1. Desarrolle en Node.JS un cliente que se conecte a un servidor Web, envíe una petición GET y descargue el archivo solicitado mostrando la cantidad de bytes de la respuesta recibida del servidor web. Su programa deberá usar los módulos https y fs. vamos a descargar este archivo de la pagina:

http://www.informatica.edu.bo/media/noticias/WhatsApp_Image_2022-10-12_at_11.56. 52.jpeq



codigo del e1cliente.js

```
//-----EJERCICIO 1------

//creamos la conexion para conectarse hacia un servidor web usando el
modulo http

const http = require('http');

//vamos a usar los modulos fs y url

var fs = require('gr');

//definimos los parametros para hacer una consulta a la pagina web,

usando metodo get

//http://www.informatica.edu.bo/media/noticias/WhatsApp_Image_2022-10-

12_at_11.56.52.jpeg

parametros = {
    host:'www.informatica.edu.bo',
    path:'/media/noticias/WhatsApp_Image_2022-10-12_at_11.56.52.jpeg',
    port: 80,
    method:'GET'

}

//obtenemos el nombre del archivo a descargar

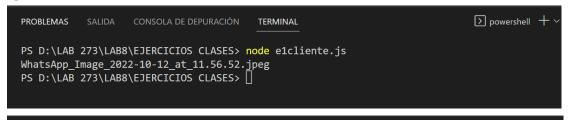
var aux = (url.parse(parametros.path).pathname).split('/')

//guardamosen el nombre del archivo a descargar

var nombrearchivo = aux[aux.length-1];
```

```
//mostramos el nombre del archivo a descargar
console.log(nombrearchivo);
//creamos el archivo a descargar y almacenar
var archivo = fs.createWriteStream(nombrearchivo);
//hacemos una consulta a la pagina web
const cliente = http.request(parametros, function (peticion) {
    //guardamos el archivo de descargar
    peticion.pipe(archivo);
    peticion.on('end', function () {
        //podemos colocar algun mensaje
     });
});
//cerramos la conexion con la pagina web.
cliente.end();
```

ejecutamos el archivo



PROBLEMAS	SALIDA	CONSOLA DE	DEPURACIÓN	TERMINAL	∑ powershell + ∨ □
Mode	LastWriteTime			Length	Name
-a	13/10	0/2022	9:19	1241	e1cliente.js
-a	13/1	0/2022	9:46	803	e2clienteTCP.js
-a	13/1	0/2022	9:45	832	e2servidorTCP.js
-a	13/1	0/2022	9:52	226218	WhatsApp_Image_2022-10-12_at_11.56.52.jpeg
PS D:\LAB 2	273\LAB8	\EJERCICI(OS CLASES> []	

2. Observe el código de los programas TCP proporcionados en la Guía de Laboratorio y notara dos situaciones: a) el cliente no envía ningún dato al servidor, solo realiza la conexión y recibe un mensaje de respuesta, b) el servidor, siempre retorna el mismo mensaje "Hola Mundo".

Ahora deberá modificar ambos programas de manera que las nuevas versiones se ajusten a las siguientes especificaciones:

- a) El programa cliente deberá recibir como argumento el mensaje a enviar, por lo que la ejecución de su programa cliente debería ser invocada con: nodejs programa mensaje.
- b) El programa servidor deberá retornar la cadena recibida en mayúsculas.

```
//EJERCICIO 2-SERVIDOR
//creamos el modulo net TCP
```

```
var net = require('net');
var servidor = net.createServer(function (connection) {
   console.log('cliente connectado');
   connection.on('data', function(mensaje) {
       mensajerecibido = mensaje.toString().toLocaleUpperCase();
       //enviamos el mensaje en mayusculas
       connection.write(mensajerecibido);
   });
   connection.on('end', function () {
      console.log('cliente desconectado');
   });
});
servidor.listen(8080, function () {
   console.log('---el servidor esta conectado---');
});
var net = require('net');
Let mensaje="";
for (let j = 2; j < process.argv.length; j++) {</pre>
   mensaje+=(process.argv[j]+" ");
var cliente = net.connect({ port: 8080 }, function () {
   console.log('--conectado al servidor!--');
   cliente.write(mensaje);
});
cliente.on('data', function (data) {
   mensajerecibido = data.toString();
    console.log(`texto en mayusculas = ${mensajerecibido}`);
```

```
cliente.end();
});

//
cliente.on('end', function () {
    console.log('---desconectado del servidor---');
});
```