

## Laboratorio #2

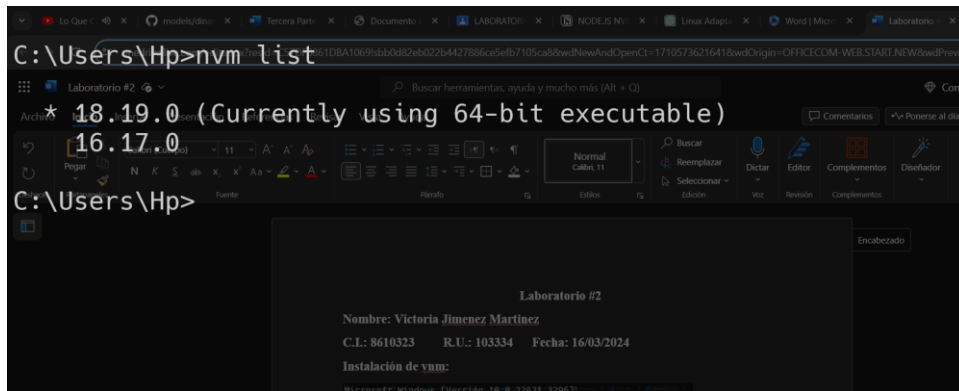
Nombre: Victoria Jimenez Martinez

C.I.: 8610323      R.U.: 103334      Fecha: 16/03/2024

Instalación de vnm:



Lista:



Uso de versiones:

```
C:\Users\Hp>nvm list
18.19.0
* 16.17.0 (Currently using 64-bit executable)

C:\Users\Hp>nvm use 16.17.0
Now using node v16.17.0 (64-bit)

C:\Users\Hp>nvm list
18.19.0
* 16.17.0 (Currently using 64-bit executable)

C:\Users\Hp>nvm use 18.19.0
Now using node v18.19.0 (64-bit)

C:\Users\Hp>nvm list
* 18.19.0 (Currently using 64-bit executable)
16.17.0

C:\Users\Hp>
```

## Yarn

```
C:\Users\Hp>npm install --global yarn

changed 1 package in 4s
npm notice
npm notice New major version of npm available! 9.6.2 -> 10.5.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.0
npm notice Run npm install -g npm@10.5.0 to update!
npm notice

C:\Users\Hp>
```

## Paquetes:

```
C:\Users\Hp>npm install --g typescript

changed 1 package in 7s

C:\Users\Hp>tsc -v
Version 5.4.2

C:\Users\Hp>
```

## Node y la ejecución de archivos .js y .ts :

```
C:\Users\Hp>npm install --g typescript
changed 1 package in 7s

C:\Users\Hp>tsc -v
Version 5.4.2

C:\Users\Hp>cd desktop

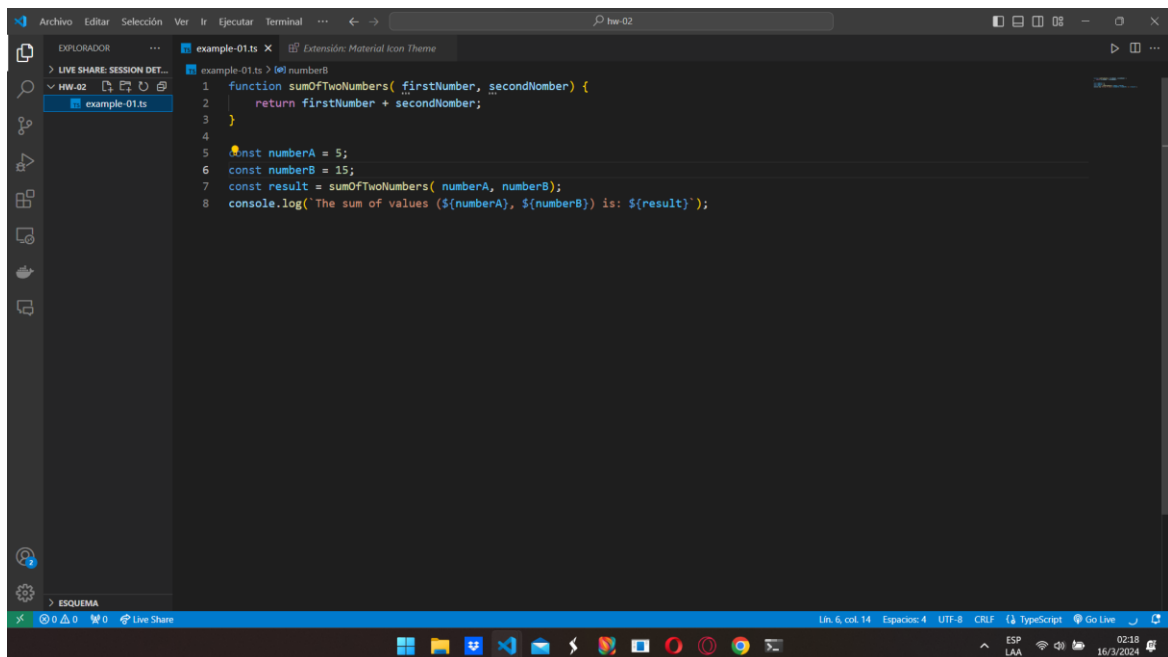
C:\Users\Hp\Desktop>cd Seminario

C:\Users\Hp\Desktop\Seminario>mkdir hw-02

C:\Users\Hp\Desktop\Seminario>cd hw-02

C:\Users\Hp\Desktop\Seminario\hw-02>code

C:\Users\Hp\Desktop\Seminario\hw-02>
```



```
C:\Users\Hp\Desktop\Seminario\hw-02>tsc .\example-01.ts

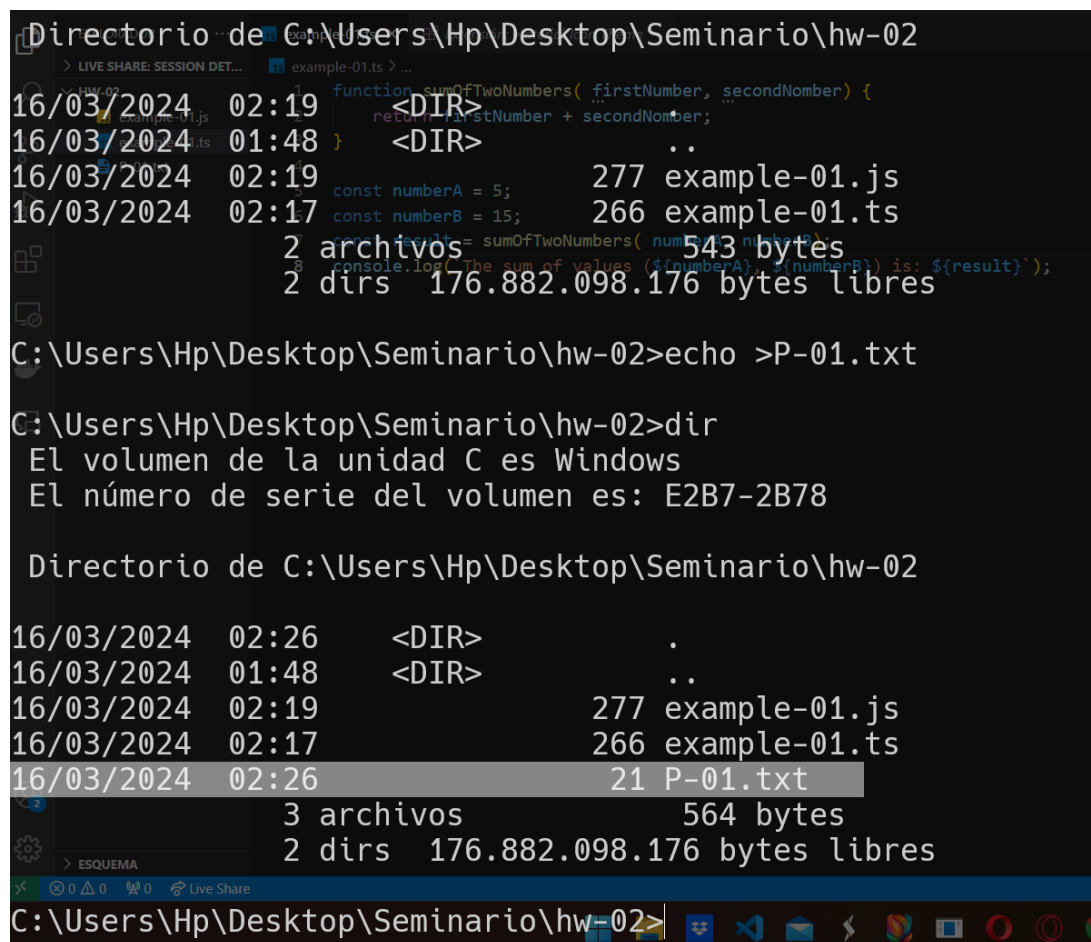
C:\Users\Hp\Desktop\Seminario\hw-02>node .\example-01.ts
The sum of values (5, 15) is: 20

C:\Users\Hp\Desktop\Seminario\hw-02>
```

## Evaluación

\*\*\* Todos los archivos deben crearse en la carpeta de la siguiente manera .txt para aquellas preguntas que no requieran de código y .js o .ts para aquellas que requieran hacer uso de esos lenguajes para completar la tarea \*\*\*

1.- En la siguiente línea de código, se pueden observar el uso de comillas invertidas, estas se usan para la creación de template Strings, nombra algunas características importantes y como es que funcionan los anteriormente mencionados “template Strings” y como estos se diferencian de las comillas simples y dobles. (Deberá de crearse un archivo con el nombre P-01.txt).



The screenshot shows a Windows File Explorer window and a Command Prompt window. The File Explorer window displays the directory C:\Users\Hp\Desktop\Seminario\hw-02, which contains two files: example-01.js (277 bytes) and example-01.ts (266 bytes). The Command Prompt window shows the execution of the command 'echo >P-01.txt', which creates a new file P-01.txt in the same directory. The Command Prompt also shows the output of the 'dir' command, which lists the files and their sizes.

```
Directorio de C:\Users\Hp\Desktop\Seminario\hw-02
16/03/2024 02:19 <DIR> .
16/03/2024 01:48 <DIR> ..
16/03/2024 02:19 277 example-01.js
16/03/2024 02:17 266 example-01.ts
2 archivos 543 bytes
2 dirs 176.882.098.176 bytes libres

C:\Users\Hp\Desktop\Seminario\hw-02>echo >P-01.txt

C:\Users\Hp\Desktop\Seminario\hw-02>dir
El volumen de la unidad C es Windows
El número de serie del volumen es: E2B7-2B78

Directorio de C:\Users\Hp\Desktop\Seminario\hw-02
16/03/2024 02:26 <DIR> .
16/03/2024 01:48 <DIR> ..
16/03/2024 02:19 277 example-01.js
16/03/2024 02:17 266 example-01.ts
16/03/2024 02:26 21 P-01.txt
3 archivos 564 bytes
2 dirs 176.882.098.176 bytes libres
```

```
1 Template Strings en JavaScript:
2 También conocidos como "template literals", son una característica introducida en ECMAScript 6 que ofrece una forma más flexible
3 Característica Importante:
4
5 Interpolación de variables:
6 Una de las características más útiles de los template strings es la capacidad de incluir variables directamente dentro del string
7 Ejemplo:
8 console.log(`The sum of values (${numberA}, ${numberB}) is: ${result}`);
9
10 Multilínea:
11 Los template strings pueden abarcar múltiples líneas de texto sin necesidad de utilizar caracteres de escape como `\\n`. Esto mejo
12 Ejemplo:
13 const mensaje = `Hola ¿cómo estas?, me llamo Victoria Jimenez Martinez.`;
14
15 Expresiones dentro de placeholders:
16 Además de variables, también se pueden incluir expresiones JavaScript válidas dentro de los placeholders `${...}`. Esto permite r
17 Ejemplo:
18 const a = 10;
19 const b = 5;
20 console.log(`El resultado de la suma de ${a} y ${b} es ${a + b}.`);
21
22
23
24 Los template strings son una adición poderosa a JavaScript que facilita la creación de strings complejos de una manera más legibl
```

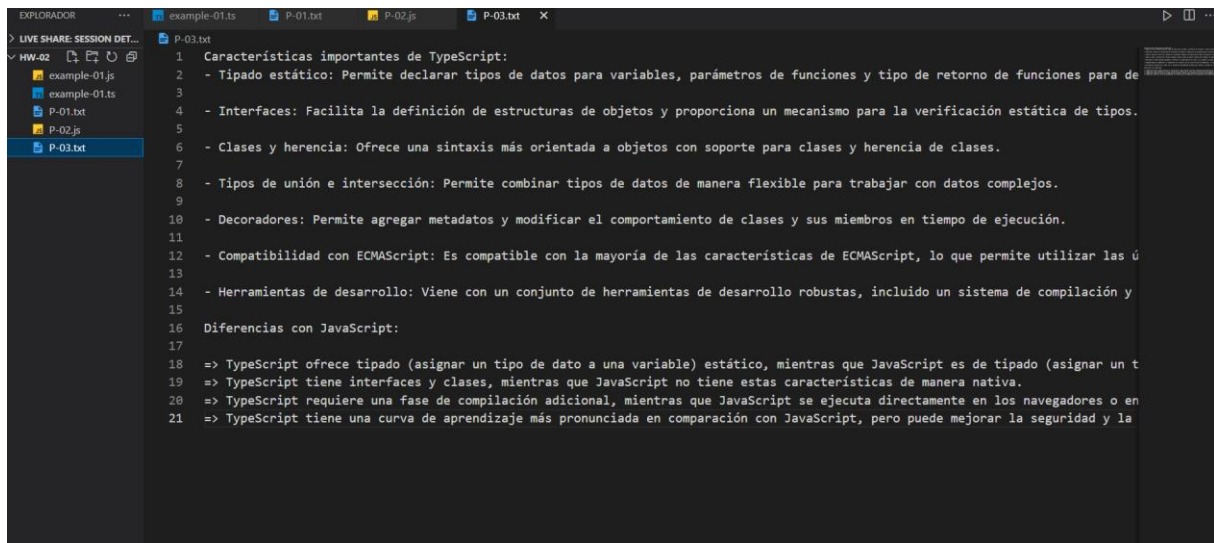
**2.- Crea una función que salude a un usuario, la función contara con al menos 1 parámetro, posteriormente muestre capturas de la consola, no olvides ejecutar el código haciendo uso de node en la terminal, como se mostró anteriormente. (Deberá de crearse un archivo con el nombre P-02.js).**

```
1 function SaludarPregunta02(Nombre) {
2   console.log(`Hola,${Nombre} !! Bienvenido`)
3 }
4
5 SaludarPregunta02("Victoria");
6 SaludarPregunta02("Alan");
7 SaludarPregunta02("Daniel");
```

```
C:\Users\Hp\Desktop\Seminario\hw-02>node .\P-02.js
Hola,Victoria !! Bienvenido
Hola,Alan !! Bienvenido
Hola,Daniel !! Bienvenido
C:\Users\Hp\Desktop\Seminario\hw-02>
```

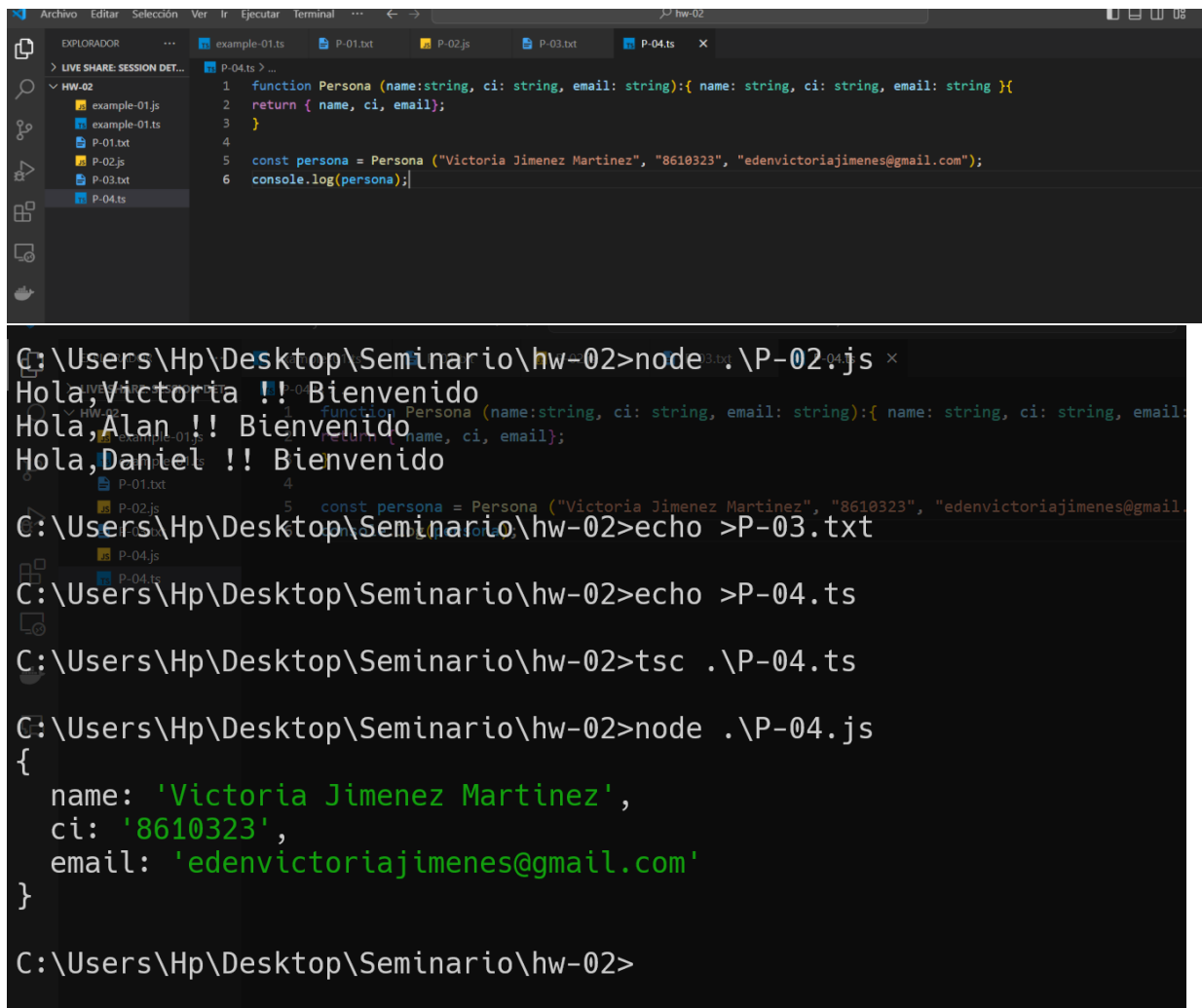
**3.- TypeScript talvez te suene familiar, pero sin duda es muy relevante en cuanto al desarrollo web. Nombra algunas características importantes de TypeScript y que**

diferencia se tiene si lo comparamos con JavaScript. (Deberá de crearse un archivo con el nombre P-03.txt).



```
1 Características importantes de TypeScript:
2 - Tipado estático: Permite declarar tipos de datos para variables, parámetros de funciones y tipo de retorno de funciones para de
3
4 - Interfaces: Facilita la definición de estructuras de objetos y proporciona un mecanismo para la verificación estática de tipos.
5
6 - Clases y herencia: Ofrece una sintaxis más orientada a objetos con soporte para clases y herencia de clases.
7
8 - Tipos de unión e intersección: Permite combinar tipos de datos de manera flexible para trabajar con datos complejos.
9
10 - Decoradores: Permite agregar metadatos y modificar el comportamiento de clases y sus miembros en tiempo de ejecución.
11
12 - Compatibilidad con ECMAScript: Es compatible con la mayoría de las características de ECMAScript, lo que permite utilizar las ú
13
14 - Herramientas de desarrollo: Viene con un conjunto de herramientas de desarrollo robustas, incluido un sistema de compilación y
15
16 Diferencias con JavaScript:
17
18 => TypeScript ofrece tipado (asignar un tipo de dato a una variable) estático, mientras que JavaScript es de tipado (asignar un t
19
20 => TypeScript tiene interfaces y clases, mientras que JavaScript no tiene estas características de manera nativa.
21
22 => TypeScript requiere una fase de compilación adicional, mientras que JavaScript se ejecuta directamente en los navegadores o en
23
24 => TypeScript tiene una curva de aprendizaje más pronunciada en comparación con JavaScript, pero puede mejorar la seguridad y la
```

**4.- Crea una función en TypeScript, que reciba 3 parámetros en el siguiente orden, name: string, ci: string, email: string, dicha función deberá de retornar un objeto con los parámetros como atributos del objeto. (Deberá de crearse un archivo con el nombre P-04.ts).**

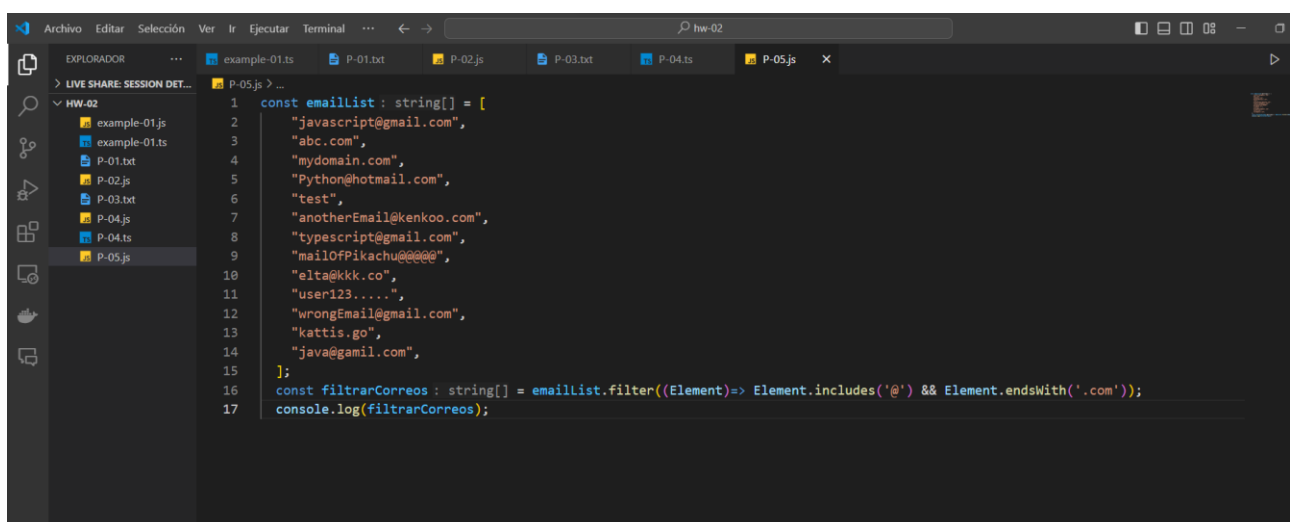


The image shows a VS Code editor with a file explorer on the left containing files like example-01.ts, P-01.txt, P-02.js, P-03.txt, and P-04.ts. The main editor displays the content of P-04.ts, which defines a function `Persona` and a constant `persona` with specific values. Below the editor, a terminal window shows the execution of `node .\P-02.js`, which prints three lines of text: `Hola,Victoria !! Bienvenido`, `Hola,Alan !! Bienvenido`, and `Hola,Daniel !! Bienvenido`. Subsequent terminal commands include `echo >P-03.txt`, `echo >P-04.ts`, `tsc .\P-04.ts`, and `node .\P-04.js`. The final command's output is a JSON object: `{ name: 'Victoria Jimenez Martinez', ci: '8610323', email: 'edenvictoriajimenes@gmail.com' }`.

```
1 function Persona (name:string, ci: string, email: string):{ name: string, ci: string, email: string }{
2   return { name, ci, email};
3 }
4
5 const persona = Persona ("Victoria Jimenez Martinez", "8610323", "edenvictoriajimenes@gmail.com");
6 console.log(persona);
```

```
C:\Users\Hp\Desktop\Seminarior\hw-02>node .\P-02.js
Hola,Victoria !! Bienvenido
Hola,Alan !! Bienvenido
Hola,Daniel !! Bienvenido
C:\Users\Hp\Desktop\Seminarior\hw-02>echo >P-03.txt
C:\Users\Hp\Desktop\Seminarior\hw-02>echo >P-04.ts
C:\Users\Hp\Desktop\Seminarior\hw-02>tsc .\P-04.ts
C:\Users\Hp\Desktop\Seminarior\hw-02>node .\P-04.js
{
  name: 'Victoria Jimenez Martinez',
  ci: '8610323',
  email: 'edenvictoriajimenes@gmail.com'
}
C:\Users\Hp\Desktop\Seminarior\hw-02>
```

**5. Dada una lista de emails, usted tiene la tarea de filtrar solo los emails validos, un email valido será aquel que contenga el caracter '@' y la terminación '.com'. El resultado esperado debería ser el siguiente:**



The image shows a VS Code editor with a file explorer on the left containing files like example-01.ts, P-01.txt, P-02.js, P-03.txt, P-04.ts, and P-05.js. The main editor displays the content of P-05.js, which defines an array `emailList` containing various email addresses and a function `filtrarCorreos` that filters the array to only include valid emails (containing '@' and ending in '.com').

```
1 const emailList : string[] = [
2   "javascript@gmail.com",
3   "abc.com",
4   "mydomain.com",
5   "Python@hotmail.com",
6   "test",
7   "anotherEmail@kenkoo.com",
8   "typescript@gmail.com",
9   "mailOfPikachu@@",
10  "elita@kkk.co",
11  "user123.....",
12  "wrongEmail@gmail.com",
13  "kattis.go",
14  "java@gamil.com",
15 ];
16 const filtrarCorreos : string[] = emailList.filter((Element)=> Element.includes('@') && Element.endsWith('.com'));
17 console.log(filtrarCorreos);
```

```
C:\Users\Hp\Desktop\Seminarario\hw-02>echo >P-05.js
C:\Users\Hp\Desktop\Seminarario\hw-02>node .\P-05.js
[
  3      "abc.com",
  4      "mydomain.com",
  5      "python@hotmail.com",
  6      "es",
  7      "anotherEmail@kenkoo.com",
  8      "typescript@gmail.com",
  9      "asuloffkachu@eee",
 10      "kkkk.co",
 11      "User123.....",
 12      "wrongEmail@gmail.com",
 13      "kattis.go",
 14      "java@gamil.com",
 15  ]
C:\Users\Hp\Desktop\Seminarario\hw-02> emailList.filter((Element)=> Element.includes('@') && Element.endsWith('.com'));
17 console.log(filtrarCorreos);
```