** nuevo en Claude.

Escribir:

¿Tienes herramientas MCP de Zotero disponibles? Lista las herramientas que puedes usar.

La respuesta puede ser:

¿Tienes herramientas MCP de Zotero disponibles? Lista las herramientas que puedes usar.

11:57 a. m.   

Searched available tools

Sí, tengo acceso al conector de Zotero, con estas tres herramientas:

1. `zotero_search_items` — Busca ítems en tu biblioteca de Zotero a partir de una consulta. Permite elegir el modo de búsqueda (por título/autor/año, o en todo el contenido) y filtrar por etiquetas (con búsquedas booleanas).
2. `zotero_item_metadata` — Obtiene los metadatos de un ítem específico (autor, título, fecha, tipo de publicación, etc.) a partir de su clave de ítem.
3. `zotero_item_fulltext` — Obtiene el texto completo de un ítem (o de un adjunto específico), a partir de su clave.

El flujo típico sería: primero buscar con `zotero_search_items` para encontrar el ítem que te interesa, y luego usar su clave (`item_key`) para consultar metadatos o texto completo.

Nota

Los siguientes ejemplos están basados en consultas temáticas según el grupo de documentos que existen en mi colección, los cuales están relacionados con, IA, NLP, BERT, embedding, por lo que los ejemplos son solo para mostrar los resultados de la integración de mcp-zotero con Claude. Deberías hacer consultas en función del grupo de documentos de su colección.

Ejemplo 1 de consulta temática:

Busca en mi Zotero documentos sobre procesamiento de lenguaje natural, BERT, embeddings y clasificación de textos científicos.

Resultado:

He encontrado varios resultados relevantes en tu biblioteca de Zotero. Muchos son adjuntos PDF sin metadatos visibles en la búsqueda (aparecen como "PDF" o "Texto completo"), pero destaco los ítems con metadatos claros y que están directamente relacionados con tus temas:

Procesamiento de Lenguaje Natural

- *Procesamiento de Lenguaje Natural con Deep Learning para el Análisis de Perfiles Profesionales en Consejos de Administración* — Antón, Manuel Astorgano (JournalArticle) — clave `LFL4XVVI`
- Nota: "Metodologías de Clustering con BERT" — clave `CP9MU2QV`

BERT / SciBERT

- *SciBERT: A Pretrained Language Model for Scientific Text* — Beltagy, Lo, Cohan (2019, ACL) — clave `SIUCDS2T`
- *Efficient Context-Limited Telescope Bibliography Classification for the WASP-2025 Shared Task Using SciBERT* — Naidu, Madhusudhan (2025, ACL) — clave `YYZIWJYI`
- *Fine-tuning SciBERT to enable ASJC-based assessments of the disciplinary orientation of research collections* — Gusenbauer, Endermann, Huber, et al. (04/2026, Scientometrics) — clave `43R7XADE`

Ejemplo 2 de consulta temática:

Muéstrame información estructurada de los documentos sobre Clasificación de textos científicos, titulo, resumen, revista, año, autores, ordenados por año

Resultado

Aquí tienes la información estructurada de los documentos sobre **clasificación de textos científicos**, ordenados por año:

1. SciBERT: A Pretrained Language Model for Scientific Text (2019)

- **Autores:** Beltagy, Iz; Lo, Kyle; Cohan, Arman
- **Revista/Fuente:** Association for Computational Linguistics (conferencePaper, DOI 10.18653/v1/D19-1371)
- **Resumen:** No hay resumen guardado en el registro de Zotero (solo metadatos e identificadores).
- **Clave:** SIUCDS2T

2. Hierarchical text classification for web of science scientific fields (2024)

- **Autores:** Jahani Rad, Pouyan; Bahaghighat, Mahdi
- **Revista:** *Facta universitatis - series: Electronics and Energetics*, Vol. 37, No. 4, pp. 703-732
- **Resumen:** El estudio se centra en construir un clasificador de texto eficiente para asignar corpus a campos científicos específicos, según las categorías de Web of Science (WOS), usando arquitecturas de deep learning como CNN, DNN y RNN en niveles jerárquicos (padre-hijo). El modelo alcanzó un F1 de 94.99% y una precisión de 99.33%, combinando CNN y DNN de forma jerárquica.

Ejemplo 3 de consulta temática:

Indicame la relevancia de estos documentos según su objetivo, metodología y resultados, evalúa el texto completo de estos títulos

Resultado

Con la información obtenida (incluyendo el texto completo de dos artículos), aquí tienes el análisis de relevancia de los cuatro documentos según **objetivo, metodología y resultados**, ordenados por año:

1. SciBERT: A Pretrained Language Model for Scientific Text (2019) — SIUCDS2T

- Objetivo:** Crear un modelo de lenguaje preentrenado basado en BERT para abordar la falta de datos científicos etiquetados de alta calidad y a gran escala, mejorando el rendimiento en tareas de PLN científico.
- Metodología:** SciBERT sigue la misma arquitectura que BERT pero se preentrena sobre texto científico, usando una muestra aleatoria de 1.14 millones de artículos de Semantic Scholar (18% de ciencias de la computación y 82% de biomedicina), empleando el texto completo de los artículos y no solo los resúmenes. Se evaluó en cinco tareas núcleo: reconocimiento de entidades, extracción PICO, clasificación de texto, clasificación de relaciones y análisis de dependencias.
- Resultados:** SciBERT supera a BERT-Base en tareas científicas (+2.11 F1 con ajuste fino y +2.43 F1 sin él), logrando nuevos resultados estado del arte en varias de estas tareas.
- Relevancia:** Es el documento **fundacional** del conjunto — establece el modelo base (SciBERT) que los otros dos trabajos de 2025 y 2026 utilizan y afinan. Máxima relevancia como referencia metodológica común.

Parte III. Uso académico de la integración

Ejemplos de Prompts para investigación

Ejemplo 1. Búsqueda temática

Busca en mi Zotero documentos relacionados con procesamiento de lenguaje natural, embeddings, transformers, BERT, clasificación de textos y recuperación de información. Agrupa los resultados por tema e indica título, autores, año, item key y justificación temática.

Resultado

Con todas las búsquedas realizadas (PLN, BERT, embeddings, transformers, clasificación de textos científicos y recuperación de información), aquí tienes los resultados agrupados por tema:

★ Modelos de lenguaje preentrenados (BERT / Transformers)

Título	Autores	Año	Item Key	Justificación temática
SciBERT: A Pretrained Language Model for Scientific Text	Beltagy, Lo, Cohan	2019	SIUCDS2T	Modelo BERT adaptado a dominio científico; base técnica de la mayoría de los demás trabajos.
SPECTER: Document-level Representation Learning using Citation-informed Transformers	Cohan, Feldman, Beltagy, Downey, Weld	2020	ZSBTWQEU (duplicado: 582FUITG)	Usa Transformers para generar representaciones (embeddings) de documentos científicos a partir de señales de citación.

Ejemplo.2. Revisión de literatura

Usa mi biblioteca de Zotero para construir una revisión preliminar de literatura sobre Procesamiento de Lenguaje Natural. Organiza los documentos por líneas temáticas, identifica autores clave, conceptos principales, métodos frecuentes y vacíos de investigación.

Resultado

1. Modelos de lenguaje preentrenados para dominio científico

Documentos: SciBERT ([SIUCDS2T](#) , 2019) · Fine-tuning SciBERT para ASJC ([43R7XADE](#) , 2026) · Clasificación bibliográfica de telescopios con SciBERT ([YYZIWJYI](#) , 2025)

Esta línea traza la evolución de un mismo modelo base —**SciBERT**, desarrollado por Beltagy, Lo y Cohan (Allen Institute for AI)— desde su creación hasta aplicaciones especializadas. SciBERT nació para resolver la escasez de datos etiquetados de calidad en dominios científicos, preentrenándose sobre 1.14M de artículos de Semantic Scholar. Sus derivados posteriores lo afinan (*fine-tuning*) para tareas concretas: clasificación disciplinar a gran escala (307 categorías ASJC) y clasificación bibliográfica de nicho (telescopios, 4 categorías), ambos limitados por la ventana de contexto de 512 tokens del modelo.

Autor clave transversal: Iz Beltagy y Arman Cohan aparecen tanto en SciBERT como en SPECTER (línea 2), consolidándose como referentes metodológicos del PLN científico.

2. Representación semántica y embeddings de documentos

Documentos: SPECTER ([ZSBTWQEU](#) / [582FUITG](#) , 2020) · SciRepEval ([XYWI4HPW](#) , 2022)

Esta línea se centra en cómo representar documentos científicos completos (no solo palabras) como vectores útiles para tareas posteriores sin reentrenamiento. SPECTER introduce embeddings a nivel de documento informados por señales de citación. SciRepEval identifica una limitación clave de SPECTER: generaliza mal entre distintos formatos de tarea (clasificación, regresión, ranking, búsqueda), y  pone SPECTER2, que aprende múltiples embeddings por documento según el formato.

Autores clave

Autor	Documentos	Rol
Arman Cohan	SciBERT, SPECTER, SciRepEval	Nodo central — conecta las tres primeras líneas temáticas
Iz Beltagy	SciBERT, SPECTER	Co-creador de los modelos base
Susie Xi Rao	Clasificación jerárquica WOS (2023)	Clasificación a escala masiva
Michael Gusenbauer	Fine-tuning SciBERT/ASJC (2026)	Trabajo más reciente y de mayor escala de categorías

Ejemplo 3. Comparación conceptual

¿Cuál es la diferencia entre BERT, SciBERT y Sentence-BERT? Usa mi biblioteca de Zotero si hay documentos relacionados. Explica sus diferencias en objetivo, arquitectura, dominio de entrenamiento y posibles usos en investigación científica.

Resultado

Aspecto	BERT	SciBERT	Sentence-BERT (SBERT)
Objetivo	Modelo de lenguaje de propósito general para entender contexto bidireccional en tareas de PLN (clasificación, NER, QA, inferencia).	Adaptar BERT al dominio científico, donde el vocabulario técnico y la escritura especializada degradan el rendimiento de BERT genérico.	Generar embeddings de oraciones/documentos completos que sean comparables directamente por similitud (coseno), algo que BERT no permite de forma eficiente.
Arquitectura	Transformer <i>encoder</i> bidireccional, preentrenado con <i>Masked Language Modeling + Next Sentence Prediction</i> .	Misma arquitectura que BERT (sin cambios estructurales); la innovación es el corpus y vocabulario de preentrenamiento , no la arquitectura.	Red <i>siamesa/triple</i> que reutiliza un encoder tipo BERT (o SciBERT) en dos ramas gemelas con pesos compartidos, añadiendo <i>pooling</i> y funciones de pérdida (softmax, MSE, <i>triplet loss</i>) para optimizar directamente la similitud semántica entre pares de oraciones.
Dominio de entrenamiento	Texto genérico: Wikipedia + BookCorpus.	Preentrenado sobre 1.14 millones de párrafos científicos de Semantic Scholar.	Se afina sobre datasets de pares/tripletas de oraciones etiquetadas por similitud (p. ej.

Ejemplo 4. Organización por temas

Agrupar mis documentos de Zotero sobre NLP en categorías como modelos de lenguaje, embeddings, clasificación textual, análisis semántico, recuperación de información y aplicaciones científicas.

Resultado

1. Modelos de lenguaje

Título	Autores	Año	Key
SciBERT: A Pretrained Language Model for Scientific Text	Beltagy, Lo, Cohan	2019	SIUCDS2T

Único modelo de lenguaje preentrenado como tal en tu biblioteca; los demás lo usan como componente (*fine-tuning*) más que como objeto de estudio en sí mismo.

2. Embeddings y representación de documentos

Título	Autores	Año	Key
SPECTER: Document-level Representation Learning using Citation-informed Transformers	Cohan, Feldman, Beltagy, et al.	2020	ZSBTWQEU (dup. 582FUITG)
SciRepEval: A Multi-Format Benchmark for Scientific Document Representations	Singh, D'Arcy, Cohan, et al.	2022	XYWI4HPW
Multi-label classification of research articles using Word2Vec...	Mustafa, Usman, Yu, et al.	2021	7QCAYG9V (dup. HHNUCJAI)

Ejemplo 5. Trazabilidad bibliográfica

Para cada documento encontrado, dame item key de Zotero, título, autores, año, DOI, colección o tema asociado, resumen breve y nivel de confianza.

Resultado

Aquí tienes la tabla completa con todos los documentos identificados en tu biblioteca a lo largo de esta conversación:

Item Key	Título	Autores	Año	DOI	Colección/Tema
SIUCDS2T	SciBERT: A Pretrained Language Model for Scientific Text	Beltagy, I.; Lo, K.; Cohan, A.	2019	10.18653/v1/D19-1371	U3T79UYD
ZSBTWQEU (dup. 582FUITG)	SPECTER: Document-level Representation Learning using Citation-informed Transformers	Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; Weld, D.	2020	10.18653/v1/2020.acl-main.207	Sin colección asignada
XYWI4HPW	SciRepEval: A Multi-Format Benchmark for Scientific Document	Singh, A.; D'Arcy, M.; Cohan, A.; Downey, D.; Feldman, S.	2022	10.48550/ARXIV.2211.13308	U3T79UYD

¿Qué es el Item Key?

En todos los ejemplos que hemos realizado con CLAUDE, se ha mostrado en la respuesta el **Item Key**. Este es un **identificador interno o código único** de Zotero que se asigna a cada registro dentro de la biblioteca. Sirve para **identificar con precisión un documento dentro de Zotero**, especialmente cuando hay documentos con títulos parecidos, duplicados o versiones distintas (**ID Zotero = Item Key**).

Por ejemplo:

Título: SciBERT: A Pretrained Language Model for Scientific Text
Item Key: SIUCDS2T

Eso significa:

En la biblioteca Zotero, el registro exacto de ese documento está identificado internamente con el código **SIUCDS2T**.

Por qué es útil

El Item Key es útil para trazabilidad ya que permite saber exactamente qué registro encontró Claude, esto es importante porque en Zotero puede haber:

- Documentos duplicados
- Versiones distintas del mismo artículo
- Registros incompletos
- Registros con DOI
- Registros sin DOI
- PDF asociados
- Notas o anotaciones diferentes

Ejemplo:

```
SPECTER: Document-level Representation Learning using Citation-informed Transformers
Item Key: ZSBTWQEU
duplicado: 582FUITG
```

Claude está diciendo que encontró un registro principal (**ZSBTWQEU**) y además detectó otro registro posiblemente duplicado (**582FUITG**).

Parte IV. Opciones avanzadas

1. Indexar texto completo de PDFs

Primero se recomienda probar la integración con metadatos.

Cuando todo funcione, se puede indexar texto completo ejecutando:

```
zotero-mcp update-db --fulltext --force-rebuild
```

Esto puede tardar más y ocupar más espacio, por lo que se recomienda usarlo si se desea que Claude consulte contenido dentro de PDFs y no solo títulos, resúmenes, tags o metadatos.

Resultado:

```

Using configuration: C:\Users\danny\.config\zotero-mcp\config.json
Starting database update...
Extracting full-text content from local Zotero database...
Scanning local Zotero database for items...
Found 361 candidate items.
Extracting content...

Note: Most papers take 1-3 seconds. Some larger or complex PDFs
may take up to 30 seconds. Password-protected or corrupted files
will be skipped automatically. The system moves on to the next
paper if a file can't be processed in time.

Extraction complete: 361 items to index

Indexing 361 items...

Done: 361 indexed, 0 skipped, 0 errors

Database update completed:
- Total items: 361
- Processed: 361
- Added: 361
- Updated: 0
- Skipped: 0
- Errors: 0
- Duration: 0:00:16.322374

```

Verificar indexación de texto completo en PDF

Ejemplo 1. Búsqueda de frase en un documento PDF

Busca en mi Zotero documentos que mencionen “contextualized word representations” o “document-level representation learning”. No te limites al título. Revisa contenido disponible, resumen, notas, anotaciones o texto completo si está indexado. Para cada resultado indica:

- Título
- ID Zotero / Item Key
- Fragmento o evidencia textual encontrada
- Si la coincidencia parece venir del título, resumen, metadatos o texto completo

Resultado

SciBERT: A Pretrained Language Model for Scientific Text	SIUCDS2T	Texto completo verificado: el artículo describe que BERT/ELMo "return contextualized embeddings for each token" y dedica la sección 3.5 ("Frozen BERT Embeddings") a usar BERT como " <i>pretrained contextualized word embeddings, like ELMo</i> ". Además cita explícitamente en su bibliografía el paper "Deep contextualized word representations" (Peters et al., 2018).	Texto completo (confirmado directamente)
SPECTER: Document-level Representation Learning using Citation-informed Transformers	ZSBTWQEU (dup. 582FUITG , BC4US67T , 2G22X629 , VK6LZLEQ)	La frase aparece literalmente en el título del artículo; el snippet de una de sus versiones en arXiv añade: "Representation learning is a critical ingredient for natural language processing systems. Rec Transformer language	Título (coincidencia exacta) + texto completo

2. Usar API key de Zotero

La API key no es necesaria para el modo local. Se utiliza cuando se desea trabajar con la biblioteca web de Zotero, sincronización remota o modo híbrido.

Variables frecuentes:

```
ZOTERO_API_KEY
ZOTERO_LIBRARY_ID
ZOTERO_LIBRARY_TYPE
```

Ejemplo:

```
{
  "mcpServers": {
    "zotero": {
      "command": "C:\\Users\\USUARIO\\.local\\bin\\uvx.exe",
      "args": [
        "zotero-mcp"
      ],
      "env": {
        "ZOTERO_LOCAL": "true",
        "ZOTERO_LIBRARY_TYPE": "user",
        "ZOTERO_LIBRARY_ID": "TU_LIBRARY_ID",
        "ZOTERO_API_KEY": "TU_API_KEY",
        "NO_PROXY": "localhost,127.0.0.1,::1"
      }
    }
  }
}
```

Nota: No se debe compartir la API key en capturas públicas, repositorios, chats o documentos abiertos.

3. Usar instalador alternativo zotero-mcp-setup

Existe una ruta alternativa basada en un instalador guiado llamado **zotero-mcp-setup**.

Puede ser útil para usuarios que prefieren una instalación más automatizada.

Sin embargo, la ruta principal de esta guía es:

```
instalar uv;
instalar zotero-mcp;
configurar Claude manualmente;
verificar cada paso.
```

La instalación manual tiene la ventaja de que permite entender, verificar y depurar cada componente.

Parte V. Errores comunes y solución

1. Zotero no responde

Probar:

```
curl.exe http://localhost:23119/connector/ping
```

Si no responde:

```
Abrir Zotero  
activar comunicación local  
cerrar y abrir Zotero  
reiniciar el equipo si es necesario
```

2. Cannot connect to Zotero

Significa que `zotero-mcp` no puede comunicarse con Zotero.

Solución:

```
curl.exe http://localhost:23119/connector/ping
```

Si responde, ejecutar:

```
zotero-mcp db-status
```

3. WinError 10061

Normalmente ocurre porque Zotero está cerrado o la API local no está activa.

Solución:

```
Abrir Zotero;  
confirmar comunicación local;  
probar curl;  
volver a ejecutar zotero-mcp.
```

4. Document count: 0

La base semántica está vacía.

Solución:

```
zotero-mcp update-db --limit 100
```

Luego:

```
zotero-mcp db-status
```

5. Claude no muestra Zotero

Buscar rutas de configuración:

```
Get-ChildItem "$env:APPDATA\Claude", "$env:LOCALAPPDATA\Packages" -Recurse -Filter
"claude_desktop_config.json" -ErrorAction SilentlyContinue |
Select-Object FullName, LastWriteTime, Length
```

Si aparecen dos, configurar ambas.

Cerrar Claude:

```
taskkill /IM Claude.exe /F
```

Abrir Claude nuevamente.

6. Error de JSON

Si Claude muestra:

```
Unexpected non-whitespace character after JSON
```

significa que el archivo JSON tiene texto duplicado, una llave mal cerrada o contenido extra.

Validar con:

```
Get-Content "$env:APPDATA\Claude\claude_desktop_config.json" -Raw | ConvertFrom-Json | Out-
Null
```

Parte VI. Lista de verificación

Antes de realizar los ejemplos, verificar:

```
[ ] Zotero está abierto.
[ ] Zotero permite comunicación local.
[ ] curl responde Zotero is running.
[ ] uv está instalado.
[ ] uvx está instalado.
[ ] zotero-mcp está instalado.
[ ] zotero-mcp db-status funciona.
[ ] La base semántica tiene documentos indexados.
[ ] Claude tiene mcpServers configurado.
[ ] Claude fue cerrado y abierto nuevamente.
[ ] Claude muestra el servidor MCP zotero.
[ ] Claude puede buscar documentos de Zotero.
[ ] Se probó una consulta temática.
[ ] Se guardaron capturas de cada paso.
```

Parte VII. Conclusión

Conclusión

La integración de Zotero local con Claude Desktop mediante **zotero-mcp** representa una alternativa práctica y metodológicamente valiosa para convertir una biblioteca bibliográfica en una fuente activa de consulta mediante inteligencia artificial. Su principal aporte no consiste únicamente en automatizar búsquedas, sino en facilitar procesos académicos más amplios, como la exploración inicial de un campo de estudio, el mapeo temático de la literatura, la identificación de conceptos clave, la comparación de enfoques teóricos, la síntesis de documentos, la preparación de estados del arte y la detección de vacíos de investigación.

No obstante, esta integración debe entenderse como un apoyo al trabajo investigativo y no como un sustituto de la revisión crítica del investigador. Las respuestas generadas por Claude a partir de Zotero pueden acelerar la localización, agrupación y análisis preliminar de documentos, pero deben ser verificadas cuidadosamente. Por ello, se recomienda revisar siempre los documentos originales, confirmar los DOI y metadatos, identificar posibles duplicados, solicitar el **ID Zotero / Item Key** para mantener trazabilidad, pedir justificación temática de los resultados, diferenciar entre hallazgos confirmados y resultados probables, y documentar con claridad la metodología de búsqueda utilizada.

En el caso de una colección de Zotero sobre **Procesamiento de Lenguaje Natural (NLP)**, esta integración puede ayudar a identificar qué documentos tratan sobre embeddings, cuáles emplean modelos como BERT, SciBERT o Sentence-BERT, cuáles se enfocan en clasificación textual, cuáles aplican NLP al análisis de literatura científica y qué vacíos aparecen dentro del corpus documental. De esta manera, Zotero deja de funcionar únicamente como un repositorio pasivo de referencias y se convierte en una base de conocimiento consultable, organizada y enriquecida mediante inteligencia artificial.

En síntesis, la integración **Zotero + Claude + zotero-mcp** puede fortalecer la revisión de literatura, la organización conceptual, la comparación de enfoques, la generación de preguntas de investigación, la preparación de taxonomías o categorías analíticas y la construcción preliminar de conocimiento. Sin embargo, la calidad de los resultados dependerá de la calidad de la biblioteca de Zotero, la claridad de las preguntas formuladas, la capacidad crítica del investigador y la trazabilidad de las fuentes recuperadas. Por tanto, su uso debe estar acompañado de validación académica, revisión documental y criterios metodológicos claros.