## **DIPLOMADO FULL STACK DEVELOPER BACK-END Y FRONT-END**

### **MÓDULO 4 - TAREA NODE JS STREAMS**

**DIPLOMANTE: Karina Carrasco Torrejón**

Cochabamba 09/09/2022

**Enunciado:**

Desarrollar un ejemplo simple de uso de streams en node. Puede ser uno de los siguientes: read, write, duplex o transform stream.  
Adjuntar el archivo de código fuente, una breve explicación del código y una captura de la corrida del código.

|  |
| --- |
| **Respuesta:**  Se ha creado un proyecto que contiene varios archivos javascript con varias formas de usar Streams con Node JS, utilizando read, write, duplex y transform stream. Además de controlar algunos problemas que ocurren durante el flujo.   1. backpressure.js 2. duplex.js 3. fileCompressor.js 4. httpUploadFile.js 5. streamVideo.js 6. transform.js |
| GITHUB: <https://github.com/KarinaCarrascoT/Tarea_Ejemplo_Uso_Streams.git> |

**Ejercicio 1:** **backpressure.js**

**Descripción:**

Copia el archivo ./origen/video.mp4 a ./destino/copy.mp4 utilizando streams. Además, controla el problema llamado Backpressure imprimiendo cuando ocurre y se soluciona.

**Conceptos aplicados:**

Ejercicio de ReadStream y WriteStream con control del caso llamado Backpressure. Pausando la lectura si hay el error para acumular(dained) y luego escribir

Backpressure => Contrapresión en streams

Hay un problema general que ocurre durante el manejo de datos llamado contrapresión y describe una acumulación de datos detrás de un búfer durante la transferencia de datos.

Cuando el extremo receptor de la transferencia tiene operaciones complejas, o es más lento por la razón que sea, hay una tendencia a que los datos de la fuente de entrada se acumulen.

**Ejecutar:** node backpressure.js

|  |  |
| --- | --- |
| Antes de Ejecutar: Carpeta Destino está vacía | Luego de copiar, está la copia del video en la carpeta destino y ha mostrado las veces que ha controlado backpressure al hacer el stream read/write |
| Interfaz de usuario gráfica, Texto  Descripción generada automáticamente |  |

**Ejercicio 2**  **duplex.js**

**Descripción:**

Copia el archivo ./origen/video.mp4 a ./destino/copy.mp4 utilizando streams dividendo en pedazos e imprimiendo el tamaño acumulado.

**Conceptos Aplicados:**

Trabaja streams de escritura y lectura con Throttle (demora) y pipe (canalización) de los datos para dar tiempo al manejo de chunk (pedazos) del stream.

**Ejecutar:** node duplex.js

Captura de pantalla de computadora

Descripción generada automáticamente

**Ejercicio 3** **fileCompressor.js**

**Descripción:**

Compresor de archivos utilizando ReadStream, WriteStream con la librería zlib para comprimir a formato .gz

Lee archivo texto.txt dentro de la carpeta texto con uso de stream escribe otro archivo textodestino.txt con el contenido leído de textdestino.txt y lo comprime en un tercer archivo llamado textdestino.txt.gz

**Conceptos Aplicados:**

Trabaja streams de escritura y lectura y utilización de librería zlib para comprimir.

**Instalar:** npm -i zlib

**Ejecutar:** node fileCompressor.js

|  |  |
| --- | --- |
| Carpeta Textos: tiene el archivo Texto.txt | Luego escribe el archivo textodestino.txt en la misma carpeta y crea un archivo compreso textdestino.txt.gz |
|  |  |

**Ejercicio 4** **httpUploadFile.js**

**Descripción:**

Subir archivo seleccionado a una ruta fija, utilizando ReadStream, WriteStream con la librería multiparty para el upload

Crea un servidor que prepara html para la operación de selección y Upload de archivo por el puerto 5050

**Conceptos Aplicados:**

Trabaja streams de escritura y lectura y utilización de librería multiparty

**Instalar:** npm -i multiparty

**Ejecutar el servidor:** node httpUploadFile.js

**En el navegador:** http://localhost:5050

|  |  |
| --- | --- |
| 1. Se corre el servidor | 1. En el navegador escribimos localhost:5050 |
| 1. Seleccionamos el Archivo “GUIA 1.pdf” y Presionamos Upload File | 1. Resultado el archivo “GUIA 1.pdf” sube a la carpeta “destino” |
| 1. Muestra mensaje de File Uploaded |

**Ejercicio 5** **streamVideo.js**

**Descripción:**

Reproducir video utilizando ReadStream usando Tubería pipe y controlando error. Se crea un stream con el contenido de un video en el encabezado del http para reproducirlo en navegador

**Conceptos Aplicados:**

Trabaja streams lectura

**Instalar:** npm -i multiparty

**Ejecutar el servidor:** node streamVideo.js

**En el navegador:** http://localhost:3030

**Capturas: Ejecución del server**

Texto

Descripción generada automáticamente

**Captura del resultado en el navegador:**

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Ejercicio 6 transform.js**

**Descripción:**

Transform stream, transforma el stream de cadenas de entrada a mayúsculas en la salida

**Ejecutar**: node transform.js

Texto

Descripción generada automáticamente