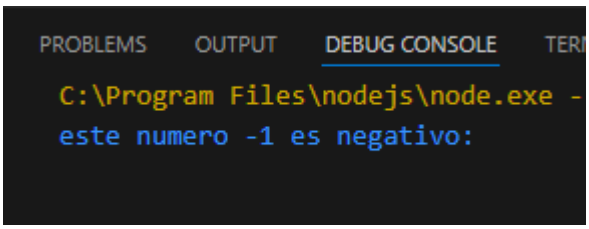
Practica N° 1

Apellidos: IBAÑEZ SOLIS
Nombres: ELIAN MAURICIO
C.I.: 9981267 LP
Fecha: 22 DE SEPTIEMBRE
Sigla: INF 131

1. **Pedir un número y mostrar si es positivo, negativo o cero (if/else)**.

```
let n  = '-1';
n = parseInt(n);
if (n > 0) {
    console.log('este numero '+ n +' es positivo:');
} else {
    if (n ==0) {
        console.log('este numero '+ n +' es cero:');
    } else {
        console.log('este numero '+ n +' es negativo:');
    }
}
```

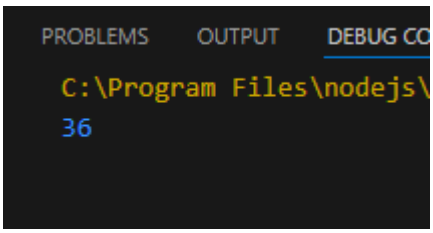
CAPTURA



2. Crear una función flecha que calcule el cuadrado de un número.

```
const cuadrado = n =>{
    return n*n;
}
console.log(cuadrado(6));
```

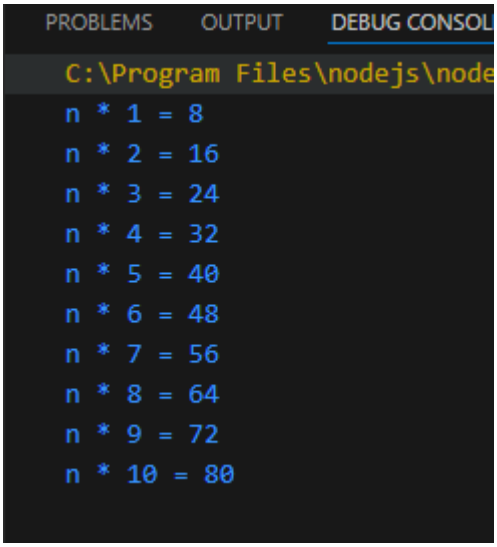
CAPTURA



3. Pedir un número y mostrar su tabla de multiplicar del 1 al 10.

```
let n = 8;
for (let i = 1; i <= 10; i++) {
    let m = n*i;
    console.log('n * '+ i +' = '+ m)
}
```

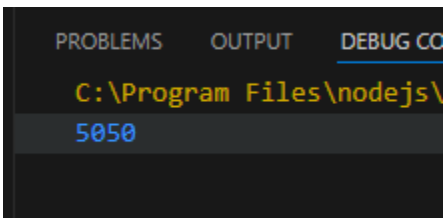
CAPTURA



4. Sumar los números del 1 al 100 usando un ciclo while.

```
let i = 1;
let suma = 0;
while (i <=100) {
    suma = suma + i;
    i = i+1;
}
console.log(suma);
```

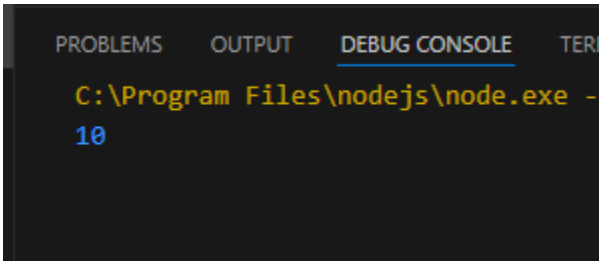
CAPTURA



5. Crear una función anónima que reciba dos números y devuelva el mayor.

```
const mayor = function(a,b){
    if (a>b) {
        console.log(a);
    } else {
        console.log(b);
    }
}
mayor(10,9);
```

CAPTURA

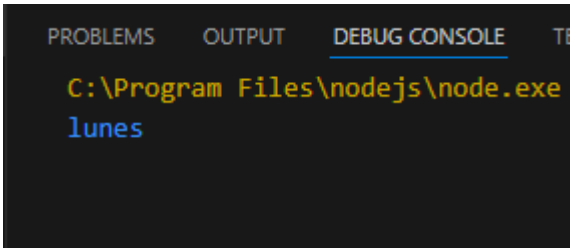


6. Usar switch para mostrar el nombre del día de la semana según un número del 1 al 7.

```
let dia = '1';
switch (dia) {
  case '1':
    console.log('lunes');
    break;

  case '2':
    console.log('martes');
    break;

  case '3':
    console.log('miercoles');
    break;
  case '4':
    console.log('jueves');
    break;
  case '5':
    console.log('viernes');
    break;
  case '6':
    console.log('sabado');
```

```
    break;
  case '7':
    console.log('domingo');
    break;


  default:

    break;
}
```

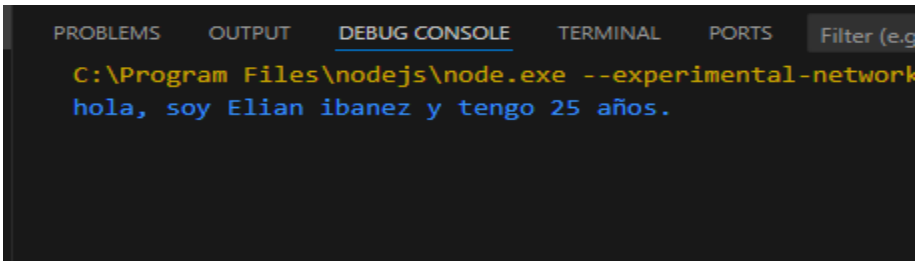
CAPTURA



7. Crear un objeto persona con nombre y edad, y mostrar un mensaje: “Hola, soy [nombre] y tengo [edad] años”.

```
let  persona = {
    nombreP: "Elian ibanez",
    edad :25,

    saludar: function(){
        console.log("hola, soy "+ this.nombreP + " y tengo "+ this.edad+ "
años.")
    }
}
persona.saludar();
```

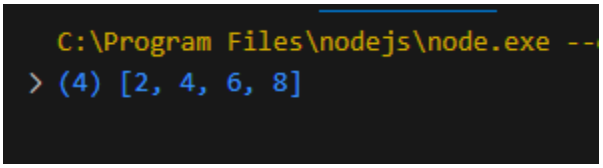
CAPTURA



8. Recorrer un arreglo de números y mostrar solo los pares.

```
let areglo = [1,2,3,4,5,6,7,8,9];
let lista=[];
for (let i = 0; i < areglo.length; i++) {
    if (areglo[i]%2==0) {
        lista.push(areglo[i]);
    }

}
console.log(lista);
```

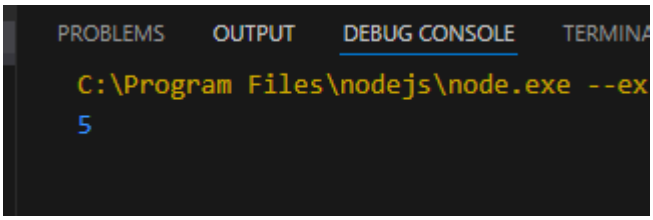
CAPTURA



9. Contar cuántos números impares hay en un arreglo.

```
let areglo = [1,2,3,4,5,6,7,8,9];
let suma = 0;
for (let i = 0; i < areglo.length; i++) {
    if (areglo[i] % 2 != 0) {
        suma++;
    }
}
console.log(suma);
```

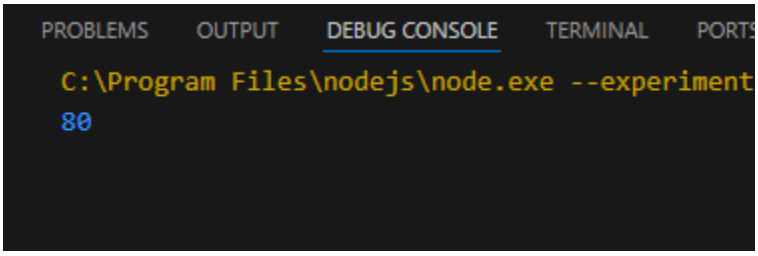
CAPTURA



10. Crear una función que reciba un objeto rectangulo (base, altura) y devuelva su área.

```
function area(rectangulo){
    let a = rectangulo.base * rectangulo.altura;
    return a;
}
 let rectangulo = {base: 8, altura: 10};
 console.log(area(rectangulo));
```

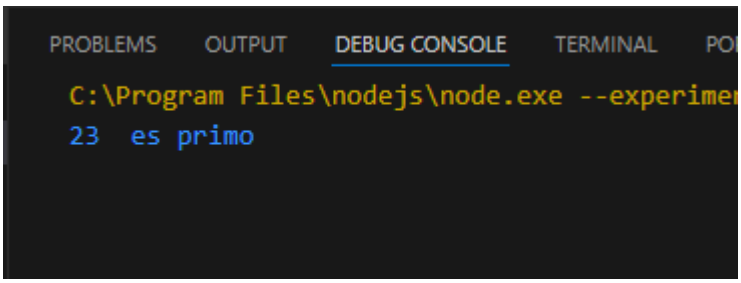
CAPTURA



11. Escribir un programa que determine si un número ingresado es primo.

```
let n = 23;
let suma=0;
for (let i = 2; i <= Math.sqrt(n); i++) {
    if (n%i==0) {
        suma = suma+1;
    }
}
if (suma==0 & n>1) {
    console.log(n+'  es primo')
}else{
    console.log(n+' no es primo')
}
```

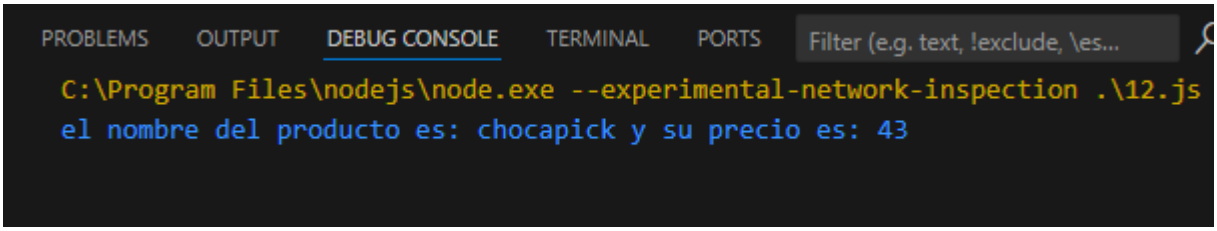
CAPTURA



12. Crear un objeto producto con nombre y precio, y una función flecha que devuelva un string con esa información.

```
let producto = {nombre: 'chocapick', precio: 43};

const parrafo = () => {
    console.log('el nombre del producto es: '+ producto.nombre+ ' y su precio es: '+producto.precio)
}
parrafo(producto);
```

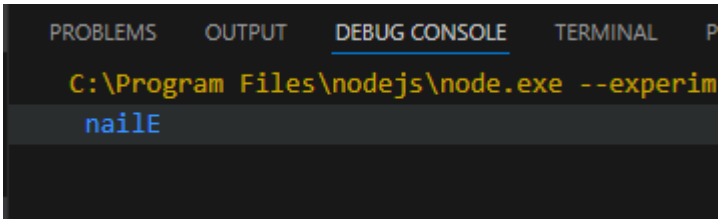
CAPTURA



13. Invertir una cadena sin usar .reverse().

```
let cadena =  "Elian";
let Cvacia = " ";
for (let i = cadena.length-1 ; i >= 0 ; i--){
    Cvacia = Cvacia+cadena[i];
}
console.log(Cvacia);
```

CAPTURA

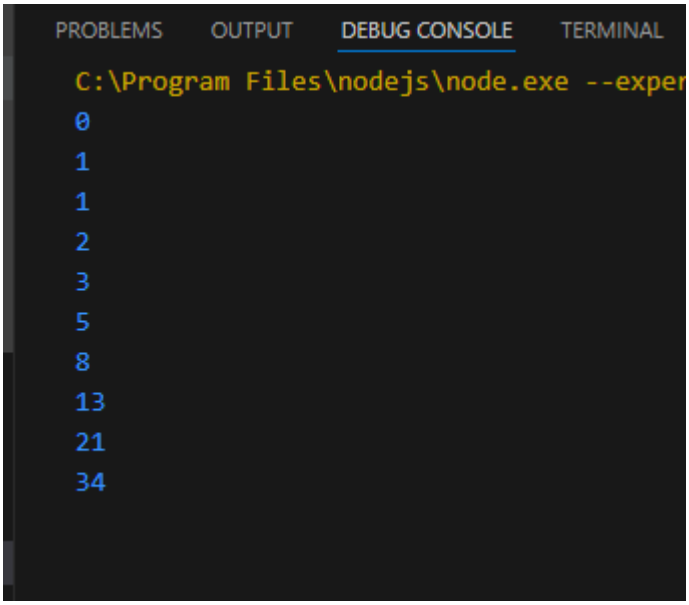


14. Generar los primeros 10 números de la serie de Fibonacci con un ciclo.

```
let a = 0;
let b = 1;
console.log(a);
console.log(b);
let c = a+b;
for (let i = 0; i < 8; i++) {
```

```
    console.log(c);
    a=b;
    b=c;
    c = a+b;
}
```

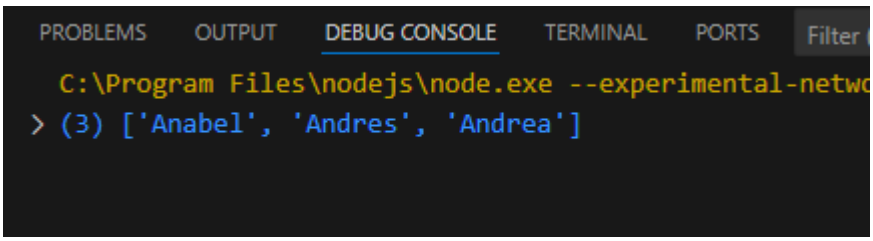
CAPTURA



15. Recorrer un arreglo de nombres y mostrar solo los que empiezan con la letra “A”.

```
let nombres = ['elian','Anabel','juan','Andres','Andrea','Carla'];
let nombresV = [];

for (let i = 0; i < nombres.length; i++) {
    if (nombres[i][0]=='A') {
      nombresV.push(nombres[i]);
    }
    //console.log(nombres[i][0]);

}
console.log(nombresV);
```

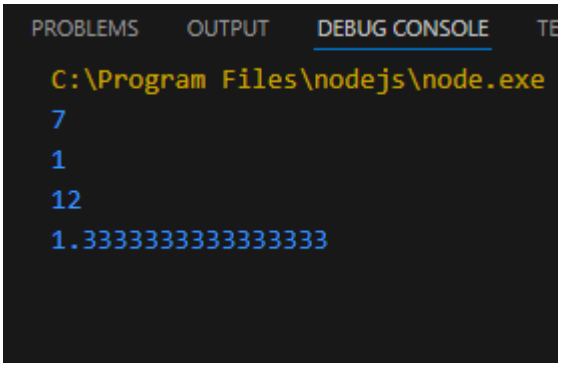
CAPTURA



16. Crear un objeto calculadora con métodos para sumar, restar, multiplicar y dividir.

```
let calculadora = {
    sumar : function(n1,n2){
        console.log(n1+n2);
    },
    restar : function(n1,n2){
        console.log(n1-n2);
    },
    multiplica : function(n1,n2){
        console.log(n1*n2);
    },
    dividir : function(n1,n2){
        console.log(n1/n2);
    }
};
calculadora.sumar(4,3);
calculadora.restar(4,3);
calculadora.multiplica(4,3);
calculadora.dividir(4,3);
```

CAPTURA



17. Recorrer un objeto y mostrar sus claves y valores sin usar Object.entries().

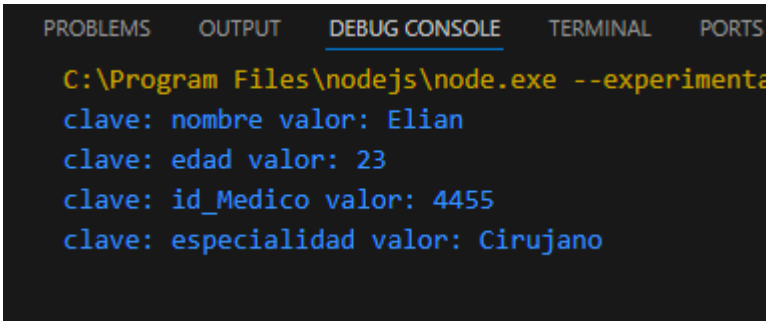
```
let medico={
    nombre: 'Elian',
    edad: 23,
    id_Medico: 4455,
    especialidad: 'Cirujano',
}
for (const clave in medico) {
    let valor = medico[clave];
    console.log('clave: ' + clave + ' valor: '+ valor);
```

```
}
```

18. Crear un arreglo de objetos alumnos con nombre y nota, y mostrar los aprobados (nota ≥ 51).

```
let alumnos=[
    {nombre: 'Elian', nota: 89, edad: 25, RU: 242343},
    {nombre: 'Juan', nota: 66, edad: 19, RU:24244 },
    {nombre: 'Andrea', nota: 41, edad: 18, RU:24124 },
    {nombre: 'Carlos', nota: 66, edad: 24, RU: 24121},
    {nombre: 'Gustavo', nota:23 , edad: 30, RU: 123112},
    {nombre: 'Lucas', nota: 51, edad:21 , RU: 22123},
];
console.log('APROBADOS');
for (let i = 0; i < alumnos.length; i++) {
    if (alumnos[i].nota>=51){
        console.log(alumnos[i]);
    }

}
```

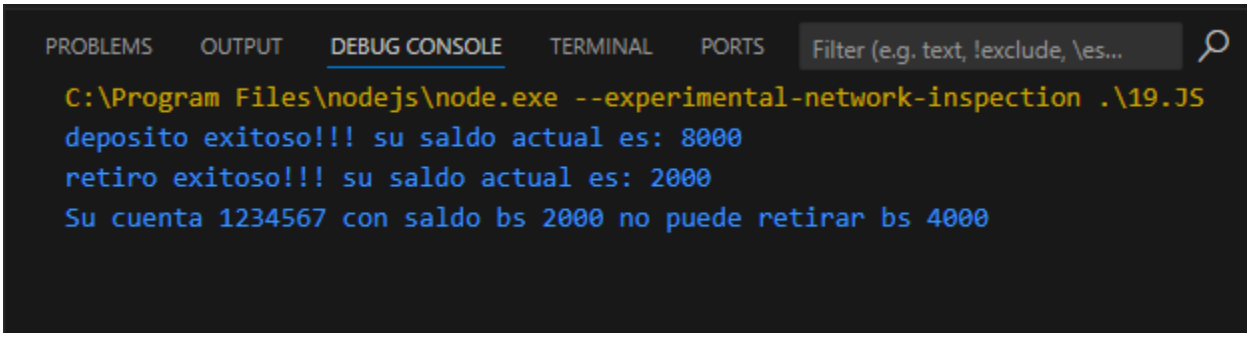
CAPTURA



19. Crear un objeto cuentaBancaria con saldo y métodos depositar y retirar (usar condicional si no hay saldo suficiente).

```
let CuentaBancaria = {
    nroCuenta: 1234567,
    saldo: 3000,

    depositar : function(Dmonto){
        this.saldo=this.saldo+Dmonto;
        console.log('deposito exitoso!!! su saldo actual es: ' + this.saldo);
    },
    retirar : function(Rmonto){
        if (this.saldo>=Rmonto) {
            this.saldo=this.saldo-Rmonto;
            console.log('retiro exitoso!!! su saldo actual es: ' + this.saldo);
        } else {
            console.log('Su cuenta ' + this.nroCuenta+ ' con saldo bs
'+this.saldo + ' no puede retirar bs ' +Rmonto );
        }
    }
}
CuentaBancaria.depositar(5000);
CuentaBancaria.retirar(6000);
CuentaBancaria.retirar(4000);
```

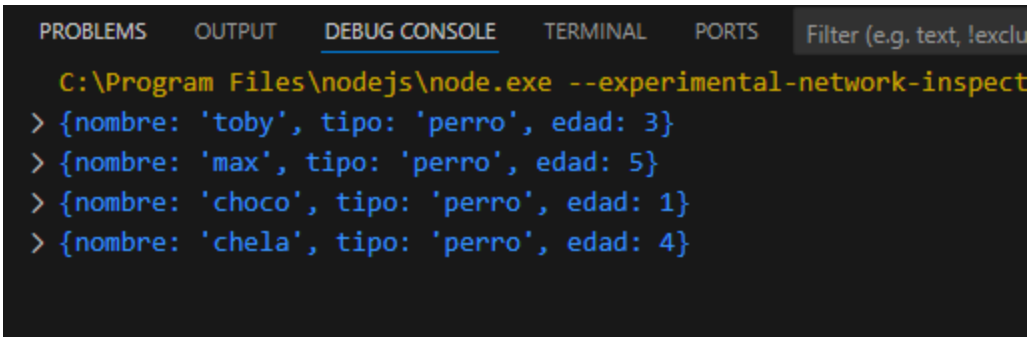
CAPTURA



20.Crear un arreglo de objetos mascotas y recorrerlo para mostrar solo las que sean de tipo “perro”.

```
let mascotas = [
    {nombre: 'toby', tipo: 'perro',edad:3},
    {nombre: 'pelusa', tipo: 'gato',edad:1},
    {nombre: 'rodolfo', tipo: 'loro',edad:6},
    {nombre: 'max', tipo: 'perro',edad:5},
    {nombre: 'tommy', tipo: 'gato',edad:3},
    {nombre: 'choco', tipo: 'perro',edad:1},
    {nombre: 'chiquitin', tipo: 'hamster',edad:7},
    {nombre: 'chela', tipo: 'perro',edad:4},
];
for (let i = 0; i < mascotas.length; i++) {
    if (mascotas[i].tipo=='perro') {
        console.log(mascotas[i]);
    }
}
```

CAPTURA

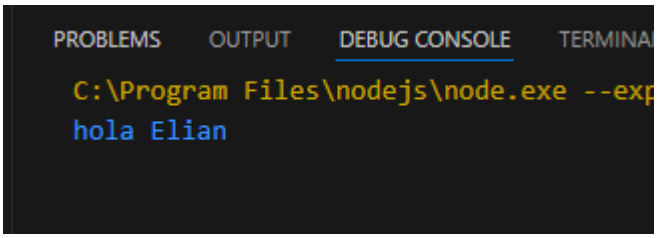


PROMESAS + ASYNC/AWAIT

1. Crear una promesa que devuelva un mensaje después de 2 segundos y mostrarlo en consola.

```
const m = new Promise((r) => {
  setTimeout(() => {
    r("hola Elian");
  }, 2000);
});

m.then((mensaje) => {
  console.log(mensaje);
});
```

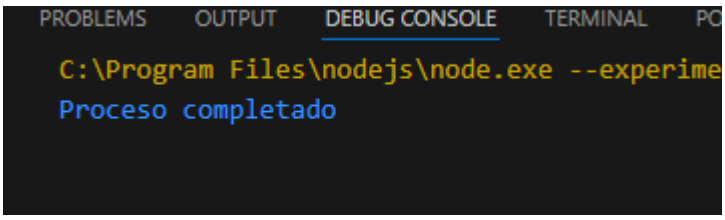
CAPTURA



2. Escribir una función async que espere 3 segundos y luego muestre “Proceso completado”.

```
function esperarTresSegundos() {
  return new Promise((r) => {
    setTimeout(() => {
      r();
    }, 3000);
  });
}

async function ejecutarProceso() {
  await esperarTresSegundos();
  console.log("Proceso completado");
}

ejecutarProceso();
```

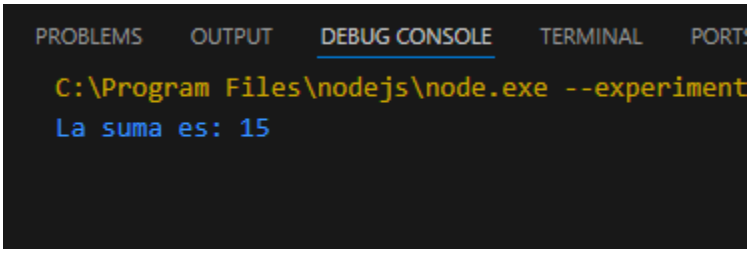
CAPTURA



3. Crear dos promesas que devuelvan números distintos y usar Promise.all para sumarlos.

```
const promesa1 = new Promise((r) => {
  setTimeout(() => r(5), 1000);
});

const promesa2 = new Promise((r) => {
  setTimeout(() => r(10), 1500);
});

Promise.all([promesa1, promesa2])
  .then(([num1, num2]) => {
    const suma = num1 + num2;
    console.log("La suma es:", suma);
  });
```

CAPTURA

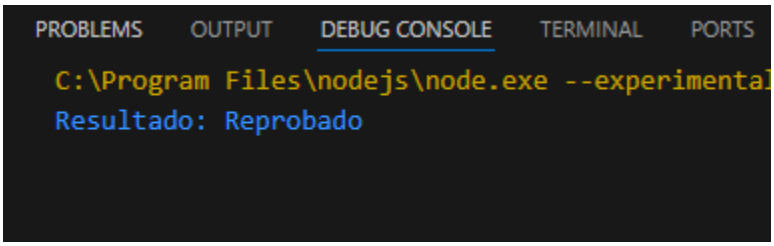


4. Hacer una promesa que simule el resultado de “aprobado” o “reprobado” y consumirla con await.

```
function rExamen() {
  return new Promise((r) => {
    const nota = Math.random() * 10;
    if (nota >= 6) {
      r("Aprobado");
    } else {
```

```
      r("Reprobado");
    }
  });
}

async function verificarResultado() {
  const resultado = await rExamen();
  console.log("Resultado:", resultado);
}

verificarResultado();
```

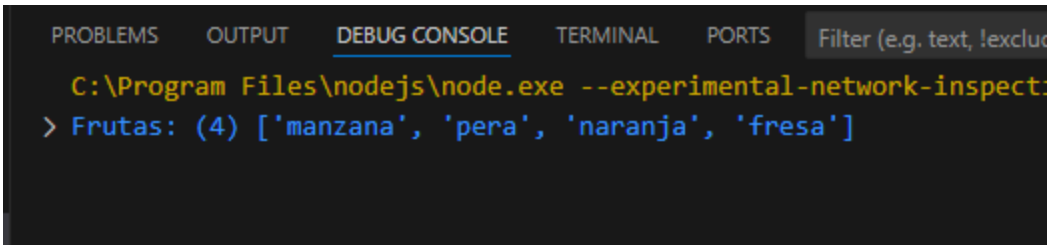
CAPTURA



5. Crear una promesa que devuelva un arreglo de frutas y mostrarlo usando async/await.

```
function obtenerFrutas() {
  return new Promise((r) => {
    setTimeout(() => {
      r(["manzana", "pera", "naranja", "fresa"]);
    }, 1000);
  });
}
async function mostrarFrutas() {
  const frutas = await obtenerFrutas();
  console.log("Frutas:", frutas);
}
mostrarFrutas();
```

CAPTURA


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Filter (e.g. text, !exclu
C:\Program Files\nodejs\node.exe --experimental-network-inspect
> Frutas: (4) ['manzana', 'pera', 'naranja', 'fresa']