

### Guía de trabajo No. 3

1. Entregar en los grupos de trabajo establecidos la aplicación de IA en la que usted se inscribió en el documento de google drive con su correo electrónico en la guía pasada. Realizar la infografía completa sobre su trabajo con IA explicado paso a paso su trabajo. Los mejores trabajos serán expuestos en la feria de **TECNO-INFORMÁTICA** para el 16 de octubre de 2024.
2. En grupos de 4 estudiantes usted debe presentar un proyecto para ser expuesto en la feria de TECNO\_INFORMÁTICA. La primera revisión de este trabajo se hará en la primera semana de septiembre y la revisión para elegir los proyectos de exposición se hará en la última semana de septiembre. Los trabajos para la feria serán elegidos antes de finalizar el mes de septiembre y servirán como nivelación de este trimestre para los estudiantes que presenten bajo rendimiento durante el mismo.

Cada uno de los problemas planteados deben ser documentados en su cuaderno de trabajo incluyendo la explicación suficiente del paso a paso, el respectivo diagrama de flujo y los procesos y procedimientos que implican dicha solución. Clase por clase se revisará y registrará la solución y entrega de cada problema sin acumulación de trabajos para las últimas semanas de clase del año escolar 2024.

1. Objetivos:
  - a. Aplicar la programación en Java Script, CSS y HTML en el análisis y solución de diversos tipos de problemas.
  - b. Utilizar diferentes fuentes de información para encontrar soluciones adecuadas a cada uno los problemas propuestos.
  - c. Seleccionar y utilizar las herramientas tecnológicas y de programación necesarias para solucionar cada uno de los problemas previstos en la presente guía.
  - d. Dar uso adecuado a los recursos institucionales y propios para alcanzar los aprendizajes propuestos en la programación de computadores.

#### **Actividad Inicial:**

Preguntas para registrar en su cuaderno de trabajo de informática:

- ¿En qué consiste la programación en Java Script?
- ¿Cuáles son las principales características del lenguaje JS?
- ¿Qué aplicaciones se pueden desarrollar en JS?

#### **Actividad No. 1**

Desarrollar cada uno de los siguientes problemas de manera ordenada y completa con el software indicado (Java Script, CSS-HTML). documentar en el cuaderno de manera completa cada ejercicio. Entregar clase a clase. No acumular trabajos ya que máximo se evaluará una actividad por clase.

### **Problema 1: Ahorrador Feliz.**

Qué cantidad de dinero logra ahorrar un cliente de banco y cuánto interés gana este mismo cliente según el siguiente planteamiento:

Un ahorrador feliz cada mes se dirige al banco de su confianza y allí deposita cantidades diferentes de dinero durante un año completo. A la cantidad de dinero ahorrado en cada mes se le debe asignar el interés respectivo que debe pagarle el banco de la siguiente manera: 4.5%, para el segundo mes el 4%, para el tercer mes el 5%, para el cuarto mes el 6%, para el quinto mes el 8%, para el sexto mes el 7%, para el séptimo mes el 8.5%, para el octavo mes el 7.5%, para el noveno mes el 12%, para el décimo mes el 13,5%, para el onceavo mes el 11.5% y para el mes doce el 10.5%. Par el segundo mes debe tenerse en cuenta lo siguiente: el ahorrador ingresa la cantidad que va a ahorrar y a esa cantidad le suma el valor del mes anterior más los intereses causados en el primer mes y así sucesivamente para todo el periodo que dura el ahorro de este cliente bancario. Esta condición se repite durante los siguientes meses de tal manera que al final del periodo de ahorro anual se deben mostrar los resultados totales y los resultados parciales del ahorro que ha tenido este cliente.

### **Problema 2: Crear una calculadora**

Crear una calculadora que resuelva las 4 operaciones básicas, la potenciación, la conversión de números binarios a decimal, la conversión de números decimales a binarios.

### **Problema 3: Maquina Dispensadora.**

Una maquina dispensadora de alimentos, tiene disponible para sus clientes los siguientes productos y precios: Pasabocas \$7.250 pesos; Refresco \$6.200 pesos; Paquete \$8.500 pesos; Bebida Caliente \$8.000 pesos.

Realizar un programa que genere la factura completa para la compra de productos que realice un cliente.

### **Problema 4: Juego interactivo**

Diseñar un juego interactivo completo usando las herramientas necesarias de JS, CSS y HTML.

### **Problema 5: Circuito paralelo con 3 elementos y fuente.**

Utilizar el mismo esquema del punto 2G de la guía anterior. Crear un programa que resuelva el siguiente problema: Un circuito mixto consta de 3 resistencias cuyos valores oscilan entre 1 y 50  $\Omega$  y una fuente de voltaje en voltios entre 1 y 50 voltios. Crear un programa que calcule la resistencia total, la corriente total, el voltaje en cada resistencia, la potencia en cada resistencia, la potencia total. A parte de los resultados se deben mostrar las unidades de medida de cada cantidad eléctrica.

### **Problema 6: Funciones trigonométricas básicas**

Crear un programa JS... para dibujar las tres funciones trigonométricas básicas (Seno, Coseno, Tangente).

### **Problema 7:**

En una caja de pimpones se tiene una cantidad exacta de los mismos y según los siguientes colores: Rojo, Morado, Amarillo y Blanco. Crear un programa que al ingresar la cantidad de pimpones que tiene la caja en su interior los clasifique de manera automática a partir de suministrar el total de los mismos y mostrar la cantidad de cada uno de ellos.

### **Problema 8: Clasificación fecha por Estación**

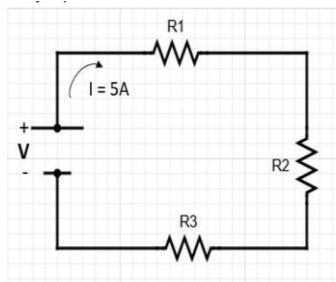
Crear un programa que al ingresar una fecha (día y mes) muestre a que estación del año pertenece, teniendo en cuenta la siguiente información: Verano:1 mayo-31\_Julio, Primavera:16 enero-30 abril, Otoño: 1 agosto-30 septiembre, Invierno: 1 octubre-15 enero.

**Problema 9:** Al ingresar un valor cualquiera, obtener los primeros 20 valores de la tabla de multiplicar para este número que se ha ingresado.

### **Ejemplo de programación No. 1**

#### **CIRCUITO EN SERIE**

Un circuito eléctrico en serie consta de tres resistencias R1, R2 y R3 y una fuente de Voltaje V, como lo muestra la siguiente figura:



Analizar, solucionar y aplicar los procedimientos y operaciones completas y adecuados que permitan solucionar el siguiente problema: solicita al usuario ingresar los valores de cada una de las resistencias cuyos valores están limitados entre 1 y 50 ohmios y para la fuente de voltaje entre 1 y 50 voltios. A partir de los datos ingresados por teclado o digitalmente, el programa debe calcular el valor de la resistencia total, la corriente total, V1, V2 y V3, así como, la potencia P1, P2, P3 y PT, indicando las respectivas unidades o cantidades eléctricas estipuladas. Tener en cuenta las condiciones que limitan el problema si se requiere: calcular la resistencia total del circuito, la corriente total del circuito, el valor del voltaje en cada

una de las resistencias del circuito, la potencia en cada resistencia, la potencia total del circuito.

Abrir un archivo nuevo en Visual Studio Code y crear una carpeta de trabajo con el nombre: Circuito eléctrico.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cálculo de Circuito en Serie</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Cálculo de Circuito en Serie</h1>
    <form id="circuitForm">
      <label for="R1">Resistencia R1 ( $\Omega$ ):</label>
      <input type="number" id="R1" name="R1" min="1" max="50" required>
      <label for="R2">Resistencia R2 ( $\Omega$ ):</label>
      <input type="number" id="R2" name="R2" min="1" max="50" required>
      <label for="R3">Resistencia R3 ( $\Omega$ ):</label>
      <input type="number" id="R3" name="R3" min="1" max="50" required>
      <label for="V">Fuente de Voltaje V (V):</label>
      <input type="number" id="V" name="V" min="1" max="50" required>
      <button type="submit">Calcular</button>
    </form>
    <div id="results">
      <table>
        <thead>
```

```
        <tr>
            <th>Parámetro</th>
            <th>Valor</th>
            <th>Unidad</th>
        </tr>
    </thead>
    <tbody id="resultsBody">
        <!-- Resultados se mostrarán aquí -->
    </tbody>
</table>
</div>
</div>
<script src="script.js"></script>
</body>
</html>
```

Abrir un Nuevo archivo y dentro de la carpeta respectiva asignar el nombre:

Styles.css y digitar el siguiente código:

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}
```

```
.container {  
  background-color: white;  
  padding: 20px;  
  border-radius: 10px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
  width: 300px;  
}
```

```
h1 {  
  text-align: center;  
}
```

```
form {  
  display: flex;  
  flex-direction: column;  
}
```

```
label {  
  margin-top: 10px;  
}
```

```
input {  
  margin-bottom: 10px;  
  padding: 5px;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
}
```

```
button {  
  padding: 10px;  
  background-color: #007bff;  
  color: white;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

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

```
table {  
  width: 100%;  
  margin-top: 20px;  
  border-collapse: collapse;  
}
```

```
th, td {  
  border: 1px solid #ccc;  
  padding: 10px;  
  text-align: left;  
}
```

Ahora se va a crear un nuevo archivo dentro de la carpeta inicial llamada circuito serie con el nombre styles.js y copiar el siguiente código:

```
document.getElementById('circuitForm').addEventListener('submit', function(event)  
{
```

```
event.preventDefault();
```

```
let R1 = parseFloat(document.getElementById('R1').value);
```

```
let R2 = parseFloat(document.getElementById('R2').value);
```

```
let R3 = parseFloat(document.getElementById('R3').value);
```

```
let V = parseFloat(document.getElementById('V').value);
```

```
let RT = R1 + R2 + R3;
```

```
let IT = V / RT;
```

```
let VR1 = IT * R1;
```

```
let VR2 = IT * R2;
```

```
let VR3 = IT * R3;
```

```
let PR1 = IT * VR1;
```

```
let PR2 = IT * VR2;
```

```
let PR3 = IT * VR3;
```

```
let PT = PR1 + PR2 + PR3;
```

```
let resultsBody = document.getElementById('resultsBody');
```

```
resultsBody.innerHTML = `
```

```
  <tr>
```

```
    <td>Resistencia Total</td>
```

```
    <td>${RT.toFixed(2)}</td>
```

```
    <td> $\Omega$ </td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Corriente Total</td>
```



```
<td>${IT.toFixed(2)}</td>
<td>A</td>
</tr>
<tr>
<td>Voltaje en R1</td>
<td>${VR1.toFixed(2)}</td>
<td>V</td>
</tr>
<tr>
<td>Voltaje en R2</td>
<td>${VR2.toFixed(2)}</td>
<td>V</td>
</tr>
<tr>
<td>Voltaje en R3</td>
<td>${VR3.toFixed(2)}</td>
<td>V</td>
</tr>
<tr>
<td>Potencia en R1</td>
<td>${PR1.toFixed(2)}</td>
<td>W</td>
</tr>
<tr>
<td>Potencia en R2</td>
<td>${PR2.toFixed(2)}</td>
<td>W</td>
</tr>
```

```
<tr>
  <td>Potencia en R3</td>
  <td>${PR3.toFixed(2)}</td>
  <td>W</td>
</tr>
<tr>
  <td>Potencia Total</td>
  <td>${PT.toFixed(2)}</td>
  <td>W</td>
</tr>
`
});
```

El resultado que usted debe obtener al final cuando el código completo funcione es el siguiente:

# Cálculo de Circuito en Serie

Resistencia R1 ( $\Omega$ ):

Resistencia R2 ( $\Omega$ ):

Resistencia R3 ( $\Omega$ ):

Fuente de Voltaje V (V):

Calcular

Parámetro	Valor	Unidad
-----------	-------	--------

# Cálculo de Circuito en Serie

Resistencia R1 ( $\Omega$ ):

Resistencia R2 ( $\Omega$ ):

Resistencia R3 ( $\Omega$ ):

Fuente de Voltaje V (V):

Calcular

Parámetro	Valor	Unidad
-----------	-------	--------

Actividad No. 2.

Obtener los primeros 15 resultados para una tabla de multiplicar que se ingresa por teclado y organizar los mismos en una tabla organizada con operaciones y resultados obtenidos.

Resistencia R2 ( $\Omega$ ):

Resistencia R3 ( $\Omega$ ):

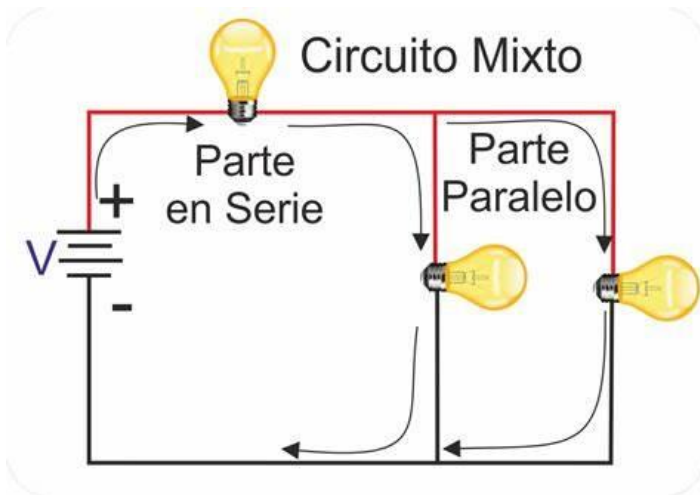
Fuente de Voltaje V (V):

Calcular

Parámetro	Valor	Unidad
Resistencia Total	10.00	$\Omega$
Corriente Total	2.00	A
Voltaje en R1	4.00	V
Voltaje en R2	6.00	V
Voltaje en R3	10.00	V
Potencia en R1	8.00	W
Potencia en R2	12.00	W
Potencia en R3	20.00	W
Potencia Total	40.00	W

### **Problema 10**

Utilizar la programación respectiva analizar y solucionar el siguiente circuito eléctrico: Debe obtener todos los valores de voltajes, corrientes, resistencias y potencias. Los valores a utilizar se encuentran en: las resistencias entre 1 y 100 ohmios y el voltaje entre 1 y 20 voltios.



### Ejemplo de programación No. 2:

#### Juego con dados y dos jugadores:

Realizar el siguiente juego y establecer las condiciones y reglas que esta cumpliendo el juego: registrar el enunciado del problema en su cuaderno de trabajo:

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dice Game</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Dice Game</h1>
    <div id="game">
      <div id="dice">
```

```
<div id="player1-dice" class="dice red"></div>
<div id="player2-dice" class="dice blue"></div>
</div>
<div id="scores">
  <table>
    <tr>
      <th>Player 1</th>
      <th>Player 2</th>
    </tr>
    <tr>
      <td id="player1-score">0</td>
      <td id="player2-score">0</td>
    </tr>
  </table>
</div>
<div id="status"></div>
<div id="turn-counter">Turns: 0</div>
</div>
</div>

<script src="script.js"></script>
</body>
</html>
```

### **STYLES.CSS**

```
// Variables to track scores and turns
let player1Score = 0;
let player2Score = 0;
```

```
let currentPlayer = 1;
let turns = 0;

// Function to roll the dice
function rollDice() {
    return Math.floor(Math.random() * 6) + 1; // Returns a random number from 1 to 6
}

// Function to handle a player's turn
function playTurn() {
    let dice1 = rollDice();
    let dice2 = rollDice();

    if (currentPlayer === 1) {
        player1Score += dice1 + dice2;
        document.getElementById('player1-dice').textContent = dice1;
        document.getElementById('player2-dice').textContent = dice2;
        document.getElementById('player1-score').textContent = player1Score;
    } else {
        player2Score += dice1 + dice2;
        document.getElementById('player1-dice').textContent = dice1;
        document.getElementById('player2-dice').textContent = dice2;
        document.getElementById('player2-score').textContent = player2Score;
    }

    turns++;
    document.getElementById('turn-counter').textContent = `Turns: ${turns}`;
```



```
// Check if a player has reached 50 points
if ((currentPlayer === 1 && player1Score >= 50) || (currentPlayer === 2 &&
player2Score >= 50)) {
    document.getElementById('status').textContent = 'You have reached 50 points.
Remove the higher die to continue.';
}
```

```
// Check if a player has reached 56 points
if (player1Score >= 56) {
    endGame(1);
} else if (player2Score >= 56) {
    endGame(2);
} else {
    currentPlayer = currentPlayer === 1 ? 2 : 1; // Switch turn to the other player
}
}
```

```
// Function to end the game
function endGame(winner) {
    document.getElementById('status').textContent = `Player ${winner} wins!`;
    let audio = new Audio('bell.mp3'); // Replace with your audio file
    audio.play();
}
```

```
// Event listener for clicking to play
document.addEventListener('DOMContentLoaded', function() {
    document.getElementById('game').addEventListener('click', playTurn);
});
```

## JAVA SCRIPT:

```
// Variables to track scores and turns
```

```
let player1Score = 0;
```

```
let player2Score = 0;
```

```
let currentPlayer = 1;
```

```
let turns = 0;
```

```
// Function to roll the dice
```

```
function rollDice() {
```

```
    return Math.floor(Math.random() * 6) + 1; // Returns a random number from 1 to 6
```

```
}
```

```
// Function to handle a player's turn
```

```
function playTurn() {
```

```
    let dice1 = rollDice();
```

```
    let dice2 = rollDice();
```

```
    if (currentPlayer === 1) {
```

```
        player1Score += dice1 + dice2;
```

```
        document.getElementById('player1-dice').textContent = dice1;
```

```
        document.getElementById('player2-dice').textContent = dice2;
```

```
        document.getElementById('player1-score').textContent = player1Score;
```

```
    } else {
```

```
        player2Score += dice1 + dice2;
```

```
        document.getElementById('player1-dice').textContent = dice1;
```

```
        document.getElementById('player2-dice').textContent = dice2;
```

```
        document.getElementById('player2-score').textContent = player2Score;
```

```
    }
```

```

turns++;

document.getElementById('turn-counter').textContent = `Turns: ${turns}`;

// Check if a player has reached 50 points
if ((currentPlayer === 1 && player1Score >= 50) || (currentPlayer === 2 &&
player2Score >= 50)) {
    document.getElementById('status').textContent = 'You have reached 50 points.
Remove the higher die to continue.';
}

// Check if a player has reached 56 points
if (player1Score >= 56) {
    endGame(1);
} else if (player2Score >= 56) {
    endGame(2);
} else {
    currentPlayer = currentPlayer === 1 ? 2 : 1; // Switch turn to the other player
}
}

// Function to end the game
function endGame(winner) {
    document.getElementById('status').textContent = `Player ${winner} wins!`;
    let audio = new Audio('bell.mp3'); // Replace with your audio file
    audio.play();
}

// Event listener for clicking to play

```

```
document.addEventListener('DOMContentLoaded', function() {  
  document.getElementById('game').addEventListener('click', playTurn);  
});
```