

# Informe de Laboratorio 3

## Tema: JAVASCRIPT

**Nota**

Estudiante	Escuela	Asignatura
Sergio Hanco Mullisaca shanccom@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: II Código:

Laboratorio	Tema	Duración
3	JAVASCRIPT	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 8 Mayo 2024	Al 15 Mayo 2024

### 1. Tarea

- Informe de laboratorio
- Video en Flip
- Ejercicios en W3Schools
- Ejercicios Propuestos

### 2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux.
- VS
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

### 3. URL de Repositorio Github

- URL del video.
- <https://youtu.be/1YSNDHUQqAM>
- URL del GITHUB.
- [https://github.com/shanccom/Programacion\\_Web\\_2.git](https://github.com/shanccom/Programacion_Web_2.git)

### 4. Actividades

#### 4.1. Actividad 1

- Escriba una función que reciba el número de día de la fecha actual `new Date()` y devuelva el texto del día de la semana correspondientes. Por ejemplo si recibe 0, devolvería “Domingo”.

Listing 1: EJERCICIO\_01.js

```
function main (){
    const fechaA = new Date();
    const numeroDia = fechaA.getDay();
    const nombreDia = obtenerDia(numeroDia);
    console.log("Hoy es: " + nombreDia)
}

main();

function obtenerDia(numeroDia){
    const diaSemana = ["Domingo", "Lunes", "Martes", "Miercoles", "Jueves",
        "Viernes", "Sabado", "Domingo"];

    if (numeroDia >= 0 && numeroDia <= 6){
        return diaSemana[numeroDia];
    } else {
        const diaExtra = (numeroDia) % 7;
        return diaSemana[diaExtra];
    }
}
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
C:\Program Files\nodejs\node.exe .\EJERCICIO_01.js
Hoy es: Sabado
```

- Escriba una página web que reciba un texto y al presionar un botón muestre el mismo texto invertido en otra sección (div). Por ejemplo si se escribe “Hola”, se mostraría como “aloH”.

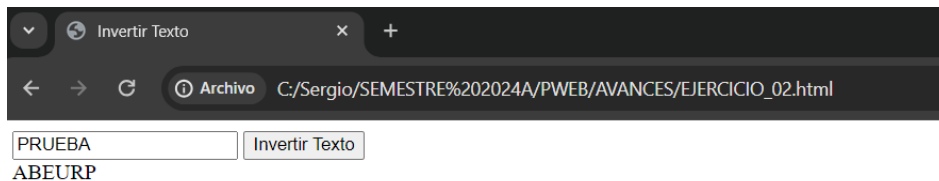
Listing 2: EJERCICIO\_02.js

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Invertir Texto</title>
</head>
<body>
<div>
  <input type="text" id="texto" placeholder="Escribe un texto">
  <button onclick="invertirTexto()">Invertir Texto</button>
  <div id="resultado"></div>
</div>

<script>

function invertirTexto() {
  const textoOriginal = document.getElementById("texto").value;
  let textoInvertido = "";
  for (let i = textoOriginal.length - 1; i >= 0; i--) {
    textoInvertido += textoOriginal[i];
  }
  document.getElementById("resultado").innerText = textoInvertido;
}

</script>
</body>
</html>
```



- Escribir una página que muestre cuántos días faltan para el día de Arequipa!

Listing 3: EJERCICIO\_03.js

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Das Faltantes para el Da de Arequipa</title>
</head>
<body>
<div>
  <h1>Das Faltantes para el Da de Arequipa</h1>
  <p id="diasFaltantes"></p>
</div>
```

```
<script>
function calcularDiasFaltantes() {
    const fechaActual = new Date();
    const diaArequipa = new Date(fechaActual.getFullYear(), 7, 15);

    if (fechaActual > diaArequipa) {
        diaArequipa.setFullYear(diaArequipa.getFullYear() + 1);
    }

    const diferencia = diaArequipa - fechaActual;
    //Transformacion a Dias
    const diasFaltantes = Math.ceil(diferencia / (1000 * 60 * 60 * 24));

    document.getElementById("diasFaltantes").innerText = 'Faltan ${diasFaltantes} das
        para el Da de Arequipa';
}

calcularDiasFaltantes();
</script>
</body>
</html>
```

← → ↻ ⓘ Archivo C:/Sergio/SEMESTRE%202024A/PWEB/AVANCES/EJERCICIO\_03.html

## Días Faltantes para el Día de Arequipa

Faltan 96 días para el Día de Arequipa

- Escribir un página que reciba el URL de la sesión de google meet de hoy y devuelva el código de la sesión sin guiones separadores

Listing 4: EJERCICIO\_04.js

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Extraer Cdigo de Sesin de Google Meet</title>
</head>
<body>
<div>
    <h1>Extraer Cdigo de Sesin de Google Meet</h1>
    <label for="url">Introduce el URL de la sesin de Google Meet:</label>
    <br> <input type="text" id="url"> <br>
    <button onclick="obtenerCodigo()">Obtener Cdigo</button><br>
    <p id="codigo"></p>
```

```
</div>

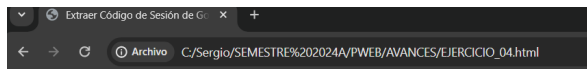
<script>
function obtenerCodigo() {

    const url = document.getElementById("url").value.trim();
    const regex = /https:\/\/meet\.google\.com\/([a-zA-Z0-9_-]+)/;
    const match = url.match(regex);

    if (match) {

        const codigo = match[1];
        const codigoSinGuiones = codigo.replace(/-/g, '');
        document.getElementById("codigo").innerText = 'Código de la sesión: ' +
            `${codigoSinGuiones}`;

    } else {
        document.getElementById("codigo").innerText = "URL no válido. Por favor, introduce un URL de Google Meet válido.";
    }
}
</script>
</body>
</html>
```



### Extraer Código de Sesión de Google Meet

Introduce el URL de la sesión de Google Meet:

- Escribir una página que permita calcular la suma de todos los valores de una tabla de valores dinámica. La idea es crear una página web con un formulario que te permita decir cuantos valores tendrá la tabla, luego, al enviar el formulario la tabla se debe crear dinámica y aleatoriamente, junto con otro botón de envío para calcular la suma.

Listing 5: EJERCICIO\_05.js

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Suma de Valores en una Tabla Dinámica</title>

<style>
    table {
        border-collapse: collapse;
        width: 50%;
        margin-bottom: 20px;
    }
```

```
    th, td {
        border: 1px solid red;
        padding: 8px;
        text-align: center;
    }
</style>

</head>
<body>
<div>
    <h1>Suma de Valores en una Tabla Dinamica</h1>
    <form id="formulario">
        <label for="numValores">Cantidad de Valores:</label>
        <input id="numValores" name="numValores" required min="1">
        <button type="submit">Crear Tabla</button>
    </form>
    <div id="tablaContainer"></div>
    <button id="calcularSuma" style="display:none;">Calcular Suma</button>
    <p id="resultado"></p>
</div>

<script>

document.getElementById("formulario").addEventListener("submit", function(event) {
    event.preventDefault();
    const numValores = parseInt(document.getElementById("numValores").value);
    let tablaHtml = "<table><tr><th>Valores</th></tr>";

    for (let i = 0; i < numValores; i++) {
        const valor = Math.floor(Math.random() * 100) + 1;
        tablaHtml += '<tr><td>${valor}</td></tr>';
    }

    tablaHtml += "</table>";
    document.getElementById("tablaContainer").innerHTML = tablaHtml;
    document.getElementById("calcularSuma").style.display = "block";

});

document.getElementById("calcularSuma").addEventListener("click", function() {
    const valores = document.querySelectorAll("#tablaContainer table td");
    let suma = 0;
    valores.forEach(function(valor) {
        suma += parseInt(valor.textContent);
    });
    document.getElementById("resultado").innerText = 'La suma de los valores es:
        ${suma}';
});

</script>

</body>
</html>
```

← → ↻ Archivo C:/Sergio/SEMESTRE%202024A/PWEB/AVANCES/EJERCICIO\_05.html

## Suma de Valores en una Tabla Dinámica

Cantidad de Valores:

Valores
41
72
45

- Pagina1.html - Cree una página web con un texto y dos botones (al estilo del ejemplo del foco que se enciende y apaga) que permitan cambiar el tamaño de la letra de un texto, intente hacerlo también con los colores.

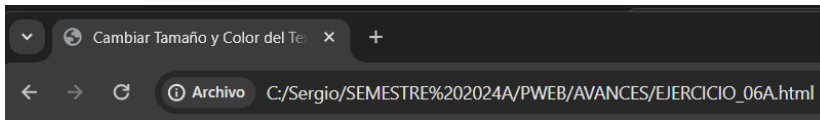
Listing 6: EJERCICIO\_06A.js

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Cambiar Tamao y Color del Texto</title>
<style>
  #texto {
    font-size: 16px;
    color: black;
  }
</style>
</head>
<body>
<div>
  <h1>Cambiar Tamao y Color del Texto</h1>
  <p id="texto">Este es un texto de ejemplo.</p>
  <button onclick="cambiarTamao(1.1)">Aumentar Tamao</button>
  <button onclick="cambiarTamao(0.9)">Reducir Tamao</button>
  <br>
  <button onclick="cambiarColor('red')">Rojo</button>
  <button onclick="cambiarColor('green')">Verde</button>
  <button onclick="cambiarColor('blue')">Azul</button>
  <button onclick="cambiarColor('black')">Negro</button>
</div>

<script>
function cambiarTamao(factor) {
  const texto = document.getElementById("texto");
  const fontSizeActual = window.getComputedStyle(texto).fontSize;
  const nuevoTamao = parseFloat(fontSizeActual) * factor;
  texto.style.fontSize = nuevoTamao + "px";
}

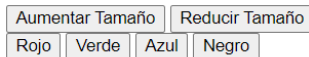
function cambiarColor(color) {
```

```
const texto = document.getElementById("texto");
texto.style.color = color;
}
</script>
</body>
</html>
```



## Cambiar Tamaño y Color del Texto

Este es un texto de ejemplo.



- Pagina2.html - Cree una página web que permita realizar las operaciones aritmética, lógicas y de bits básicas, de manera dinámica( se podrá elegir cualquier operador) y se trabajará con dos argumentos.

Listing 7: EJERCICIO\_06B.js

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculadora</title>
  <style>
    .calculator {
      max-width: 400px;
      margin: 0 auto;
      text-align: center;
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }

    .calculator input, .calculator select, .calculator button {
      margin-bottom: 10px;
      padding: 5px;
    }

    .calculator button {
      background-color: #007bff;
      color: #fff;
      border: none;
      border-radius: 3px;
      cursor: pointer;
    }

    .calculator button:hover {
      background-color: #0056b3;
```



```
    }

    </style>
</head>
<body>
    <h1>Calculadora</h1>
    <div class="calculator">
        <input id="arg1">
        <select id="operator">
            <option value="+">Suma</option>
            <option value="-">Resta</option>
            <option value="*">Multiplicación</option>
            <option value="/">División</option>
        </select>
        <input id="arg2">
        <button onclick="calculate()">Calcular</button>
        <p id="result"></p>
    </div>

    <script>

    function calculate() {
        var arg1 = parseInt(document.getElementById("arg1").value);
        var arg2 = parseInt(document.getElementById("arg2").value);
        var operator = document.getElementById("operator").value;
        var result;

        switch(operator) {
            case "+":
                result = arg1 + arg2;
                break;
            case "-":
                result = arg1 - arg2;
                break;
            case "*":
                result = arg1 * arg2;
                break;
            case "/":
                result = arg1 / arg2;
                break;
            case "&":
                result = arg1 & arg2;
                break;
            case "|":
                result = arg1 | arg2;
                break;
            case "^":
                result = arg1 ^ arg2;
                break;
            default:
                result = "Operador no válido";
        }

        document.getElementById("result").innerText = "Resultado: " + result;
    }
}
```

```
</script>
</body>
</html>
```

← → ↻ Archivo C:/Sergio/SEMESTRE%202024A/PWEB/AVANCES/EJERCICIO\_06B.html


## Calculadora

Suma

Calcular

Resultado: 7

- Resolver los 67 ejercicios de JavaScript en w3schools.com y subir un pantallazo con su nombre y apellido.

 **BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$** [Start today](#)

### Exercise:

On one single line, declare three variables with the following names and values:

```
firstName = "John"
lastName = "Doe"
age = 35
```

## Correct!

[Next >](#)

[Next Exercise >](#)

Completed 5 of 67 Exercises:

JS Variables ✓

Exercise 1 ✓

Exercise 2 ✓

Exercise 3 ✓

Exercise 4 ✓

Exercise 5 ✓




[Go to JS Variables Tutorial](#)


JS Operators

JS Data Types

JS Functions

JS Objects



Completed 10 of 67 Exercises:

JS Variables ✓

JS Operators ✓

✓ Exercise 1

✓ Exercise 2

✓ Exercise 3

✓ Exercise 4

✓ Exercise 5


[Go to JS Operators Tutorial](#)

JS Data Types


JS Functions

JS Objects

JS Events



**BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$**
[Start today](#)






## Exercise:


Use the correct **assignment operator** that will result in `x` being 50 (same as `x = x * y`).

Correct!

Next >

Next Exercise >



Completed 11 of 67 Exercises:

JS Variables ✓

JS Operators ✓

JS Data Types ✓

✓ Exercise 1

[Go to JS Data Types Tutorial](#)

JS Functions


JS Objects

JS Events


JS Strings

JS String Methods

JS Arrays



**BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$**
[Start today](#)






## Exercise:


Use comments to describe the correct data type of the following variables:

Correct!

Next >


Next Exercise >




Completed 15 of 67 Exercises:

- JS Variables ✓
- JS Operators ✓
- JS Data Types ✓
- JS Functions ✓
- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- ✓ Exercise 4
- [Go to JS Functions Tutorial](#)
- JS Objects
- JS Events
- JS Strings



**BUILD YOUR CAREER. GET  
FULL ACCESS. SAVE 770\$**

Start today






## Exercise:


Make the function display "Hello" in the inner HTML of an element with the ID "demo".

Correct!

[Next >](#)


[Next Exercise >](#)



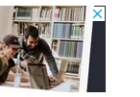
Completed 18 of 67 Exercises:

- JS Variables ✓
- JS Operators ✓
- JS Data Types ✓
- JS Functions ✓
- JS Objects ✓
- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- [Go to JS Objects Tutorial](#)
- JS Events
- JS Strings
- JS String Methods
- JS Arrays



**BUILD YOUR CAREER. GET  
FULL ACCESS. SAVE 770\$**

Start today






## Exercise:


Create an object called person with name = John, age = 50.  
Then, access the object to alert("John is 50").

Correct!

[Next >](#)


[Next Exercise >](#)



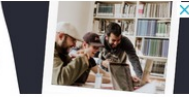
Completed 21 of 67 Exercises:

- JS Variables ✓
- JS Operators ✓
- JS Data Types ✓
- JS Functions ✓
- JS Objects ✓
- JS Events ✓
- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- [Go to JS Events Tutorial](#)
- JS Strings



**BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$**

Start today






## Exercise:


The <div> element should turn red when someone moves the mouse over it.

Correct!

[Next >](#)


[Next Exercise >](#)




Completed 24 of 67 Exercises:

- JS Variables ✓
- JS Operators ✓
- JS Data Types ✓
- JS Functions ✓
- JS Objects ✓
- JS Events ✓
- JS Strings ✓
- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- [Go to JS Strings Tutorial](#)
- JS String Methods



**BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$**

Start today






## Exercise:


Concatenate the two strings to alert "Hello World!".

Correct!

[Next >](#)

[Next Exercise >](#)





Completed 32 of 67 Exercises:

JS Variables ✓

JS Operators ✓

JS Data Types ✓

JS Functions ✓

JS Objects ✓

JS Events ✓

JS Strings ✓

JS String Methods ✓

JS Arrays ✓




✓ Exercise 1

✓ Exercise 2

✓ Exercise 3

[Go to JS Arrays Tutorial](#)

JS Array Methods

**BUILD YOUR CAREER. GET  
FULL ACCESS. SAVE 770\$**




## Exercise:


Alert the number of items in an array, using the correct Array property.

### Correct!

[Next >](#)

[Next Exercise >](#)





Completed 35 of 67 Exercises:

JS Variables ✓

JS Operators ✓

JS Data Types ✓

JS Functions ✓

JS Objects ✓

JS Events ✓

JS Strings ✓

JS String Methods ✓

JS Arrays ✓

JS Array Methods ✓


✓ Exercise 1

✓ Exercise 2

✓ Exercise 3


[Go to JS Array Methods Tutorial](#)

JS Array Sort



**BUILD YOUR CAREER. GET  
FULL ACCESS. SAVE 770\$**

Start today







## Exercise:

Use the `splice()` method to remove "Orange" and "Apple" from `fruits`.

Correct!

[Next >](#)

[Next Exercise >](#)



Completed 36 of 67 Exercises:

JS Variables ✓

JS Operators ✓

JS Data Types ✓

JS Functions ✓

JS Objects ✓

JS Events ✓

JS Strings ✓

JS String Methods ✓

JS Arrays ✓



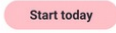
JS Array Methods ✓

JS Array Sort ✓

✓ Exercise 1

[Go to JS Array Sort Tutorial](#)

JS Dates

**BUILD YOUR CAREER. GET  
FULL ACCESS. SAVE 770\$**

## Exercise:




Use the correct Array method to sort the `fruits` array alphabetically.


### Correct!

[Next >](#)

[Next Exercise >](#)







Completed 40 of 67 Exercises:

- JS Objects ✓
- JS Events ✓
- JS Strings ✓
- JS String Methods ✓
- JS Arrays ✓
- JS Array Methods ✓
- JS Array Sort ✓
- JS Dates ✓
- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- ✓ Exercise 4
- [Go to JS Dates Tutorial](#)
- JS Math



**BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$**

Start today






## Exercise:


Use the correct Date method to set the year of a date object to 2020.

Correct!

Next >


Next Exercise >




Completed 44 of 67 Exercises:

- JS Array Sort ✓
- JS Dates ✓
- JS Math ✓
- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- ✓ Exercise 4
- [Go to JS Math Tutorial](#)
- JS Comparisons
- JS Conditions
- JS Switch
- JS For Loops
- JS While



**BUILD YOUR CAREER. GET FULL ACCESS. SAVE 770\$**

Start today






## Exercise:


Use the correct Math method to get the square root of 9.

Correct!

Next >

Next Exercise >



Completed 48 of 67 Exercises:

JS Array Methods ✓

JS Array Sort ✓

JS Dates ✓

JS Math ✓

JS Comparisons ✓

✓ Exercise 1

✓ Exercise 2




✓ Exercise 3

✓ Exercise 4

[Go to JS Comparisons Tutorial](#)

JS Conditions

JS Switch


**BUILD YOUR CAREER. GET  
FULL ACCESS. SAVE 770\$**






## Exercise:


Choose the correct conditional (ternary) operator to alert "Too young" if age is less than 18, otherwise alert "Old enough".

Correct!

Next >

Next Exercise >



Completed 50 of 67 Exercises:

JS Arrays ✓

JS Array Methods ✓

JS Array Sort ✓

JS Dates ✓

JS Math ✓

JS Comparisons ✓

JS Conditions ✓




✓ Exercise 1

✓ Exercise 2

[Go to JS Conditions Tutorial](#)

JS Switch

JS For Loops


**BUILD YOUR CAREER. GET  
FULL ACCESS. SAVE 770\$**






## Exercise:


Fix the if statement to alert "Hello World" if x is greater than y, otherwise alert "Goodbye".

Correct!

Next >

Next Exercise >





Completed 52 of 67 Exercises:

JS Array Sort

✓

JS Dates

✓

JS Math

✓

JS Comparisons

✓

JS Conditions

✓

JS Switch

✓

✓ Exercise 1

✓ Exercise 2

Go to JS Switch Tutorial

JS For Loops

JS While Loops

JS Break Loops

JS HTML DOM




## Exercise:


Add a section that will alert("Neither") if `fruits` is neither "banana" nor "apple".

Correct!

[Next >](#)

[Next Exercise >](#)





Completed 54 of 67 Exercises:

JS Array Sort ✓

JS Dates ✓

JS Math ✓

JS Comparisons ✓

JS Conditions ✓

JS Switch ✓

JS For Loops ✓

✓ Exercise 1

✓ Exercise 2

[Go to JS For Loops Tutorial](#)

JS While Loops

JS Break Loops

JS HTML DOM




## Exercise:


Create a loop that runs through each item in the `fruits` array.

### Correct!

[Next >](#)

Next Exercise >





Completed 56 of 67 Exercises:

JS Array Sort ✓

JS Dates ✓

JS Math ✓

JS Comparisons ✓

JS Conditions ✓

JS Switch ✓

JS For Loops ✓

JS While Loops ✓

✓ Exercise 1

✓ Exercise 2

[Go to JS While Loops Tutorial](#)

JS Break Loops

JS HTML DOM





## Exercise:

Create a loop that runs as long as `i` is less than 10, but increase `i` with 2 each time.

### Correct!

[Next >](#)

Next Exercise >



Completed 58 of 67 Exercises:





- JS Array Sort ✓
- JS Dates ✓
- JS Math ✓
- JS Comparisons ✓
- JS Conditions ✓
- JS Switch ✓
- JS For Loops ✓
- JS While Loops ✓
- JS Break Loops ✓
- ✓ Exercise 1
- ✓ Exercise 2
- [Go to JS Break Loops Tutorial](#)
- JS HTML DOM

## Exercise:

Make the loop jump to the next iteration when `i` is 5.

Correct!  
[Next >](#)

Next Exercise >



Completed 67 of 67 Exercises:

- JS While Loops ✓
- JS Break Loops ✓
- JS HTML DOM ✓
- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- ✓ Exercise 4
- ✓ Exercise 5
- ✓ Exercise 6
- ✓ Exercise 7
- ✓ Exercise 8
- ✓ Exercise 9
- [Go to JS HTML DOM Tutorial](#)

## Exercise:

Use the `addEventListener` to assign an onclick event to the `<button>` element.

Correct!

## 5. Rúbricas

### 5.1. Entregable Informe

Tabla 1: Tipo de Informe

<b>Informe</b>	
<b>Latex</b>	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

### 5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

<b>Puntos</b>	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
<b>2.0</b>	0.5	1.0	1.5	2.0
<b>4.0</b>	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
<b>1. GitHub</b>	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
<b>2. Commits</b>	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
<b>3. Código fuente</b>	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
<b>4. Ejecución</b>	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
<b>5. Pregunta</b>	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
<b>6. Fechas</b>	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
<b>7. Ortografía</b>	El documento no muestra errores ortográficos.	2	X	2	
<b>8. Madurez</b>	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
<b>Total</b>		20		17	