

Prueba Practica.

Nombre del estudiante:

Emilio Ñacato

Elaborado por:

Lcdo. Diego Medardo Saavedra García, Mgtr.

**Profesor Ocasional de la Universidad de las Fuerzas Armadas ESPE
Departamento de Ciencias de la Computación.**

1.2 Variables y operaciones

1.2.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es una variable y para qué sirve?

Una variable es un contenedor que almacena un valor en la memoria de un programa. Sirve para almacenar y manipular datos en un programa.

- ¿Cuál es la diferencia entre declarar e inicializar una variable?

Declarar una variable significa establecer su existencia y tipo de dato, mientras que inicializar una variable significa asignarle un valor específico por primera vez.

- ¿Cuál es la diferencia entre sumar números y concatenar strings?

Sumar números es una operación aritmética que realiza la suma matemática de los valores numéricos, mientras que concatenar strings es unir dos o más cadenas de texto en una sola.

- ¿Cuál operador me permite sumar o concatenar?

El operador "+" permite sumar números y concatenar strings.

1.2.2 Determina el nombre y tipo de dato para almacenar en variables la siguiente información:

- Nombre: string
- Apellido: string
- Nombre de usuario en ESPE: string
- Edad: number(entero)
- Correo electrónico: string
- Mayor de edad: boolean

- Dinero ahorrado: number(decimal)
- Deudas: number(decimal)

1.2.3 Traduce a código JavaScript las variables del ejemplo anterior y deja tu código en los comentarios.

```
> let nombre = "Emilio";  
  let apellido = "Niacato";  
  let nombreUsuarioESPE = "ejnacato";  
  let edad = 21;  
  let correoElectronico = "ejnacato@espe.edu.ec";  
  let mayorDeEdad = true;  
  let dineroAhorrado = 5000.00;  
  let deudas = 2000.00;
```

1.2.4 Calcula e imprime las siguientes variables a partir de las variables del ejemplo anterior:

- Nombre completo (nombre y apellido)
- Dinero real (dinero ahorrado menos deudas)

```
let nombre = "Emilio";  
let apellido = "Niacato";  
let nombreUsuarioESPE = "ejnacato";  
let edad = 21;  
let correoElectronico = "ejnacato@espe.edu.ec";  
let mayorDeEdad = true;  
let dineroAhorrado = 5000.00;  
let deudas = 2000.00;  
  
let nombreCompleto = nombre + " " + apellido;  
let dineroReal = dineroAhorrado - deudas;  
  
console.log(nombreCompleto);  
console.log(dineroReal);  
  
Emilio Niacato  
3000
```

1.3 Funciones

1.3.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es una función?

Una función es un bloque de código reutilizable que realiza una tarea específica.

- ¿Cuándo me sirve usar una función en mi código?

Una función es útil para evitar repetir el mismo código en diferentes partes del programa y para modularizar el código en unidades lógicas y manejables.

- ¿Cuál es la diferencia entre parámetros y argumentos de una función?

Los parámetros son variables que se definen en la declaración de la función, mientras que los argumentos son los valores reales que se pasan a la función al llamarla.

1.3.2 Convierte el siguiente código en una función, pero, cambiando cuando sea necesario las variables constantes por parámetros y argumentos en una función:

```
> let name = "Diego Medardo";  
let lastname = "Saavedra García";  
let nickname = "statick";  
  
function nombre(name, lastname, nickname) {  
    let nombreCompleto = name + " " + lastname;  
    console.log("Mi nombre es " + nombreCompleto + ", pero prefiero  
que me digas " + nickname + ".");  
}  
  
nombre(name, lastname, nickname);  
  
Mi nombre es Diego Medardo Saavedra García, pero prefiero VM201:7  
que me digas statick.
```

1.4 Condicionales

1.4.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es un condicional?

Un condicional es una estructura de control que permite ejecutar diferentes bloques de código dependiendo de una condición determinada.

- ¿Qué tipos de condicionales existen en JavaScript y cuáles son sus diferencias?

Los más comunes que existen son: if, if else, switch, for y while.

If: Quiere decir la sentencia “si” en caso que la sentencia sea verdadera ejecutara una decisión ya sea desde imprimir un mensaje, hacer un cálculo o ejecutar otra sentencia, esto sucederá siempre y cuando la sentencia de entrada sea verdadera.

If else: Es similar a if, primero analizará en caso que sea verdadera ejecutará una acción, pero en caso de ser falsa este ejecutará otra acción distinta. Switch: Este es más usado en el caso de tener varias opciones, en este caso se lo usará cuando se tiene muchas condiciones encadenadas llenas de decisiones, se lo suele usar más como un menú de opciones.

For: Esta condición es de ciclo repetitivo donde ejecutara una acción repetidamente hasta que llegue a su límite de entrada, ya se imprimir varios mensajes o incrementar un valor de una variable. Se utiliza cuando se conoce el número de veces que se repetirá el ciclo, o se puede conocer mediante otras funciones del lenguaje como length.

While: Es un ciclo repetitivo que permite ejecutar un código de forma repetitiva siempre que una condición sea verdadera y hasta que esa condición sea falsa.

- ¿Puedo combinar funciones y condicionales?

Sí, se pueden combinar funciones y condicionales para realizar acciones específicas dependiendo de ciertas condiciones.

1.4.2 Replica el comportamiento del siguiente código que usa la sentencia switch utilizando if, else y else if:

javascript

```
const tipoDeSuscripcion = "Basic"; if
(tipoDeSuscripcion === "Free") { console.log("Solo
puedes tomar los cursos gratis");
} else if (tipoDeSuscripcion === "Basic") {
    console.log("Puedes tomar casi todos los cursos de ESPE durante un
mes");
} else if (tipoDeSuscripcion === "Expert") {
    console.log("Puedes tomar casi todos los cursos de ESPE durante un
año");
} else if (tipoDeSuscripcion === "ExpertPlus") {
```

```
console.log("Tú y alguien más pueden tomar TODOS los cursos de
ESPE durante un año");
```

```
}
```

```
let tipoDeSuscripcion = "Basic";

if (tipoDeSuscripcion === "Free") {
  console.log("Solo puedes tomar los cursos gratis");
} else if (tipoDeSuscripcion === "Basic") {
  console.log("Puedes tomar casi todos los cursos de ESPE durante un
mes");
} else if (tipoDeSuscripcion === "Expert") {
  console.log("Puedes tomar casi todos los cursos de ESPE durante un
año");
} else if (tipoDeSuscripcion === "ExpertPlus") {
  console.log("Tú y alguien más pueden tomar TODOS los cursos de
ESPE durante un año");
}
```

Puedes tomar casi todos los cursos de ESPE durante un mes [VM233:6](#)

1.4.3 Replica el comportamiento de tu condicional anterior con if, else y else if, pero ahora solo con if (sin else ni else if).

```
let tipoDeSuscripcion = "Basic";

if (tipoDeSuscripcion === "Free") {
  console.log("Solo puedes tomar los cursos gratis");
}

if (tipoDeSuscripcion === "Basic") {
  console.log("Puedes tomar casi todos los cursos de ESPE durante un mes");
}

if (tipoDeSuscripcion === "Expert") {
  console.log("Puedes tomar casi todos los cursos de ESPE durante un año");
}

if (tipoDeSuscripcion === "ExpertPlus") {
  console.log("Tú y alguien más pueden tomar TODOS los cursos de ESPE durante un
año");
}
```

Puedes tomar casi todos los cursos de ESPE durante un mes

[VM297:3](#)

1.5 Ciclos

1.5.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es un ciclo?

Un ciclo es una estructura de control que permite repetir un bloque de código varias veces.

- ¿Qué tipos de ciclos existen en JavaScript?

En JavaScript, existen los ciclos "for", "while" y "do-while".

- ¿Qué es un ciclo infinito y por qué es un problema?

Un ciclo infinito es aquel que se repite continuamente sin una condición de finalización adecuada. Es un problema porque puede llevar a un consumo excesivo de recursos y hacer que el programa se bloquee.

- ¿Puedo mezclar ciclos y condicionales?

Sí, se pueden mezclar ciclos y condicionales para controlar la repetición de un bloque de código basado en una condición.

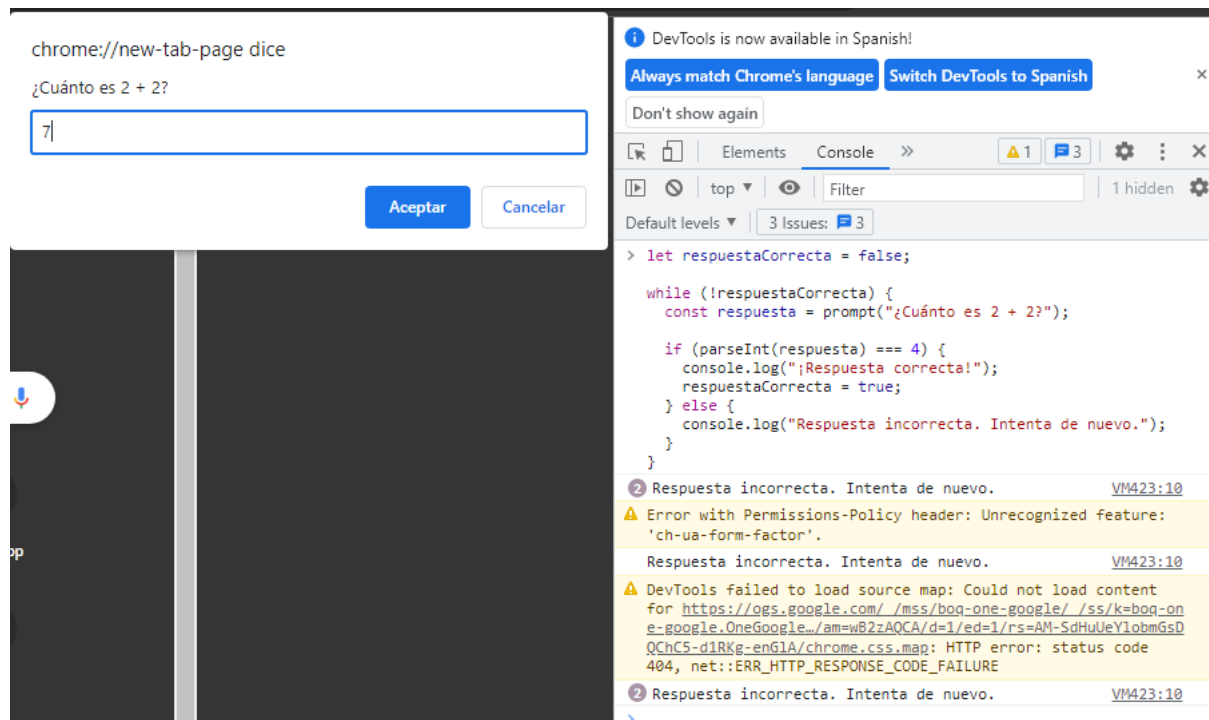
1.5.2 Replica el comportamiento de los siguientes ciclos for utilizando ciclos while:

```
> let i = 0;
while (i < 5) {
  console.log("El valor de i es: " + i);
  i++;
}

i = 10;
while (i >= 2) {
  console.log("El valor de i es: " + i);
  i--;
}
```

El valor de i es: 0
El valor de i es: 1
El valor de i es: 2
El valor de i es: 3
El valor de i es: 4
El valor de i es: 10
El valor de i es: 9
El valor de i es: 8
El valor de i es: 7
El valor de i es: 6
El valor de i es: 5
El valor de i es: 4
El valor de i es: 3
El valor de i es: 2

1.5.3 Escribe un código en JavaScript que le pregunte a los usuarios cuánto es $2 + 2$. Si responden bien, mostramos un mensaje de felicitaciones, pero si responden mal, volvemos a empezar.



chrome://new-tab-page dice

¿Cuánto es 2 + 2?

7

Aceptar Cancelar

```

let respuestaCorrecta = false;
while (!respuestaCorrecta) {
  const respuesta = prompt("¿Cuánto es 2 + 2?");

  if (parseInt(respuesta) === 4) {
    console.log("¡Respuesta correcta!");
    respuestaCorrecta = true;
  } else {
    console.log("Respuesta incorrecta. Intenta de nuevo.");
  }
}

```

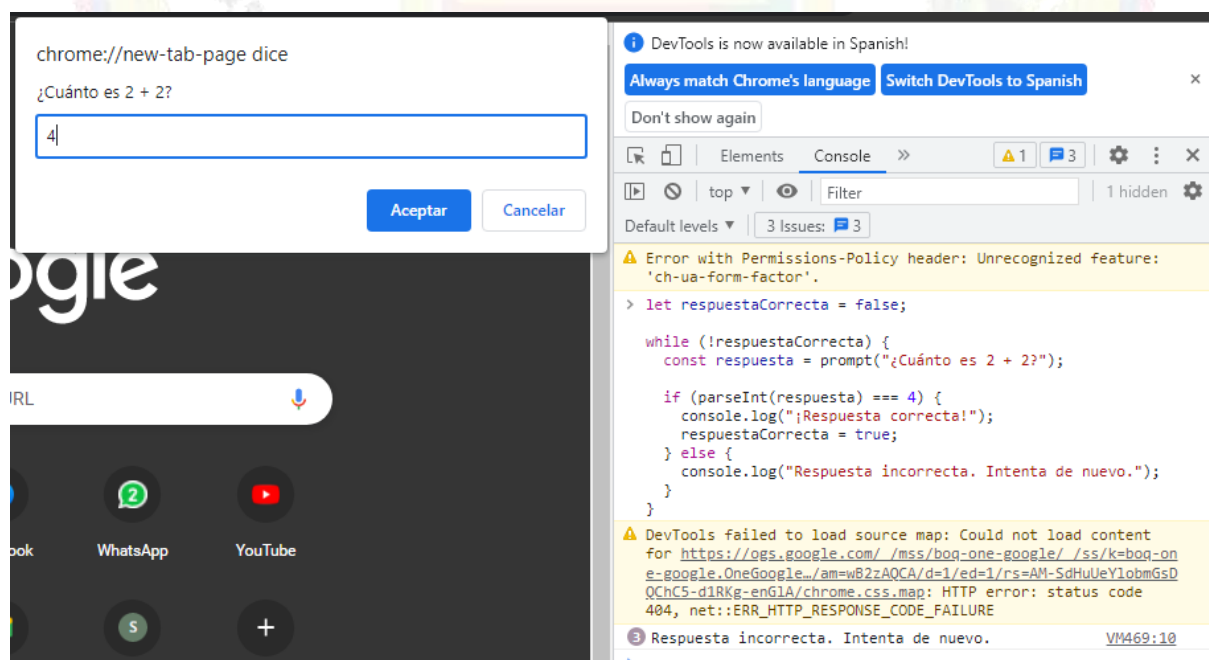
Respuesta incorrecta. Intenta de nuevo. VM423:10

Error with Permissions-Policy header: Unrecognized feature: 'ch-ua-form-factor'.

Respuesta incorrecta. Intenta de nuevo. VM423:10

DevTools failed to load source map: Could not load content for https://ogs.google.com/_/mss/boq-one-google/_/ss/k=boq-one-google.OneGoogle.../am=wB2zAQCA/d=1/ed=1/rs=AM-SdHuUeY1obmGsDQChC5-d1Rkg-enG1A/chrome.css.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE

Respuesta incorrecta. Intenta de nuevo. VM423:10



chrome://new-tab-page dice

¿Cuánto es 2 + 2?

4

Aceptar Cancelar

```

let respuestaCorrecta = false;
while (!respuestaCorrecta) {
  const respuesta = prompt("¿Cuánto es 2 + 2?");

  if (parseInt(respuesta) === 4) {
    console.log("¡Respuesta correcta!");
    respuestaCorrecta = true;
  } else {
    console.log("Respuesta incorrecta. Intenta de nuevo.");
  }
}

```

Error with Permissions-Policy header: Unrecognized feature: 'ch-ua-form-factor'.

DevTools failed to load source map: Could not load content for https://ogs.google.com/_/mss/boq-one-google/_/ss/k=boq-one-google.OneGoogle.../am=wB2zAQCA/d=1/ed=1/rs=AM-SdHuUeY1obmGsDQChC5-d1Rkg-enG1A/chrome.css.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE

Respuesta incorrecta. Intenta de nuevo. VM469:10

⚠ Error with Permissions-Policy header: Unrecognized feature: 'ch-ua-form-factor'.

```
> let respuestaCorrecta = false;

while (!respuestaCorrecta) {
  const respuesta = prompt("¿Cuánto es 2 + 2?");

  if (parseInt(respuesta) === 4) {
    console.log("¡Respuesta correcta!");
    respuestaCorrecta = true;
  } else {
    console.log("Respuesta incorrecta. Intenta de nuevo.");
  }
}
```

⚠ DevTools failed to load source map: Could not load content for https://ogs.google.com/_mss/boq-one-google/_ss/k=boq-one-google.OneGoogle.../am=wB2zAQCA/d=1/ed=1/rs=AM-SdHuUeYlobmGsDQChC5-d1RKg-enG1A/chrome.css.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE

3 Respuesta incorrecta. Intenta de nuevo. [VM469:10](#)

¡Respuesta correcta! [VM469:7](#)

< true

>

1.6 Listas

1.6.1 Responde las siguientes preguntas en la sección de comentarios:

- ¿Qué es un array?

Un array es una estructura de datos que permite almacenar varios valores en una sola variable.

- ¿Qué es un objeto?

Un objeto es una estructura de datos que permite almacenar múltiples propiedades y sus valores asociados.

- ¿Cuándo es mejor usar objetos o arrays?

Los arrays son adecuados cuando se necesita una colección ordenada de elementos accesibles por índices. Los objetos son adecuados cuando se necesita almacenar datos con etiquetas o nombres asociados a sus valores.

- ¿Puedo mezclar arrays con objetos o incluso objetos con arrays?

Sí, se pueden mezclar arrays con objetos y viceversa. Por ejemplo, se puede tener un array que contenga objetos como elementos, o un objeto que tenga propiedades que sean arrays.

1.6.2 Crea un array con los siguientes elementos: "Hola", "¿", "Cómo", "estás", "?".


```
> let listaPalabras = ["Hola", "¿", "Cómo", "estás", "?"];  
< undefined
```

1.6.3 Accede al primer elemento del array que creaste en el ejercicio anterior e imprímelo en la consola.

```
let listaPalabras = ["Hola", "¿", "Cómo", "estás", "?"];  
console.log(listaPalabras[0]);
```

Hola

VM55

1.6.4 Ahora supongamos que debes guardar datos de una persona. Crea un objeto con las siguientes propiedades: nombre, apellido, edad, hobbies (un array con varios elementos) y contacto (un objeto con las propiedades email y teléfono).

```
> let persona = {  
  nombre: "Emilio",  
  apellido: "Niacato",  
  edad: 21,  
  hobbies: ["fútbol", "música", "lectura"],  
  contacto: {  
    email: "ejnacato@espe.edu.ec",  
    telefono: "0962965756"  
  }  
};  
< undefined
```

1.6.5 Accede al valor de la propiedad hobbies y imprímelo en la consola.

```
> let persona = {  
  nombre: "Emilio",  
  apellido: "Niacato",  
  edad: 21,  
  hobbies: ["fútbol", "música", "lectura"],  
  contacto: {  
    email: "ejnacato@espe.edu.ec",  
    telefono: "0962965756"  
  }  
};  
console.log(persona.hobbies);
```

► (3) ['fútbol', 'música', 'lectura']

1.6.6 Accede al valor de la propiedad email y imprímelo en la consola.

```
> let persona = {  
  nombre: "Emilio",  
  apellido: "Niacato",  
  edad: 21,  
  hobbies: ["fútbol", "música", "lectura"],  
  contacto: {  
    email: "ejnacato@espe.edu.ec",  
    telefono: "0962965756"  
  }  
};  
console.log(persona.contacto.email);  
ejnacato@espe.edu.ec
```