Proyecto 1 Redes

José Santiago Pereira Alvarado - 22318

Información General:

- **Archivo principal**: `csv_mcp_server.py`
- **Funcionalidad**: Análisis de datos CSV con capacidades de limpieza, visualización entre otros.
- **Protocolo**: MCP sobre JSON-RPC 2.0
- **Repositorio**: https://github.com/SRV-JSPA/mcp_server.git

Herramientas Implementadas:

Herramienta	**Descripción**	**Parámetros Principales**	**Funcionalidad**
`analyze_csv`	Análisis estadístico completo	`file_path`	Estadísticas descriptivas, análisis numérico y categórico
`detect_outliers_in_csv`	Detección de valores atípicos	`file_path`, `columns`, `method`	IQR
`calculate_correlations_csv`	Análisis de correlaciones	`file_path`, `method`	Pearson
`clean_csv_data`	Limpieza de datos	`file_path`, `operations`, `save_cleaned`	Duplicados, valores faltantes
`filter_csv_data`	Filtrado de datos	`file_path`, `filters`, `save_filtered`	Condiciones múltiples
`group_csv_data`	Agrupación y agregación	`file_path`, `group_by`, `aggregations`	Mean, sum, count, min, max

`create_csv_visualization`	Visualizaciones	`file_path`, `plot_type`, `columns`	Histogramas, boxplots, scatter, heatmaps
`debug_workspace`	Diagnóstico del workspace	Ninguno	Información del sistema

Características Avanzadas:

- Gestión inteligente del workspace con `WorkspaceManager`
- Soporte múltiple de codificaciones
- Detección de outliers con múltiples algoritmos
- Visualizaciones automáticas con matplotlib/seaborn
- Análisis estadístico robusto con scipy/sklearn

Servidor MCP Remoto - Remote System Monitor Server

Información General:

- **URL de despliegue**: `https://mcp-redes-jmp.uc.r.appspot.com`
- **Plataforma**: Google Cloud Run
- **Archivo principal**: `remote_mcp_server.py`
- **Funcionalidad**: Monitoreo de sistema remoto en tiempo real
- **Framework**: Flask + Python + psutil

Herramientas de Monitoreo:

Herramienta	**Descripción**	**Parámetros**	**Información Proporcionada**
`get_system_overview`	Resumen completo del sistema	Ninguno	CPU, memoria, disco, red, alertas
`monitor_cpu_usage`	Monitoreo detallado de CPU	Ninguno	Uso por núcleo, frecuencia, load average
`monitor_memory_usage`	Análisis de memoria	Ninguno	RAM física, swap, recomendaciones
`monitor_disk_usage`	Estado de almacenamiento	Ninguno	Particiones, I/O, alertas de espacio

`monitor_network_stats`	Estadísticas de red	Ninguno	Interfaces, ancho de banda, conexiones
`get_top_processes`	Procesos principales	`limit` (1-50, default: 10)	CPU y memoria por proceso
`system_monitor_info`	Información del servidor	Ninguno	Capacidades y especificaciones

Endpoints HTTP:

Endpoint	Método	Función	Respuesta
`/`	GET	Información del servicio	Metadatos del servidor
`/health`	GET	Verificación de salud	Estado del sistema
`/mcp/tools/list`	GET	Lista de herramientas	Especificación JSON-RPC
`/mcp/tools/call`	POST	Ejecución de herramientas	Resultado JSON-RPC

Servidores Oficiales Integrados

Filesystem MCP Server (Anthropic):

- **Comando**: `npx -y @modelcontextprotocol/server-filesystem`
- **Funcionalidad**: Operaciones de sistema de archivos
- **Herramientas**: read_file, write_file, list_directory, search_files

Git MCP Server (Anthropic):

- **Comando**: `uvx mcp-server-git`
- **Funcionalidad**: Control de versiones Git
- **Herramientas**: git_status, git_add, git_commit, git_log, git_create_branch

Análisis de Comunicación por Capas de Red

Metodología de Análisis

El análisis se realizó capturando tráfico hacia el servidor remoto desplegado en Google Cloud Run con SNI `mcp-redes-jmp.uc.r.appspot.com` .

Capa de Aplicación

Protocolo: HTTP/1.1 sobre TLS 1.3 con JSON-RPC 2.0

Tipos de Mensajes Identificados:

1. **Mensajes de Sincronización:**
    - `GET /health` - Verificación de conectividad del servidor
    - `GET /mcp/tools/list` - Obtención de herramientas disponibles
    - **Propósito**: Establecer la conexión y descubrir capacidades
2. **Mensajes de Solicitud/Petición:**
    - `POST /mcp/tools/call` - Ejecución de herramientas MCP
3. **Mensajes de Respuesta:**
    - HTTP 200 OK con payload JSON-RPC 2.0

Headers HTTP Observados:

- `Content-Type: application/json`
- `User-Agent: aiohttp/3.x`
- `Host: mcp-redes-jmp.uc.r.appspot.com`
- `Accept-Encoding: gzip, deflate`

Capa de Transporte

Protocolo: TCP

Análisis de Conexiones:

- **Puerto de destino**: 443 (HTTPS)
- **Puerto de origen**: Dinámico
- **Establecimiento**: Handshake TCP de 3 vías
- **Terminación**: Cierre ordenado con FIN/ACK

Capa de Red

Protocolo: IPv4

Direccionamiento IP:

- **IP de origen**: IP pública del cliente
- **IP de destino**: Rango de Google Cloud

10.5 Capa de Enlace

Protocolo Local: Ethernet

Direccionamiento MAC:

- **MAC de origen**: B4-45-06-D0-7B-3C
- **MAC de destino**: 4e-91-f7-cf-ba-f8

Conclusiones y Comentarios sobre el Proyecto

Dificultades Encontradas

Integración de Servidores MCP:

- La configuración inicial de múltiples servidores MCP requirió manejo cuidadoso de procesos asincrónicos y gestión de errores
- La diferencia entre servidores locales y remotos en su implementación.

Desarrollo del Servidor Local CSV:

- Implementar análisis que maneje diversos formatos y codificaciones de CSV
- Gestión del workspace y persistencia de visualizaciones requirió consideración cuidadosa de rutas y permisos

Despliegue en Google Cloud:

- Configuración de CORS y manejo de requests HTTP en el servidor Flask
- Asegurar que psutil funcione correctamente en el entorno containerizado de Cloud Run

Análisis de Tráfico de Red:

- El tráfico HTTPS encripta el contenido JSON-RPC, limitando el análisis a metadatos de conexión

Lecciones Aprendidas

Sobre el Protocolo MCP:

- MCP proporciona una abstracción efectiva para integrar herramientas heterogéneas en un chatbot inteligente
- La estandarización JSON-RPC 2.0 facilita la interoperabilidad entre diferentes implementaciones
- La separación entre cliente, servidor y anfitrión permite arquitecturas modulares y escalables

Desarrollo con LLMs:

- La selección inteligente de herramientas mejora significativamente la experiencia del usuario
- La capacidad de Claude para analizar contexto y seleccionar herramientas apropiadas

Opinión Final

En general me pareció un proyecto muy interesante el cual nos hace siempre estar actualizados en nuestra profesión con las cosas nuevas que van surgiendo por la innovación, además me pareció muy útil para poder implementar y usar agentes de IA con el uso de protocolos de comunicación.