

UNIVERSIDAD DE GUAYAQUIL
FACULTAD
INGENIERÍA INDUSTRIAL
MATERIA
LENGUAJE DE PROGRAMACIÓN

DOCENTE

ING. Cesar Alcívar

INTEGRANTES

- *BRAYTHON JOSUE SORIANO PAREDES*
- *MURILLO SUAREZ IVAN JESUS*
- *DANIEL ETHAN JAYA JARAMILLO*

Entregable 7 - Documentación Final

1. INTRODUCCIÓN

Este documento corresponde al **Entregable 7 – Documentación Final**, cuyo objetivo es integrar y presentar de forma completa el proyecto desarrollado durante el curso. Aquí se describen el propósito del sistema, su arquitectura, tecnologías utilizadas, configuración, resultados y conclusiones.

2. DESCRIPCIÓN DEL PROYECTO

El proyecto consiste en un sistema de consulta y análisis de datos utilizando una base de datos SQLite, integrado con herramientas modernas de IA mediante **MCP (Model Context Protocol)**. El sistema permite realizar consultas estructuradas, obtener resultados claros y generar reportes informativos.

3. OBJETIVOS

3.1 Objetivo General

Desarrollar e integrar un sistema funcional que permita consultar, analizar y presentar información almacenada en una base de datos de manera eficiente.

3.2 OBJETIVOS ESPECÍFICOS

- Diseñar y configurar una base de datos SQLite.
- Conectar el sistema a través de MCP.
- Ejecutar consultas SQL para obtener información relevante.
- Presentar resultados claros y documentados.

4. TECNOLOGÍAS UTILIZADAS

- **Python 3.14**
- **PostgreSQL 17**
- **Base de datos Pagila**
- **Model Context Protocol (MCP)**
- **Claude Desktop**
- **MCP Inspector**
- **psycopg2 / psycopg (PostgreSQL driver)**

5. ARQUITECTURA DEL SISTEMA

El sistema está compuesto por:

- Un servidor MCP configurado con distintos módulos (filesystem, memory y sqlite).
- Una base de datos SQLite que almacena la información principal.
- Un cliente de IA (Claude) que realiza consultas y presenta resultados.

6. CONFIGURACIÓN DEL MCP

Se configuraron distintos servidores MCP mediante un archivo JSON, permitiendo:

- Acceso al sistema de archivos.
- Persistencia de memoria.
- Conexión directa a la base de datos SQLite.

Esta configuración permitió realizar consultas SQL de forma correcta y obtener resultados estructurados.

7. CONSULTAS REALIZADAS Y RESULTADOS

Se ejecutaron consultas SQL para obtener información relevante de la base de datos. Como ejemplo, se obtuvieron los **10 registros con mayor número de alquileres**, mostrando correctamente los títulos y el total de alquileres asociados.

Los resultados validan el correcto funcionamiento de la conexión entre MCP y la base de datos.

8. RESULTADOS OBTENIDOS

- Conexión exitosa entre Claude y MCP.
- Ejecución correcta de consultas SQL.
- Obtención de resultados claros y ordenados.
- Integración completa de todos los componentes del proyecto.

9. CONCLUSIONES

El proyecto cumple con los objetivos planteados inicialmente. La integración de MCP con una base de datos SQLite demuestra ser una solución eficaz para la consulta y análisis de información, facilitando el uso de herramientas de IA en proyectos de gestión de datos.

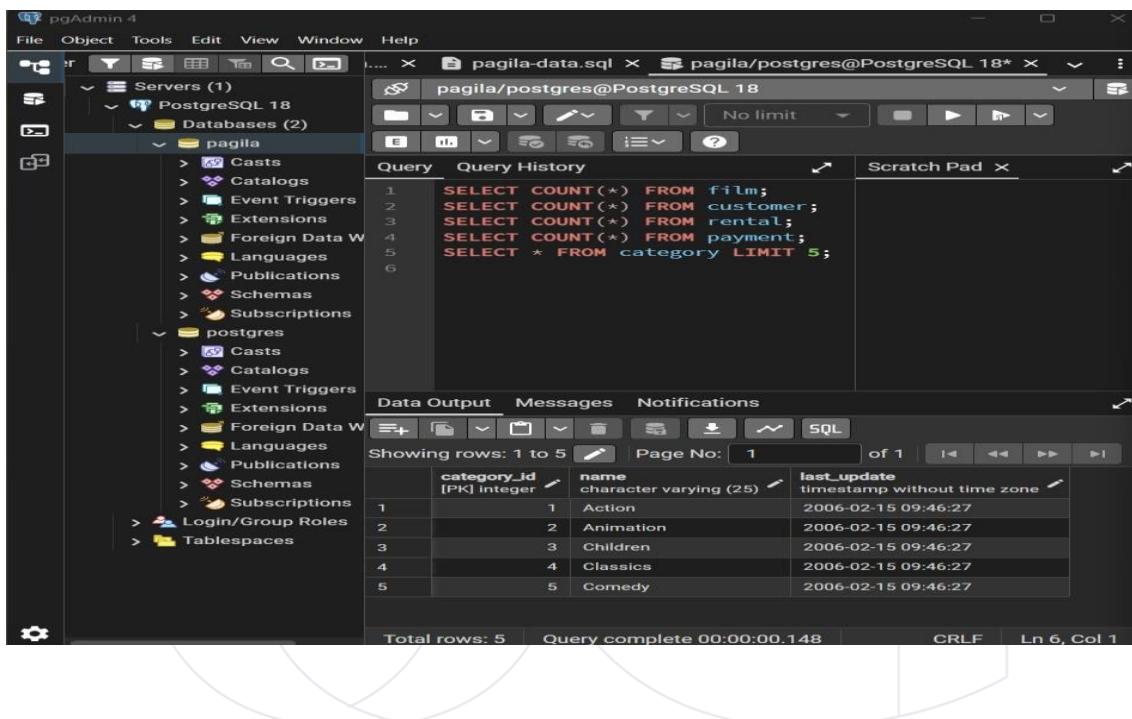
10. REPOSITORIO GITHUB

El código fuente, configuración y documentación adicional del proyecto se encuentran disponibles en el siguiente repositorio:

☞ Link al repositorio GitHub: <https://github.com/brayton992/Servidor-MCP-pagila.git>

11. ANEXOS

PREPARACIÓN DE LA BASE DE DATOS



The screenshot shows the pgAdmin 4 interface. On the left, the tree view displays the 'Servers (1)' section with 'PostgreSQL 18' selected, and within it, the 'Databases (2)' section with 'pagila' selected. The 'Query' tab in the center contains the following SQL code:

```

1 SELECT COUNT(*) FROM film;
2 SELECT COUNT(*) FROM customer;
3 SELECT COUNT(*) FROM rental;
4 SELECT COUNT(*) FROM payment;
5 SELECT * FROM category LIMIT 5;
6

```

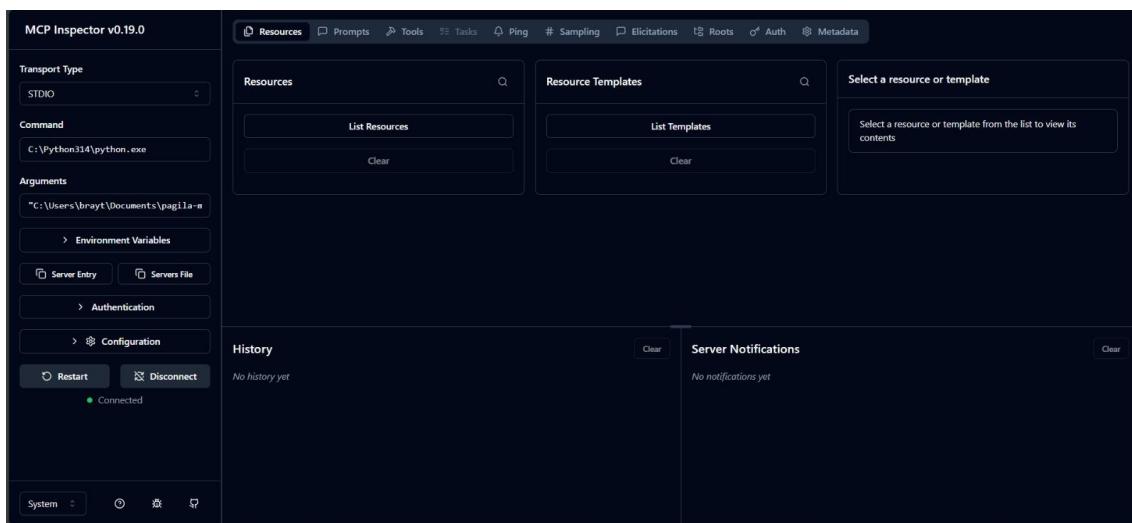
The 'Data Output' tab shows the results of the last query:

category_id	name	last_update
1	Action	2006-02-15 09:46:27
2	Animation	2006-02-15 09:46:27
3	Children	2006-02-15 09:46:27
4	Classics	2006-02-15 09:46:27
5	Comedy	2006-02-15 09:46:27

Total rows: 5 Query complete 00:00:00.148 CRLF Ln 6, Col 1

<https://www.pgadmin.org/>

IMPLEMENTACIÓN DEL SERVIDOR MCP



The screenshot shows the MCP Inspector v0.19.0 application. On the left, the 'MCP Inspector v0.19.0' sidebar includes sections for 'Transport Type' (STDIO), 'Command' (C:\Python314\python.exe), 'Arguments' ("-c:\\Users\\brayt\\Documents\\pagila.mcp"), and buttons for 'Server Entry' and 'Configuration'. Below these are 'Authentication' and 'Configuration' sections with 'Restart' and 'Disconnect' buttons. The main area has tabs for 'Resources', 'Prompts', 'Tools', 'Tasks', 'Ping', 'Sampling', 'Elicitations', 'Roots', 'Auth', and 'Metadata'. The 'Resources' tab is active, showing 'Resources' and 'Resource Templates' sections with 'List Resources' and 'List Templates' buttons. The 'History' section shows 'No history yet'. The 'Server Notifications' section shows 'No notifications yet'.

SEGURIDAD Y CONTROL

MCP Inspector v0.19.0

Transport Type: STDIO

Command: C:\Python314\python.exe

Arguments: "C:\Users\brayt\Documents\pagila-m"

Environment Variables, Server Entry, Servers File, Authentication, Configuration, Restart, Disconnect, Connected.

System, Help, Settings, Exit.

History

```
2. tools/call
Request:
{
    method: "tools/call"
    params: {
        name: "sql_query"
        arguments: {
            query: "SELECT * FROM film"
        }
    }
}
```

Server Notifications

```
2. notifications/message
Details:
{
    method: "notifications/message"
    params: {
        level: "info"
        logger: "stdio"
        data: {
    }
}
```

title: "sql_queryOutput"

Run as task, Run Tool, Copy Input, Tool Result: Success, Structured Content:

```
{
    result: {
        ok: true
        row_count: 200
        rows: [
            ...
        ]
    }
}
```

MCP Inspector v0.19.0

Transport Type: STDIO

Command: C:\Python314\python.exe

Arguments: "C:\Users\brayt\Documents\pagila-m"

Environment Variables, Server Entry, Servers File, Authentication, Configuration, Restart, Disconnect, Connected.

System, Help, Settings, Exit.

Tools

- check_db_status: Verifica conexión y devuelve información básica.
- list_tables: Lista tablas del esquema indicado.
- list_columns: Lista columnas de una tabla.

sql_query

Ejecuta un SELECT controlado (solo lectura).

query*: SELECT * FROM film

limit: 200

Tool-specific Metadata: Add Pair

Output Schema: Expand

History

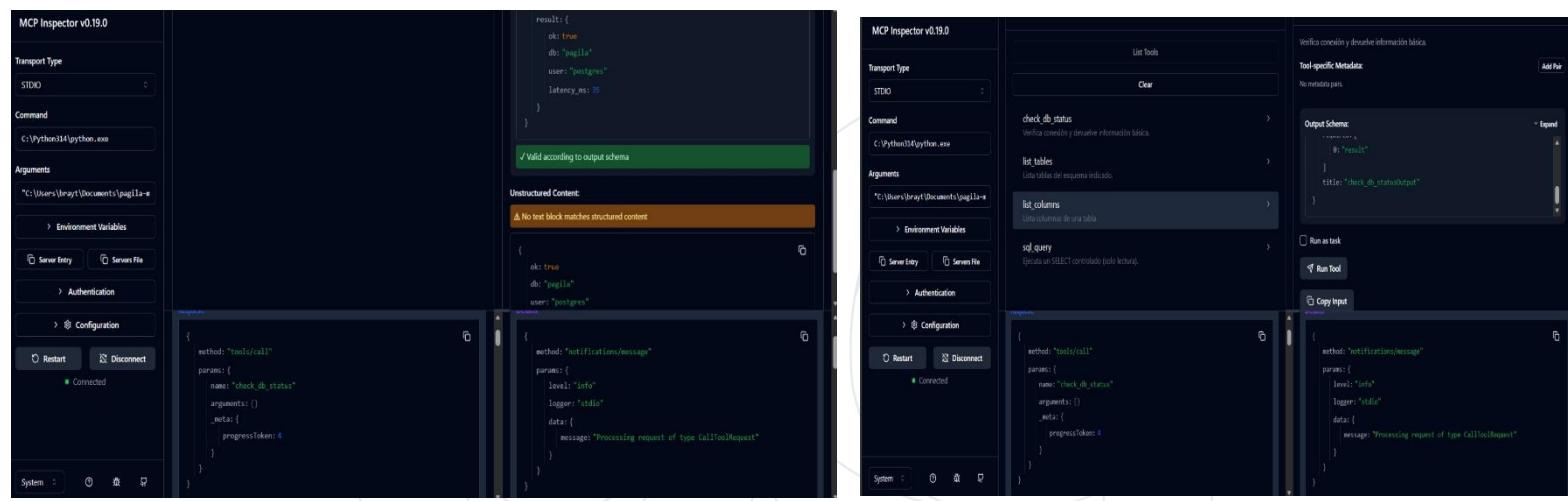
```
2. tools/call
Request:
{
    method: "tools/call"
    params: {
        name: "sql_query"
        arguments: {
            query: "SELECT * FROM film"
        }
    }
}
```

Server Notifications

```
2. notifications/message
Details:
{
    method: "notifications/message"
    params: {
        level: "info"
        logger: "stdio"
        data: {
    }
}
```

```
{
    method: "notifications/message"

    params: {
        level: "info"
        logger: "stdio"
        data: {
            message: "Processing request of type CallToolRequest"
        }
    }
}
```



The MCP Inspector v0.19.0 interface is shown with several tool windows open:

- Transport Type:** Set to STDIO.
- Command:** C:\Python314\python.exe
- Arguments:** "-c: \Users\leayt\Documents\pgcli+e" (with Environment Variables expanded).
- Tool-specific Metadata:** Shows a schema definition for a tool named "check_db_status".
- check_db_status:** A tool window showing command-line arguments and unstructured content.
- list_tables:** A tool window showing a table structure.
- list_columns:** A tool window showing column details.
- sql_query:** A tool window for running SQL queries.
- Run as task:** An option to run the tool as a task.
- Copy Input:** An option to copy input from the tool window.

CONSULTAS EN LENGUAJE NATURAL

Voy a consultar las categorías de películas que han generado mayores ingresos.

Sql query >

Las categorías con mayores ingresos son:

1. **Sports** - \$5,314.21 (1,179 alquileres)
2. **Sci-Fi** - \$4,756.98 (1,101 alquileres)
3. **Animation** - \$4,656.30 (1,166 alquileres)
4. **Drama** - \$4,587.39 (1,060 alquileres)
5. **Comedy** - \$4,383.58 (941 alquileres)
6. **Action** - \$4,375.85 (1,112 alquileres)
7. **New** - \$4,351.62 (940 alquileres)
8. **Games** - \$4,281.33 (969 alquileres)
9. **Foreign** - \$4,270.67 (1,033 alquileres)
10. **Family** - \$4,226.07 (1,096 alquileres)

La categoría **Sports** lidera claramente con más de \$5,300 en ingresos totales, seguida por Sci-Fi y Animation. Es interesante notar que aunque Animation tiene más alquileres que Sci-Fi (1,166 vs 1,101), Sci-Fi genera más ingresos, lo que sugiere que sus películas podrían tener precios de alquiler más altos.

Responder...

+ ⏪ Sonnet 4.5 ▾ ↑

Voy a consultar los clientes más activos según la cantidad de alquileres realizados.

Sql query >

Los clientes más activos son:

1. **ELEANOR HUNT** - 46 alquileres, \$216.54 gastados
2. **KARL SEAL** - 45 alquileres, \$221.55 gastados
3. **CLARA SHAW** - 42 alquileres, \$195.58 gastados
4. **MARCIA DEAN** - 42 alquileres, \$175.58 gastados
5. **TAMMY SANDERS** - 41 alquileres, \$155.59 gastados
6. **WESLEY BULL** - 40 alquileres, \$177.60 gastados
7. **SUE PETERS** - 40 alquileres, \$154.60 gastados
8. **MARION SNYDER** - 39 alquileres, \$194.61 gastados
9. **RHONDA KENNEDY** - 39 alquileres, \$194.61 gastados
10. **TIM CARY** - 39 alquileres, \$175.61 gastados

Eleanor Hunt encabeza la lista como la cliente más activa con 46 alquileres, aunque Karl Seal es quien más ha gastado entre los clientes más activos con \$221.55. Es notable que hay cierta variación en el gasto total entre clientes con similar número de alquileres, lo que sugiere que algunos rentan películas más caras que otros.

Voy a consultar los actores que han participado en más películas.

Sql query >

Los actores con más películas son:

1. **GINA DEGENERES** - 42 películas
2. **WALTER TORN** - 41 películas
3. **MARY KEITEL** - 40 películas
4. **MATTHEW CARREY** - 39 películas
5. **SANDRA KILMER** - 37 películas
6. **SCARLETT DAMON** - 36 películas
7. **VIVIEN BASINGER** - 35 películas
8. **UMA WOOD** - 35 películas
9. **VAL BOLGER** - 35 películas
10. **GROUCHO DUNST** - 35 películas
11. **ANGELA WITHERSPOON** - 35 películas
12. **HENRY BERRY** - 35 películas
13. **WARREN NOLTE** - 34 películas
14. **ANGELA HUDSON** - 34 películas
15. **SIDNEY CROWE** - 34 películas

Películas más alquiladas

Voy a consultar la base de datos para encontrar las películas más alquiladas.

Usé la Integración de pagila

Las películas más alquiladas son:

1. BUCKET BROTHERHOOD - 34 alquileres
2. ROCKETEER MOTHER - 33 alquileres
3. FORWARD TEMPLE - 32 alquileres
4. JUGGLER HARDLY - 32 alquileres
5. RIDGEMONT SUBMARINE - 32 alquileres
6. GRIT CLOCKWORK - 32 alquileres
7. SCALAWAG DUCK - 32 alquileres
8. ROBBERS JOON - 31 alquileres
9. NETWORK PEAK - 31 alquileres
10. TIMBERLAND SKY - 31 alquileres

"Bucket Brotherhood" lidera el ranking con 34 alquileres, seguida de cerca por varias películas que tienen entre 31 y 33 alquileres cada una.

Responder...

Sonnet 4.5

Claude es IA y puede cometer errores. Por favor, verifica nuevamente las respuestas.

Películas más alquiladas

Los ingresos por mes son:

2006:

- Noviembre: \$147.64 (36 pagos)
- Diciembre: \$2,425.24 (576 pagos)

2007:

- Enero: \$7,199.93 (1,707 pagos)
- Febrero: \$12,866.83 (3,117 pagos)
- Marzo: **\$17,546.10 (4,190 pagos)** ★ Mes con mayores ingresos
- Abril: \$14,890.30 (3,470 pagos)
- Mayo: \$9,311.06 (2,194 pagos)
- Junio: \$2,572.05 (598 pagos)
- Julio: \$165.42 (56 pagos)
- Agosto: \$141.50 (50 pagos)
- Septiembre: \$139.50 (48 pagos)
- Octubre: \$0.99 (2 pagos)

El mes con mayores ingresos fue marzo de 2007 con \$17,546.10 y 4,190 pagos. Se observa un pico de actividad entre febrero y julio de 2007, con una caída significativa a partir de junio. Los últimos meses del registro (julio-octubre 2007) muestran muy poca actividad.

EJECUCION DEL PROYECTO

Películas más alquiladas

Genera un informe gráfico que compare los ingresos generados por cada género de películas en la base de datos Pagila.

Voy a generar un informe gráfico comparando los ingresos por género de películas. Primero obtendré los datos y luego crearé visualizaciones interactivas.

Sql query >

Ahora voy a crear un informe interactivo con gráficos comparativos de los ingresos por género.

Creó un archivo, ejecutó un comando, archivo leído >

He creado un informe gráfico interactivo completo que compara los ingresos por género de películas. El informe incluye:

Visualizaciones incluidas:

1. Tarjetas resumen con métricas clave
2. Gráfico de barras - Ingresos totales por género
3. Gráfico circular - Distribución porcentual de ingresos
4. Gráfico de doble eje - Comparación de alquileres vs ingresos
5. Gráfico de líneas - Ingreso promedio por género
6. Tabla detallada - Datos completos ordenados por ranking

Responder...

Sonnet 4.5

Claude es IA y puede cometer errores. Por favor, verifica nuevamente las respuestas.

Informe de Ingresos por Género

Análisis Completo de la Base de Datos Pagila

INGRESOS TOTALES
\$67,406.65
16 categorías de películas

TOTAL DE ALQUILERES
16,044
Todas las categorías

GÉNERO LÍDER
Sports
\$5,314.21 en ingresos

INGRESO PROMEDIO
\$4.20
Por alquiler

Ingresos Totales por Género

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Dashboard Properties SQL pagila-schema.sql pagila-insert-data... pagila-data.sql pagila/postgres@PostgreSQL 18*

Servers (1) PostgreSQL 18 Databases (2) pagila Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas Subscriptions postgres Login/Group Roles Tablespaces

Query Query History

```
1 SELECT COUNT(*) FROM film;
2 SELECT COUNT(*) FROM customer;
3 SELECT COUNT(*) FROM rental;
4 SELECT COUNT(*) FROM payment;
5 SELECT * FROM category LIMIT 5;
```

Data Output Messages Notifications SQL

category_id	name	last_update
1	Action	2006-02-15 09:46:27
2	Animation	2006-02-15 09:46:27
3	Children	2006-02-15 09:46:27
4	Classics	2006-02-15 09:46:27
5	Comedy	2006-02-15 09:46:27

Showing rows: 1 to 5 Page No: 1 of 1 << >>

MCP Inspector v0.19.0

Transport Type: STDIO Command: C:\Python314\python.exe Arguments: "C:/Users/brahyt/Documents/pagila-mcp-server.py" Environment Variables Authentication Configuration Restart Disconnect Connected

Resources Prompts Tools Tasks Ping Sampling Elicitations Roots Auth Metadata

Tools

- check_db_status
- list_tables
- list_columns

sql_query

Ejecuta un SELECT controlado (solo lectura).

query *

```
ORDER BY ingresos DESC
```

limit

10

Tool-specific Metadata: Add Pair

No metadata pairs.

Output Schema: Expand

Server Notifications

4. notifications/message

Details:

```
{
    method: "notifications/message",
    params: {
        level: "info",
        logger: "stdio"
    }
}
```

File Edit Selection View Go Run Terminal Help

PAGILA-MCP-SERVER

main.py requirements.txt

```
main.py
import os
import re
import time
from typing import Any, Dict
from psycopg2 import extras
from psycopg2.extras import RealDictCursor
from dotenv import load_dotenv
from mcp.server.fastmcp import FastMCP
load_dotenv()
DB_HOST = os.getenv("DB_HOST", "127.0.0.1")
DB_PORT = int(os.getenv("DB_PORT", "5432"))
DB_NAME = os.getenv("DB_NAME", "pagila")
DB_USER = os.getenv("DB_USER", "postgres")
DB_PASSWORD = os.getenv("DB_PASSWORD", "")
STATEMENT_TIMEOUT_MS = int(os.getenv("DB_STATEMENT_TIMEOUT_MS", "5000"))
mcp = FastMCP("Pagila MCP Server", json_response=True)
FORBIDDEN = re.compile(
    r"^(?i)(insert|update|delete|drop|alter|create|truncate|grant|revoke|copy|call|do)\b",
    re.IGNORECASE,
)
def validate_sql(query: str) -> str:
    if not query or not query.strip():
        raise ValueError("La consulta está vacía.")
    q = query.strip()
    # bloques múltiples sentencias
    if ";" in q:
        raise ValueError("Solo se permite 1 sentencia. Quita el ';'.")
```

Line 16, Col 41 Spaces: 4 UTF-8 CRLF Python 3.11.9 (Microsoft Store) Go Live Background