|  |  |  |
| --- | --- | --- |
|  | **UNIVERSIDAD AUTONOMA “TOMAS FRIAS” CARRERA DE**   **INGENIERIA DE SISTEMAS** | |
|  | |
| **ESTUDIANTE: Univ. Dietmar Alex Apaza** | |
| **MATERIA: Tecnologías Emergentes** | | |
| **DOCENTE: Ing. Javier Choque Matos** | | **Practica N°2** |
| **AUXILIAR: Univ. Harold Maldonado Garcia** | | **GRUPO: 1** |

**PRACTICA Nº2 AUX-414G1**

Para los siguientes ejercicios se requiere que tenga instalado **node.js** y todos los programas deberán ser corridos por medio de **consola** a su elección (ejecutar de Windows, powerShell, terminal de VS Code, etc.) NO POR LA TERMINAL DEL NAVEGADOR. Así mismo se deber hacer la **captura de los resultados mostrados por CONSOLA como el copiar y pegar el código JavaScript en el archivo pdf** (el código se copia no es captura)

Para poder hacer correr el programa se debe estar en el directorio donde se encuentra el archivo .js y usar el comando: **node [nombre programa.js]**

**NOTA:** Para concatenar usar Plantillas de cadena de texto (Template literal) **`…${…} `**

1. **Arrays.** Hacer un programa en JavaScript para: **(30PTS)**
   * Mostrar el auto de marca Toyota del siguiente array. Y hacer un ciclo para mostrar la marca con más letras.

// autos = [‘Nissan’, ‘Honda’, ‘Ford’, ‘Toyota’, ‘Alfa Romeo’]

// Toyota

Ej. // Alfa Romeo

let auto = 'Toyota';

let marcas = [];

let autos = ['Nissan', 'Honda', 'Ford', 'Toyota', 'Alfa Romero'];

marcas.push(autos);

let autoLargo = [0];

for (let i = 0; i < autos.length; i++) {

        if (autoLargo.length <= autos[i].length) {

            autoLargo = autos[i];

        }

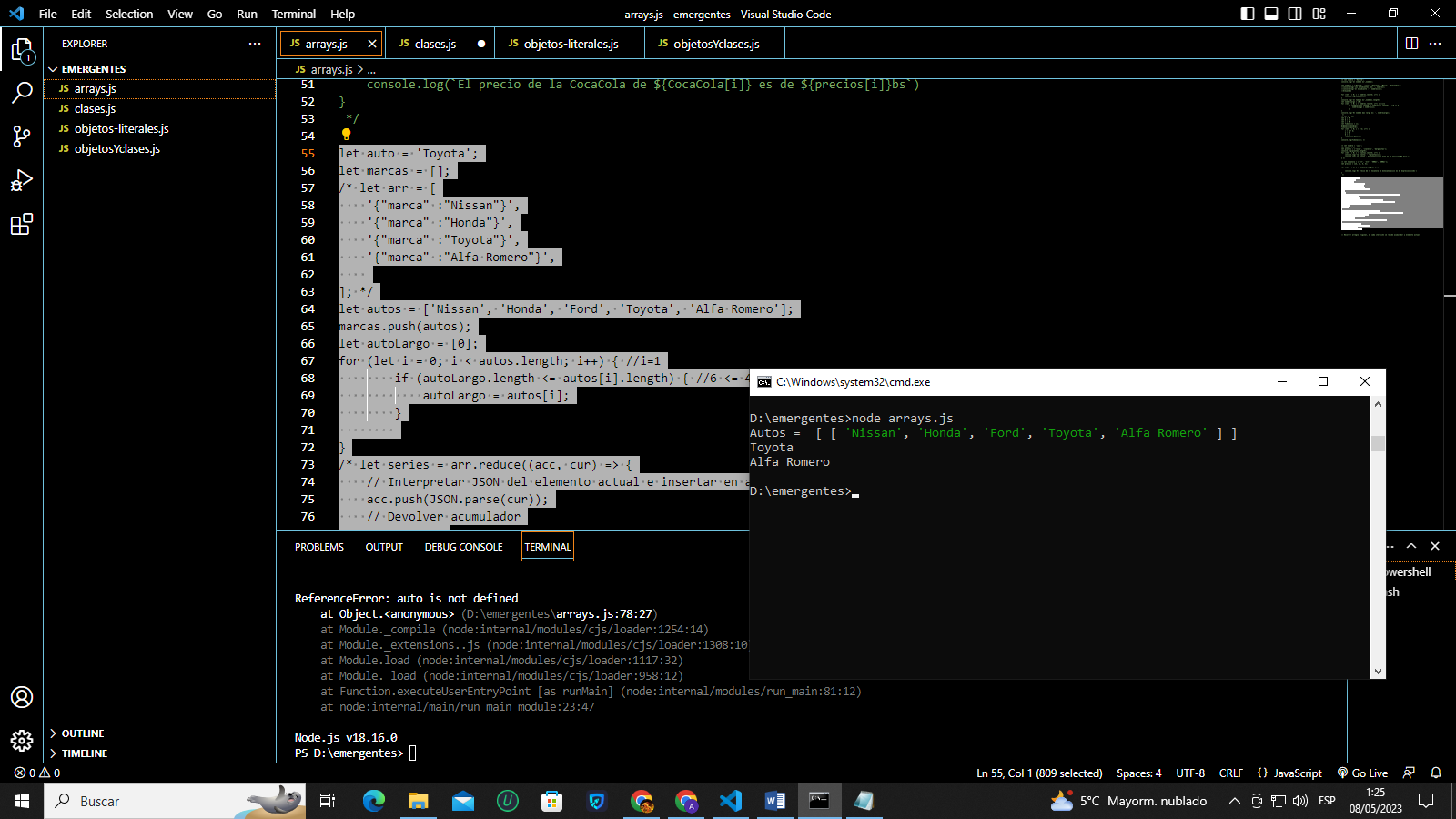
}

console.log('Autos = ',marcas);

console.log(auto);

console.log(autoLargo);

**Consola**



* + Mostrar los 7 primeros números Fibonacci

Ej. // [0, 1, 1, 2, 3, 5, 8]

let n = 7;

let a = 0;

let b = 1;

let c = 0;

let fibonacci = [];

fibonacci.push(a);

fibonacci.push(b);

for (let i = 0; i < n-2; i++) {

    c = a + b;

    a = b;

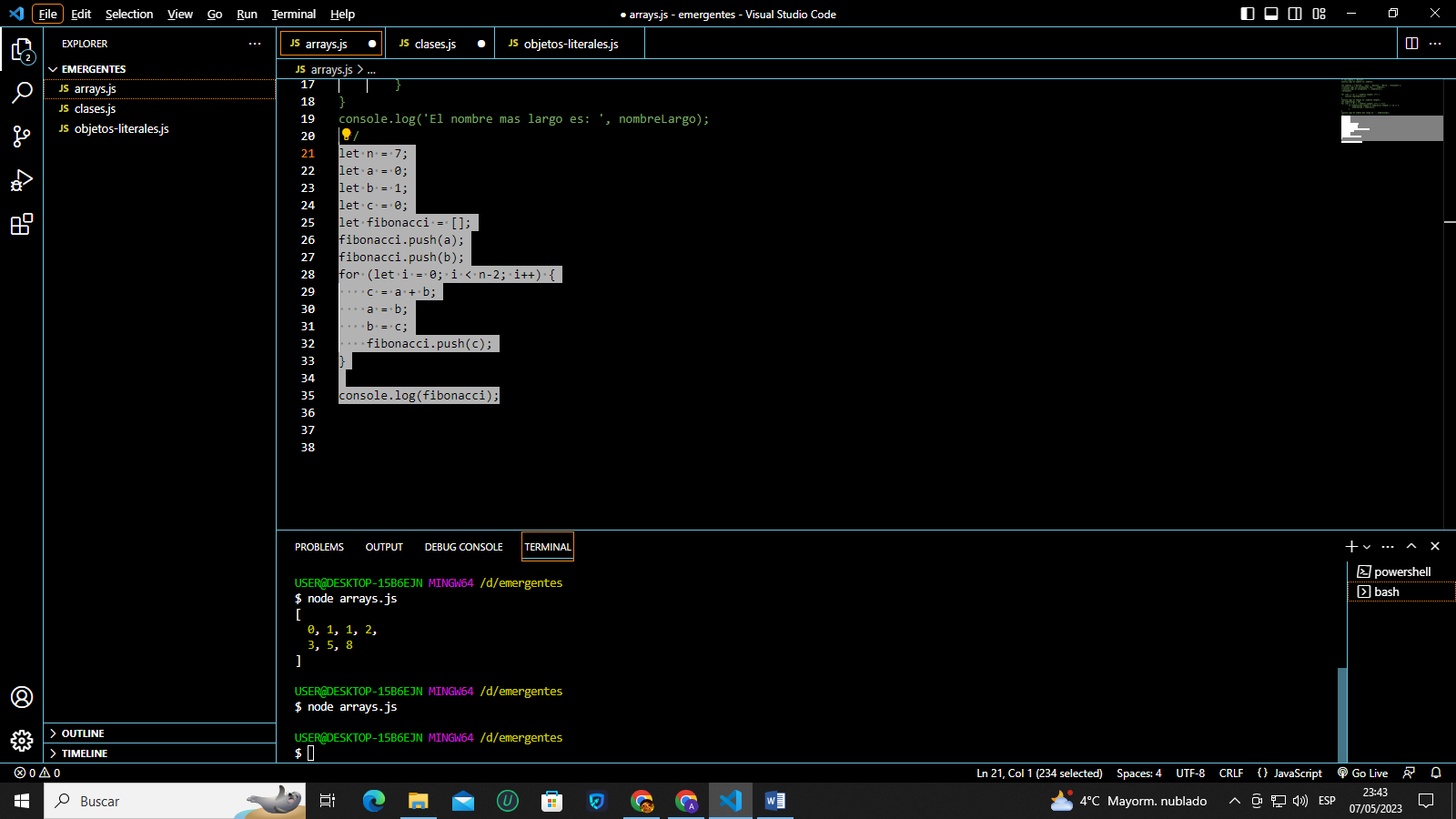
    b = c;

    fibonacci.push(c);

}

console.log(fibonacci);

Resultado



* + Separar los números pares e impares del siguiente array [2, 6, 8, 15, 17, 22, 7, 14]; además muestre el número mayor.

Ej. // [2, 6, 8, 22, 14]

Ej. // [15, 17, 7]

Ej. // 22

* + Hacer un programa que barra array por array para concatenar todos los siguientes productos con sus precios de los siguientes arrays.

cocaCola = [‘3lt’, ‘2lt’, ‘500ml’, ‘300ml’];

precio = [13, 10, 5, 3];

Ej.

// El precio de la CocaCola de 3lt es de 13bs

// El precio de la CocaCola de 2lt es de 10bs

// El precio de la CocaCola de 500ml es de 5bs

// El precio de la CocaCola de 300ml es de 3bs

let CocaCola = ['3lt', '2lt', '500ml', '300ml'];

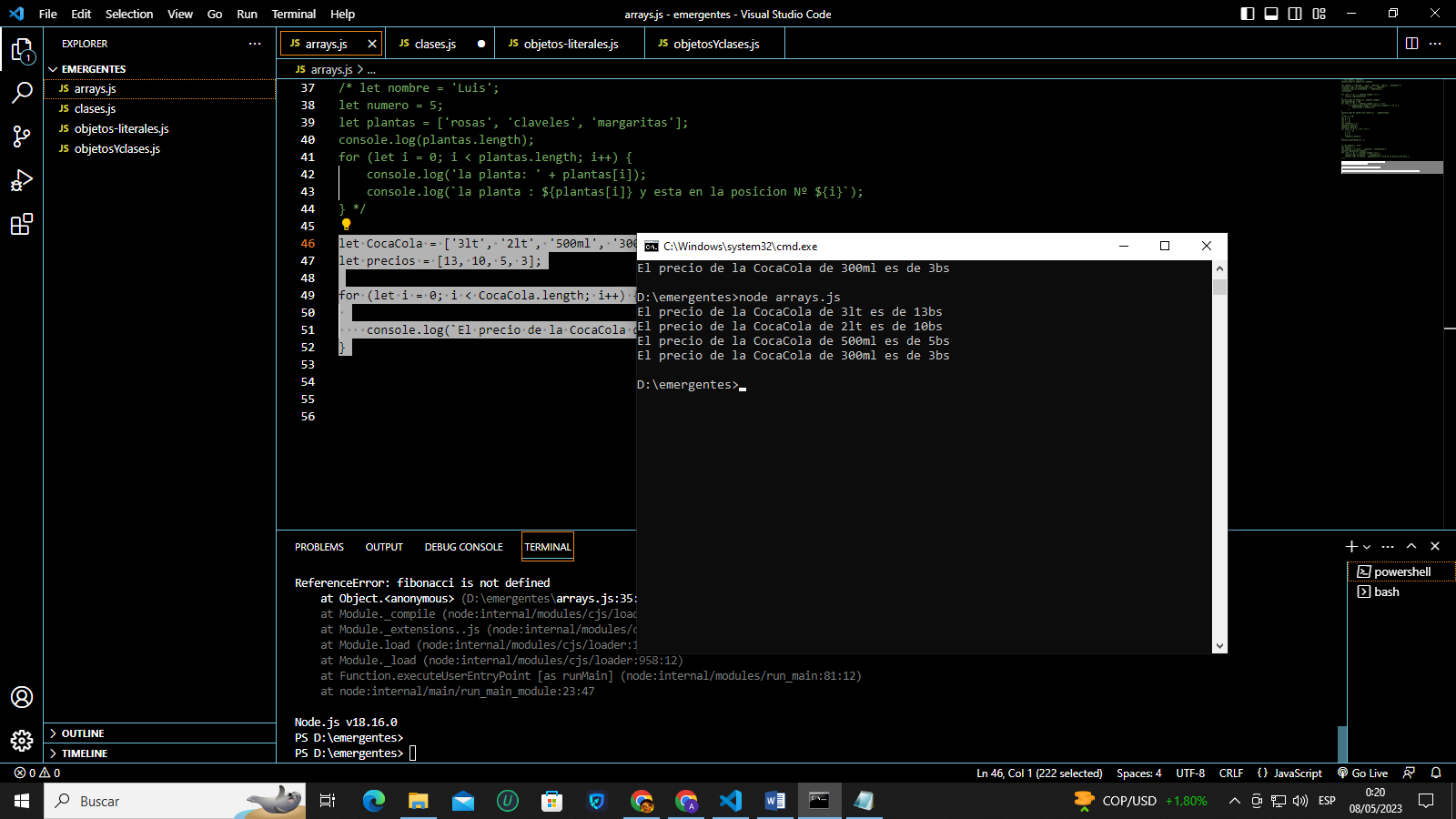
let precios = [13, 10, 5, 3];

for (let i = 0; i < CocaCola.length; i++) {

    console.log(`El precio de la CocaCola de ${CocaCola[i]} es de ${precios[i]}bs`)

}

Resultado



1. **Objetos literales.** Crea un objeto literal referido a una celebridad famosa o personaje ficticio en el que se le asigne sus propiedades, dicho personaje debe tener como minimo 5 propiedades y debe incluir al menos un tipo de dato (String, number, boolean y Array). Además, se le deberá dar dos **métodos** como mínimo dentro del objeto (**Explicar que**

**hacen los métodos dados**). **(25PTS)**

let teletubbies ={

    nombre:'Serie Teletubbies',

    integrantes:5,

    genero:'infantil',

    personajes:['Tinky', 'Winky', 'Dipsy', 'Laa-Laa', 'Po'],

    activos: false,

    informacion(){

        console.log(`la ${this.nombre} tiene ${this.integrantes} integrantes`);

    }

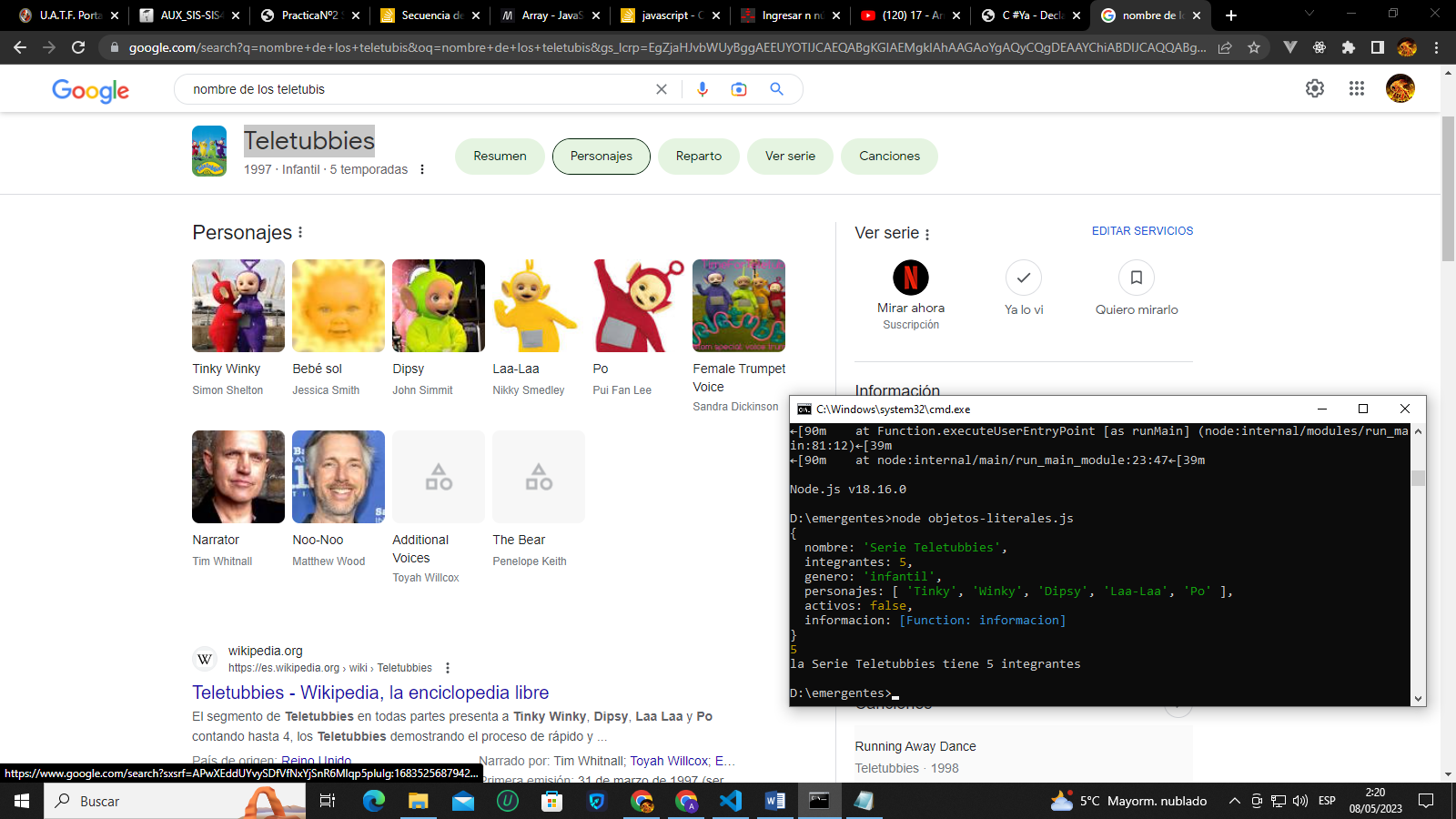
}

console.log(teletubbies);

console.log(teletubbies.integrantes);

teletubbies.informacion();

Consola



1. **Clases.** Crear una clase que pueda crear objetos referido a celebridades (futbolistas, cantantes, actores, influencers, personajes de la historia, etc.) o personajes ficticios.

**(25PTS)**

* + Crear al menos 4
  + La clase debe tener al menos 2 métodos (**explicar que hace el método y mostrar ejemplos usando dichos métodos**)

class Grupos {

    constructor (a, b, c, d) {

        this.nombre = a;

        this.genero = b;

        this.integrantes = c;

        this.activo = d;

    }

    integrantesNuevos(integrante){

        this.integrantes = integrante;

        console.log(`El grupo aun tiene ${this.integrantes}`)

    }

}

const BlackPink = new Grupos ('Banda BlackPink', 'K-POP', 4, true);

const Kjarkas = new Grupos ('Grupo Kjarkas', 'Folcklore', 9, true);

const Octavia = new Grupos ('Banda Octavia', 'Rock', 5, true);

const Los\_Daltons = new Grupos ('Grupo Los\_Daltons', 'Cumbia', 5, false);

console.log('la banda BlackPink:', BlackPink);

BlackPink.integrantesNuevos(4);

console.log('el grupo Kjarkas:', Kjarkas);

BlackPink.integrantesNuevos(9);

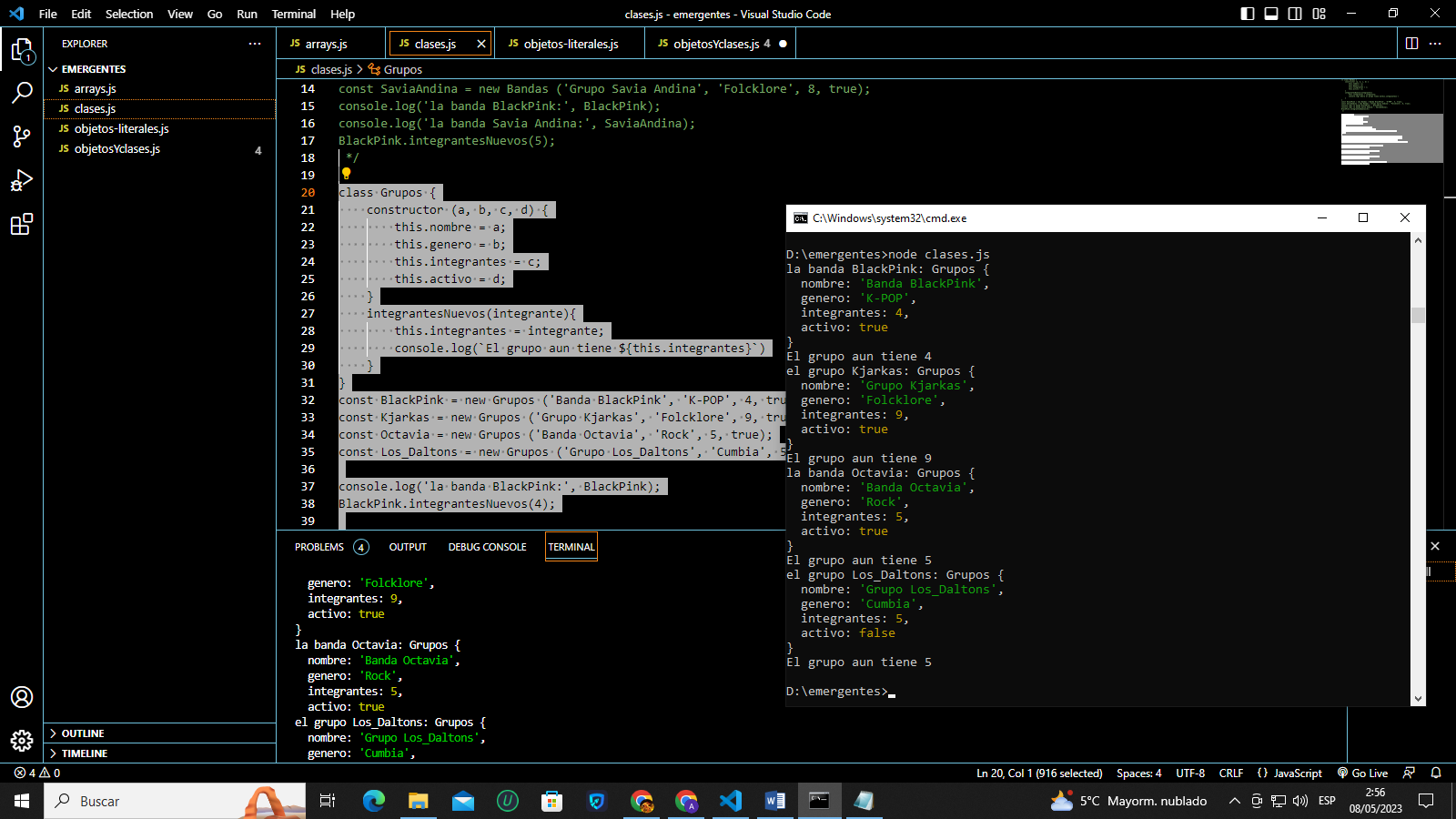
console.log('la banda Octavia:', Octavia);

BlackPink.integrantesNuevos(5);

console.log('el grupo Los\_Daltons:', Los\_Daltons);

BlackPink.integrantesNuevos(5);

**Consola**



1. **Pegar el link del repositorio de github o gitlab (a su elección).**  **(20PTS)**

https://github.com/AlexTheLion99/Apaza\_Dietmar\_Alex\_AUX414\_1\_2023