**Tarea 5: Script en Bash para validar el estado de recursos en las máquinas Linux**

Para validar el estado de los recursos de una maquina en Linux, se desarrolló el siguiente código de bash:

#!/bin/bash

# Obtener la fecha y hora actual

currentDate=$(date +"%Y-%m-%d %H:%M:%S")

# Obtener el uso de CPU

cpuUsage=$(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/" | awk '{print 100 - $1}')

# Obtener el uso de RAM

totalMemory=$(free -m | awk '/Mem:/ { print $2 }')

usedMemory=$(free -m | awk '/Mem:/ { print $3 }')

usedMemoryGB=$(awk "BEGIN {printf \"%.2f\",${usedMemory}/1024}")

totalMemoryGB=$(awk "BEGIN {printf \"%.2f\",${totalMemory}/1024}")

# Obtener el espacio en disco

diskInfo=$(df -h --output=source,size,used,avail,pcent | grep '^/dev/')

# Generar contenido HTML

htmlContent=$(cat <<EOF

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Reporte del Estado de Recursos del Sistema</title>

    <style>

        body { font-family: Arial, sans-serif; }

        h1 { color: #333; }

        table { width: 100%; border-collapse: collapse; margin-top: 20px; }

        th, td { padding: 10px; border: 1px solid #ccc; text-align: left; }

        th { background-color: #f4f4f4; }

    </style>

</head>

<body>

    <h1>Reporte del Estado de Recursos del Sistema</h1>

    <p>Fecha y Hora: $currentDate</p>

    <h2>Uso de CPU</h2>

    <p>Uso de CPU: $cpuUsage%</p>

    <h2>Uso de RAM</h2>

    <p>Memoria Usada: ${usedMemoryGB} GB</p>

    <p>Memoria Total: ${totalMemoryGB} GB</p>

    <h2>Espacio en Disco</h2>

    <table>

        <tr>

            <th>Dispositivo</th>

            <th>Tamaño</th>

            <th>Usado</th>

            <th>Disponible</th>

            <th>Uso (%)</th>

        </tr>

EOF

)

while IFS= read -r line; do

    device=$(echo $line | awk '{print $1}')

    size=$(echo $line | awk '{print $2}')

    used=$(echo $line | awk '{print $3}')

    available=$(echo $line | awk '{print $4}')

    percent=$(echo $line | awk '{print $5}')

    htmlContent+=$(cat <<EOF

        <tr>

            <td>$device</td>

            <td>$size</td>

            <td>$used</td>

            <td>$available</td>

            <td>$percent</td>

        </tr>

EOF

)

done <<< "$diskInfo"

htmlContent+=$(cat <<EOF

    </table>

</body>

</html>

EOF

)

# Guardar el resultado en un archivo HTML

outputPath="system\_resources\_report.html"

echo "$htmlContent" > $outputPath

echo "Reporte generado: $outputPath"

Para ejecutar el script, primero se tiene que guardar con la extensión .ps1, como se muestra en la imagen de ejemplo



Antes de ejecutar el script, se debe de otorgar el siguiente permiso

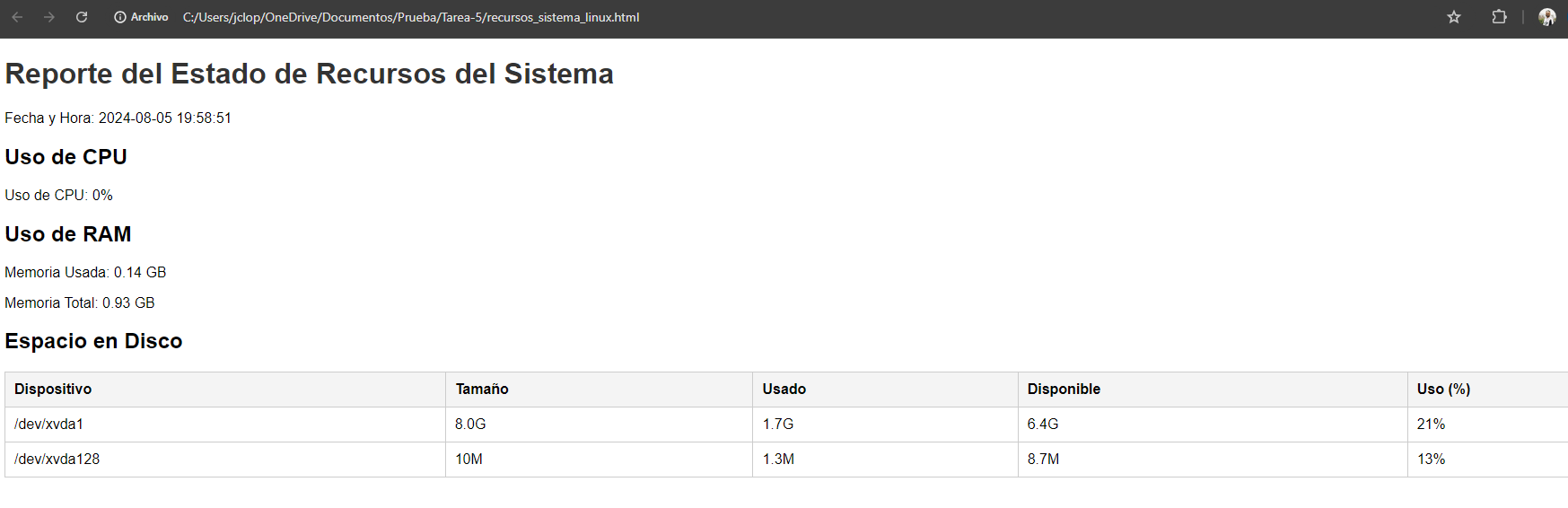


Una vez almacenado, se procede a ejecutarlo con el comando . \ + el nombre del script, que en este caso sería el siguiente:  


Una vez ejecutado, el debería de obtener un resultado en un archivo HTML como se muestra en la siguiente imagen



Procedemos a verificar los datos obtenidos dando doble clic al archivo .html y nos debería de mostrar los datos de la siguiente manera:



Con esta validación, podemos concluir que el script está funcionando correctamente y que nos está mostrando la información que realmente necesitamos.