

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Redes – Jorge Yaz



Proyecto 1

Daniel Alfredo Rayo Roldán, 22933

Repositorios

Cliente y Servidor remoto : <https://github.com/DanielRasho/Network-P1>

Servidor Local: <https://github.com/DanielRasho/GRAPH-CP>

Demostración de funcionamiento

<https://youtu.be/-wRqb2aPWsc>

MCP Local - GraphCP

El MCP local implementado permite la generación de diagramas de árbol, grafo, entidad relacion entre otros. Para ello utiliza el ejecuta el programa graphviz para la generación de imágenes a partir de un archivo .dot generado por el agente AI. Expone 3 herramientas:

Herramienta	Parámetros	Descripción
set_output_location_file	<ul style="list-style-type: none">directory_path	Permite configurar el lugar donde se depositarán los diagramas generados.
generate_dot_file	<ul style="list-style-type: none">dot_contentfilename	Recibe el contenido en formato .dot del agente y lo guarda con el nombre especificado.

generate_png	<ul style="list-style-type: none"> • dot_content • filename • width • height 	Recibe el contenido en formato .dot del agente y genera un imagen png con las dimensiones especificadas.
--------------	------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------

MCP Remoto - WeatherMCP

El servidor remoto permite conectarse a la API <https://wttr.in/> para obtener del clima actual en una ciudad.

Herramienta	Parámetros	Descripción
city_weather	<ul style="list-style-type: none"> • City 	Busca estadísticas del clima de una ciudad por nombre.

Análisis de Wireshark

Paquetes Sincronización

Los paquetes de sincronización empiezan con un paquete enviado por el cliente al endpoint `/mcp/initialize` para solicitar al servidor crear una conexión.

```
accept: application/json, text/event-stream\r\n
content-type: application/json\r\n
Content-Length: 152\r\n
\r\n
[Response in frame: 74]
[Full request URI: http://54.197.170.14:5309/mcp]
File Data: 152 bytes
```

JavaScript Object Notation: application/json

```
Object
  Member: method
    [Path with value: /method:initialize]
    [Member with value: method:initialize]
    String value: initialize
    Key: method
    [Path: /method]
  Member: params
    Object
      Key: params
      [Path: /params]
  Member: jsonrpc
    [Path with value: /jsonrpc:2.0]
    [Member with value: jsonrpc:2.0]
    String value: 2.0
    Key: jsonrpc
    [Path: /jsonrpc]
  Member: id
    [Path with value: /id:0]
    [Member with value: id:0]
    Number value: 0
    Key: id
    [Path: /id]
```

Acto seguido se el servidor responde con un mcp_session que se utilizara en siguientes conexiones para que el servidor pueda identificar al cliente.

08 5b d6 d0 bf e7 8c 61 a3 92 ef c0 08 00 45 00

01 f8 0f 6e 40 00 34 06 94 0c 36 c5 aa 0e c0 a8

00 0a 14 bd a6 64 44 99 c7 c1 b7 11 fb 17 80 18

01 e7 a4 26 00 00 01 01 08 0a e1 bd 5d 29 7e 42

18 14 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f

4b 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c

3a 20 6e 6f 2d 63 61 63 68 65 2c 20 6e 6f 2d 74

72 61 6e 73 66 6f 72 6d 0d 0a 43 6f 6e 6e 65 63

74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65

0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20

74 65 78 74 2f 65 76 65 6e 74 2d 73 74 72 65 61

6d 0d 0a 4d 63 70 2d 53 65 73 73 69 6f 6e 2d 49

64 3a 20 47 58 41 57 46 4c 48 51 55 4a 54 50 48

4e 4a 53 4a 44 58 50 48 47 44 4a 36 4a 0d 0a 44

61 74 65 3a 20 53 75 6e 2c 20 30 37 20 53 65 70

20 32 30 32 35 20 32 30 3a 32 32 3a 34 39 20 47

4d 54 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67

74 68 3a 20 32 33 35 0d 0a 0d 0a 65 76 65 6e 74

3a 20 6d 65 73 73 61 67 65 0a 69 64 3a 20 52 4c

37 5a 37 46 50 49 4a 51 4f 41 4c 35 4f 43 5a 4c

59 57 4c 59 59 46 4c 4d 5f 30 0a 64 61 74 61 3a

20 7b 22 6a 73 6f 6e 72 70 63 22 3a 22 32 2e 30

22 2c 22 69 64 22 3a 30 2c 22 72 65 73 75 6c 74

22 3a 7b 22 63 61 70 61 62 69 6c 69 74 69 65 73

22 3a 7b 22 6c 6f 67 67 69 6e 67 22 3a 7b 7d 2c

22 74 6f 6f 6c 73 22 3a 7b 22 6c 69 73 74 43 68

61 6e 67 65 64 22 3a 74 72 75 65 7d 7d 2c 22 70

72 6f 74 6f 63 6f 6c 56 65 72 73 69 6f 6e 22 3a

22 32 30 32 35 2d 30 36 2d 31 38 22 2c 22 73 65

72 76 65 72 49 6e 66 6f 22 3a 7b 22 6e 61 6d 65

22 3a 22 77 65 61 74 68 65 72 2d 6d 63 70 22 2c

22 76 65 72 73 69 6f 6e 22 3a 22 31 2e 30 2e 30

22 7d 7d 7d 0a 0a

[.....a.....E..

...n@.4. ...6.....

.....dD.

...&.....)]~B

..HTTP/1 .1 200 0

K..Cache -Control

: no-cac he, no-t

ransform ..Connec

tion: ke ep-alive

..Conten t-Type:

text/eve nt-strea

m..Mcp-S ession-I

d: GXAWF LHQUJTPH

NJSJDXPH GDJ6J..D

ate: Sun , 07 Sep

2025 20 :22:49 G

MT-Cont ent-Leng

th: 235. ...event

: messag e.id: RL

7Z7FPIJQ OAL50CZL

YWLYYFLM _0.data:

{"jsonr pc":"2.0

","id":0,"result

":{"capa bilities

":{"logg ing":{,

"tools": {"listCh

anged":t rue}},"p

rotocolV ersion":

"2025-06 -18","se

rverInfo ":{"name

":"weath er-mcp",

"version ": "1.0.0

"}}}}..

Hypertext Transfer Protocol

GET /mcp HTTP/1.1\r\n

Request Method: GET

Request URI: /mcp

Request Version: HTTP/1.1

Host: 54.197.170.14:5309\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

User-Agent: python-httpx/0.28.1\r\n

accept: application/json, text/event-stream\r\n

content-type: application/json\r\n

mcp-session-id: GXAWFLHQUJTPHNJSJDXPHGDJ6J\r\n

mcp-protocol-version: 2025-06-18\r\n

Accept: text/event-stream\r\n

Cache-Control: no-store\r\n

\r\n

[Full request URI: http://54.197.170.14:5309/mcp]

0020 aa 0e a6 80 14 bd 7c 82 1a a7 14 19 04 ed 80 18

0030 01 f6 a3 03 00 00 01 01 08 0a 7e 42 18 a0 e1 bd

0040 5d 6e 47 45 54 20 2f 6d 63 70 20 48 54 54 50 2f

0050 24 2e 24 2d 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e 2e

.....|.....

.....~B.....

]nGET /m cp HTTP/

1.1 Host: 54.197

Paquetes de solicitud

Una vez establecida la conexión, el cliente envía una petición de /list para obtener la lista de herramientas disponibles en el servidor

Wireshark packet capture showing an HTTP POST request to /mcp and its JSON response. The response body contains a JSON object with a 'tools' array. A red arrow points to the 'tools' array in the response body.

```

POST /mcp HTTP/1.1\r\n
Request Method: POST
Request URI: /mcp
Request Version: HTTP/1.1
Host: 54.197.170.14:5309\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
User-Agent: python-httpx/0.28.1\r\n
accept: application/json, text/event-stream\r\n
content-type: application/json\r\n
mcp-session-id: GYAWELH0U1TDHM1S1NYDHD161\r\n

0000 8c 61 a3 92 ef c0 08 5b d6 d0 bf e7 08 00 45 00  .a....[.....E.
0010 00 62 78 fe 40 00 40 06 20 12 c0 a8 00 0a 36 c5  .bx.@. ....6.
0020 aa 0e a6 82 14 bd be a0 18 07 75 81 d9 1e 80 18  . ....u....
0030 01 f6 a1 da 00 00 01 01 08 0a 7e 42 19 21 e1 bd  . ....~B!...
0040 5d f0 7b 22 6d 65 74 68 6f 64 22 3a 22 74 6f 6f  ]d{"meth od":"too
0050 6c 73 2f 6c 69 73 74 22 2c 22 6a 73 6f 6e 72 70  ls/list" ,"jsonrp
0060 63 22 3a 22 32 2e 30 22 2c 22 69 64 22 3a 31 7d  c":"2.0" ,"id":1}

```

El servidor responde con una lista completa de herramientas y descripciones de todos los argumentos.

Wireshark packet capture showing the JSON response body. The response body contains a JSON object with a 'tools' array. A red arrow points to the 'tools' array in the response body.

```

01 e7 79 be 00 00 01 01 08 0a e1 bd 5e 33 7e 42  .y.....^3~B
19 21 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f  .!HTTP/1 .1 200 0
4b 0d 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c  K..Cache -Control
3a 20 6e 6f 2d 63 61 63 68 65 2c 20 6e 6f 2d 74  : no-cac he, no-t
72 61 6e 73 66 6f 72 6d 0d 0a 43 6f 6e 6e 65 63  ransform ..Connec
74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65  tion: ke ep-alive
0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20  ..Conten t-Type:
74 65 78 74 2f 65 76 65 6e 74 2d 73 74 72 65 61  text/eve nt-strea
6d 0d 0a 44 61 74 65 3a 20 53 75 6e 2c 20 30 37  m..Date: Sun, 07
20 53 65 70 20 32 30 32 35 20 32 30 3a 32 32 3a  Sep 202 5 20:22:
34 39 20 47 4d 54 0d 0a 43 6f 6e 74 65 6e 74 2d  49 GMT.. Content-
4c 65 6e 67 74 68 3a 20 33 35 34 0d 0a 0d 0a 65  Length: 354...e
76 65 6e 74 3a 20 6d 65 73 73 61 67 65 0a 69 64  vent: me ssage.id
3a 20 37 54 4e 36 37 34 34 34 4c 35 36 5a 51 43  : 7TN674 44L56ZQC
37 56 4c 41 4e 44 36 4d 36 4f 56 49 5f 30 0a 64  7VLAND6M 60VI_0.d
61 74 61 3a 20 7b 22 6a 73 6f 6e 72 70 63 22 3a  ata: {"j sonrpc":
22 32 2e 30 22 2c 22 69 64 22 3a 31 2c 22 72 65  "2.0", "i d":1,"re
73 75 6c 74 22 3a 7b 22 74 6f 6f 6c 73 22 3a 5b  sult":{" tools":[
7b 22 64 65 73 63 72 69 70 74 69 6f 6e 22 3a 22  {"descri ption":
47 65 74 20 74 68 65 20 77 65 61 74 68 65 72 20  Get the weather
73 74 61 74 69 73 74 69 63 73 20 66 6f 72 20 67  statisti cs for g
69 76 65 6e 20 63 69 74 79 20 6e 61 6d 65 22 2c  iven cit y name",
22 69 6e 70 75 74 53 63 68 65 6d 61 22 3a 7b 22  "inputSc hema":{"
74 79 70 65 22 3a 22 6f 62 6a 65 63 74 22 2c 22  type":"o bject","
72 65 71 75 69 72 65 64 22 3a 5b 22 63 69 74 79  required ":["city
22 5d 2c 22 70 72 6f 70 65 72 74 69 65 73 22 3a  ],"prop erties":
7b 22 63 69 74 79 22 3a 7b 22 74 79 70 65 22 3a  {"city": {"type":
22 73 74 72 69 6e 67 22 2c 22 64 65 73 63 72 69  "string", "descri
70 74 69 6f 6e 22 3a 22 43 69 74 79 20 6e 61 6d  ption": " City nam
65 20 74 6f 20 67 65 74 20 77 65 61 74 68 65 72  e to get weather
20 66 6f 72 22 7d 7d 2c 22 61 64 64 69 74 69 6f  for"}}, "additio
6e 61 6c 50 72 6f 70 65 72 74 69 65 73 22 3a 66  nalPrope rties":f
61 6c 73 65 7d 2c 22 6e 61 6d 65 22 3a 22 63 69  else}, "n ame":"ci
74 79 5f 77 65 61 74 68 65 72 22 7d 5d 7d 7d 0a  ty_weath er"}}}.
0a

```

Por último el cliente ya es capaz de realizar solicitudes al servidor al llamar al endpoint /call y pasar los parámetros.

00	8c 61 a3 92 ef c0 08 5b d6 d0 bf e7 08 00 45 00	·a·····[······E·
10	00 a9 7f f2 40 00 40 06 18 d7 c0 a8 00 0a 36 c5	·····@·@· ······6·
20	aa 0e 81 da 14 bd 71 41 47 5a 56 16 7b 35 80 18	······qA GZV·{5·
30	01 f6 a2 21 00 00 01 01 08 0a 7e 42 4c 54 e1 bd	····!···· ···~BLT··
40	91 22 7b 22 6d 65 74 68 6f 64 22 3a 22 74 6f 6f	·{"meth od":"too
50	6c 73 2f 63 61 6c 6c 22 2c 22 70 61 72 61 6d 73	ls/call", "params
60	22 3a 7b 22 6e 61 6d 65 22 3a 22 63 69 74 79 5f	":{"name ":"city_
70	77 65 61 74 68 65 72 22 2c 22 61 72 67 75 6d 65	weather", "argume
80	6e 74 73 22 3a 7b 22 63 69 74 79 22 3a 22 47 75	nts":{"c ity":"Gu
90	61 74 65 6d 61 6c 61 20 43 69 74 79 22 7d 7d 2c	atemala City"}},
a0	22 6a 73 6f 6e 72 70 63 22 3a 22 32 2e 30 22 2c	"jsonrpc ":"2.0",
b0	22 69 64 22 3a 32 7d	"id":2}

Paquetes de respuesta

El servidor responde con la información del clima en la ciudad, datos que el cliente transmite al agente, este ciclo repitiendo varias veces hasta que el cliente o el servidor cierren las conexiones.

0090	0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20	..Conten t-Type:
00a0	74 65 78 74 2f 65 76 65 6e 74 2d 73 74 72 65 61	text/event-strea
00b0	6d 0d 0a 44 61 74 65 3a 20 53 75 6e 2c 20 30 37	m..Date: Sun, 07
00c0	20 53 65 70 20 32 30 32 35 20 32 30 3a 32 33 3a	Sep 202 5 20:23:
00d0	30 33 20 47 4d 54 0d 0a 43 6f 6e 74 65 6e 74 2d	03 GMT.. Content-
00e0	4c 65 6e 67 74 68 3a 20 34 36 39 0d 0a 0d 0a 65	Length: 469....e
00f0	76 65 6e 74 3a 20 6d 65 73 73 61 67 65 0a 69 64	vent: me ssage.id
0100	3a 20 34 45 35 36 53 49 4a 53 4e 53 59 4d 55 4e	: 4E56SI JSNSYMUN
0110	54 33 4e 4d 45 44 50 45 34 43 53 44 5f 30 0a 64	T3NMEDPE 4CSD_0.d
0120	61 74 61 3a 20 7b 22 6a 73 6f 6e 72 70 63 22 3a	ata: {"j sonrpc":
0130	22 32 2e 30 22 2c 22 69 64 22 3a 32 2c 22 72 65	"2.0","i d":2,"re
0140	73 75 6c 74 22 3a 7b 22 63 6f 6e 74 65 6e 74 22	sult":{" content"
0150	3a 5b 7b 22 74 79 70 65 22 3a 22 74 65 78 74 22	: [{"type ":"text"
0160	2c 22 74 65 78 74 22 3a 22 f0 9f 8c a4 ef b8 8f	, "text": ".....
0170	20 2a 2a 43 75 72 72 65 6e 74 20 57 65 61 74 68	**Curre nt Weath
0180	65 72 20 69 6e 20 47 75 61 74 65 6d 61 6c 61 20	er in Gu atemala
0190	43 69 74 79 2a 2a 5c 6e 5c 6e 2a 2a 4c 6f 63 61	City**\n \n**Loca
01a0	74 69 6f 6e 2a 2a 3a 20 47 75 61 74 65 6d 61 6c	tion**: Guatemal
01b0	61 20 43 69 74 79 2c 20 47 75 61 74 65 6d 61 6c	a City, Guatemal
01c0	61 5c 6e 5c 6e 2a 2a 43 75 72 72 65 6e 74 20 43	a\n\n**C urrent C
01d0	6f 6e 64 69 74 69 6f 6e 73 2a 2a 3a 5c 6e e2 80	ondition s**:\n..
01e0	a2 20 54 65 6d 70 65 72 61 74 75 72 65 3a 20 32	. Temper ature: 2
01f0	31 c2 b0 43 20 28 37 30 c2 b0 46 29 5c 6e e2 80	1..C (70 ..F)\n..
0200	a2 20 46 65 65 6c 73 20 6c 69 6b 65 3a 20 32 31	. Feels like: 21
0210	c2 b0 43 20 28 37 30 c2 b0 46 29 5c 6e e2 80 a2	..C (70 ..F)\n...
0220	20 57 65 61 74 68 65 72 3a 20 50 61 72 74 6c 79	Weather : Partly
0230	20 63 6c 6f 75 64 79 5c 6e e2 80 a2 20 48 75 6d	cloudy\ n... Hum
0240	69 64 69 74 79 3a 20 37 38 25 5c 6e e2 80 a2 20	idity: 7 8%\n...
0250	57 69 6e 64 3a 20 34 20 6b 6d 2f 68 20 4e 4e 45	Wind: 4 km/h NNE
0260	5c 6e e2 80 a2 20 50 72 65 73 73 75 72 65 3a 20	\n... Pr essure:
0270	31 30 32 33 20 6d 62 5c 6e e2 80 a2 20 55 56 20	1023 mb\ n... UV
0280	49 64 65 78 3a 20 36 5c 6e e2 80 a2 20 56 69	Index: 6 \n... Vi
0290	73 69 62 69 6c 69 74 79 3a 20 31 30 20 6b 6d 5c	sibility : 10 km\
02a0	6e 5c 6e 2a 2a 4c 61 73 74 20 55 70 64 61 74 65	n\n**Las t Update
02b0	64 2a 2a 3a 20 30 33 3a 31 38 20 50 4d 22 7d 5d	d**: 03: 18 PM"}]
02c0	7d 7d 0a 0a	}}..

El flujo de información en las diferentes capas

El protocolo MCP está construido encima sobre todas las capas aprendidas en clase, llegando a la capa de aplicación al utilizar HTTP. Cada vez que se envía un mensaje en protocolo MCP este es codificado en notación JSON, este a su vez es empaquetado con metadata y headers como `mcp_session` discutido antes usando HTTP, este protocolo a su vez depende de la capa de transporte usan TCP que divide el mensaje original en fragmentos y los transmite uno a uno usando banderas de *Acknowledge* y *Sync*.

El responsable de definir a donde se enviaran esos paquetes es la capa de enlace al viajar por routers y ser enviados al servidor destino que cuenta con su propia MAC. Estos paquetes realmente son señales eléctricas o electromagnéticas que fluyen a través del cobre y fibra óptica instalados por el ISP. Y es así como un agente de IA puede saber el clima de la ciudad.

Conclusiones y comentarios

El protocolo MCP vino a ser la interfaz que el mundo tecnológico necesitaba para poder estandarizar los métodos por los cuales un agente de inteligencia artificial interactúa con su entorno, a fecha de este reporte el protocolo siguen con en crecimiento y cambio en una afán por poder soportar la conexión a servidores remotos, al igual que los SDK teniendo comunidades activas que los mantienen.

El proyecto fue interesante de hacer para explorar implementaciones fuera de las comunes API REST y permitió terminar con productos útiles que se pueden sumar al open source.

Algunos puntos de mejora para futuras implementaciones es el uso de https y OAuth para autenticación de llamadas, así como la implementación de una UI en navegador.

Sobre el uso de IA

Este proyecto se ayudó de la inteligencia artificial Claude para la implementación de funcionalidades varias dentro del código del cliente CLI (conexión usando capo streamable y creación de interfaz de CLI) & el servidor local GraphCP, posteriormente el responsable lo utilizó como base para hacer ajustes de acuerdo a la rúbrica requerida.

Referencias

Architecture overview. (n.d.). Model Context Protocol. Retrieved September 7, 2025, from <https://modelcontextprotocol.io/docs/learn/architecture>

How to MCP - the complete guide to understanding model context protocol and building remote servers. (n.d.). Simplescraper.io. Retrieved September 7, 2025, from <https://simplescraper.io/blog/how-to-mcp>

jba. (2025). Examples/http at main · modelcontextprotocol/go-sdk.
<https://github.com/modelcontextprotocol/go-sdk/tree/main/examples/http>

Anthropic. (2024). Claude (versión del 10 de agosto) [Modelo de lenguaje de gran tamaño]. <https://chat.openai.com/>