

# ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

## INTRODUCCIÓN AL DESARROLLO WEB

### 2025 II

## LABORATORIO 11 – JAVASCRIPT: FUNDAMENTOS

I

#### OBJETIVOS

- Comprender la importancia de JavaScript en el desarrollo web
- Comprender los fundamentos de JavaScript sobre variables, constantes, expresiones, sentencias, estructuras condicionales y repetitivas
- Solucionar ejercicios aplicando los conocimientos obtenidos

TIEMPO ESTIMADO: 2 horas

II

#### CONSIDERACIONES DE EVALUACIÓN

- Se deberán utilizar los conocimientos impartidos en las clases teóricas
- Deberá utilizar nombre de variables significativos
- Deberá realizar pruebas adicionales
- El alumno deberá indicar en su código con quien colaboró, así sea la IA
- El alumno será requerido de realizar modificaciones en su código y responder a preguntas sobre el mismo
- Los ejercicios deberán realizarse en el laboratorio, a su ritmo y subirlos al aula virtual como AVANCE antes de finalizar la clase
- Todos los ejercicios completos, deberán ser subidos al aula virtual como TAREA, para lo cual se les dará un tiempo adecuado y deben cumplir el deadline estipulado. Esto se deberá cumplir, aunque en el laboratorio ya se hayan terminado todos los ejercicios
- El formato a usar para los avances y tareas será .zip, que contenga todos los archivos requeridos
- Utilizar Git con GitHub para el manejo de versiones de su trabajo, ocasionalmente se le solicitará que compartan sus repositorios

III

#### POLITICA DE COLABORACION

La política del curso es simple, a menos que se exprese lo contrario en el laboratorio, siéntase libre de colaborar con sus compañeros en todos los laboratorios, pero debe notificar expresamente con quien ha colaborado. La colaboración con alumnos, que no están matriculados en el curso está prohibida. Los laboratorios y asignaciones han sido desarrollados para ayudarlo a comprender el material. Conozca su código y esté preparado para revisiones individuales de código. Durante las revisiones es probable que se le pida realizar modificaciones y justificar sus decisiones de programación. Cada uno de sus ejercicios debe iniciar de la siguiente forma:

```
// Laboratorio Nro x - Ejerciciox
// Autor: mi nombre
// Colaboró : el nombre
// Tiempo :
```

- En cada sesión de laboratorio, los ejercicios propuestos deberán ser guardados en la misma carpeta/directorio
- La carpeta deberá tener el nombre del Laboratorio y el nombre del alumno, así por ejemplo:  
Laboratorio 11 – Juan Perez
- Utilice nombres significativos
- Su código deberá estar correctamente indentado y preferentemente documentado
- Deberá ser debidamente probado

## 1. Esquema de trabajo

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Pruebas JS</h1>
  <script src="script.js"></script>
</body>
</html>
```

script.js

```
// Salida en la consola
console.log("Hola amigo, desde console.log()");
```

## 2. Variables y Constantes

Variable (let o var): almacena datos que pueden cambiar durante la ejecución.

Constante (const): almacena datos que no cambian.

```
let nombre = "Marco";
const PI = 3.1416;
```

Buenas prácticas:

- Usa let en lugar de var
- Usa const si el valor no cambiará
- Nombres descriptivos usando convenciones (ej: precioTotal, contador, edad, IMPUESTO\_VENTA)

## 3. Entrada y Salida de Datos

**Entrada:**

En navegador

```
let nombre = prompt("Ingresa tu nombre:");
```

**Salida:**

En consola y en navegador

```
console.log("Hola, " + nombre);
alert(`Hola, ${nombre}`);
```

## 4. Estructuras Condicionales

### if / else:

```
if (edad >= 18) {  
  console.log("Eres mayor de edad");  
} else {  
  console.log("Eres menor de edad");  
}
```

### If / else if:

```
if (nota >= 17) console.log("Excelente");  
else if (nota >= 14) console.log("Bueno");  
else console.log("Desaprobado");
```

### Operador ternario:

```
let resultado = edad >= 18 ? "Adulto" : "Menor";
```

### switch:

```
switch (dia) {  
  case 1: console.log("Lunes");  
  break;  
  case 2: console.log("Martes");  
  break;  
  default: console.log("Otro día");  
}
```

## 5. Estructuras Repetitivas

### for:

```
for (let i = 0; i < 5; i++) {  
  console.log("Iteración", i);  
}
```

### while:

```
let i = 0;  
while (i < 5) {  
  console.log(i);  
  i++;  
}
```

### do...while:

```
let num;  
do {  
  num = parseInt(prompt("Ingresa un número positivo:"));  
} while (num <= 0);
```

## 6. Estructuras Compuestas

Combinan condiciones y bucles, o varios tipos de estructuras para resolver problemas más complejos (validaciones, cálculos, contadores, etc.). Indentar.

### Ejemplo:

```
let suma = 0;  
for (let i = 1; i <= 10; i++) {  
  if (i % 2 === 0)  
    suma += i;  
}  
console.log("Suma de pares:", suma);
```

1. Crear un directorio que tenga su nombre y un subdirectorio laboratorio08. Recomendación: usar minúsculas, sin espacios, sin tildes ni "ñ" y con guiones medios o bajos
2. Utilizar los atajos de teclado o combinación de teclas para agilizar su trabajo
3. Calcular promedio con validación. Pide 5 notas y calcula el promedio, asegurándote de que estén entre 0 y 20
4. Solicita edad y muestra:  
"Niño" si  $< 12$   
"Adolescente" si entre 12 y 17  
"Adulto" si entre 18 y 59  
"Adulto mayor" si  $\geq 60$
5. Contador de números pares e impares. Pide 10 números y muestra cuántos fueron pares y cuántos impares
6. Tabla de multiplicar. El usuario ingresa un número y se muestra su tabla del 1 al 12. Programa iterativo
7. Cajero automático. El usuario ingresa un monto a retirar. El programa debe indicar cuántos billetes de 100, 50, 20 y 10 se entregan. (Debe ser la menor cantidad de billetes). Programa iterativo
8. Números primos en rango. Pide un número y muestra todos los primos entre 1 y dicho número. Programa iterativo
9. Suma todos los números desde 1 hasta N, pero omite los múltiplos de 5. Programa iterativo
10. Crea un menú con opciones:  
    Calcular área del círculo  
    Calcular área del rectángulo  
    Salir  
El programa debe repetir hasta que el usuario elija salir.
11. Verificación de número Armstrong. Programa iterativo  
Un número de n dígitos es Armstrong si la suma de sus dígitos elevados a n da el mismo número  
Ejemplo:  $153 \rightarrow 1^3 + 5^3 + 3^3 = 153$ .
12. Menú con múltiples opciones. Crea un menú tipo consola con las siguientes opciones:  
    Calcular estadísticas de N notas  
        Promedio  
        Valor máximo  
        Valor mínimo  
        Cantidad de pares e impares  
        Cuántos están por encima del promedio  
    Contar números pares e impares en un rango  
    Generar una tabla de multiplicar  
    Salir  
Debe repetirse hasta que el usuario elija la opción "Salir".
13. Un supermercado desea una app que simule un carrito de compras muy simple  
El usuario puede agregar productos uno por uno, indicando sólo su precio, hasta que decida no continuar  
Al terminar, mostrar:  
    El total parcial a pagar  
    Si el total supera 100, mostrar el mensaje "Tiene un descuento del 10%" y aplicar el descuento  
    Si el total está entre 50 y 100, mostrar "Gana un cupón de 5%" y aplicar el descuento  
    Si el total es menor que 50, mostrar "No aplica descuento"  
    Finalmente, mostrar el total final a pagar

14. Mismo ejercicio anterior, pero con un límite en su presupuesto. La app no deberá permitir que se pase del presupuesto y no deberá considerar el último producto
15. Crear un repositorio remoto en GitHub y subir tu repositorio local. Compartir URL y pdf con captura de pantalla del código de los archivos js y de la ejecución

#### I. TAREA PARA LA CASA: Complete todos los ejercicios.

Crear un documento docx/pdf con la solución de los ejercicios.

Subir el documento a la tarea **Tarea 11** del Aula Virtual respetando las fechas indicadas.