

```

function calculateCost() {
  try {
    // Obtén los valores de los inputs
    const mes1_rcv =
parseFloat(document.getElementById('mes1_rcv').value);
    const mes1_bancos =
parseFloat(document.getElementById('mes1_bancos').value);
    const mes2_rcv =
parseFloat(document.getElementById('mes2_rcv').value);
    const mes2_bancos =
parseFloat(document.getElementById('mes2_bancos').value);
    const mes3_rcv =
parseFloat(document.getElementById('mes3_rcv').value);
    const mes3_bancos =
parseFloat(document.getElementById('mes3_bancos').value);
    const months =
parseFloat(document.getElementById('months').value);
    const resultDiv = document.getElementById('result');

    // Limpia el resultado previo
    resultDiv.innerHTML = '';

    // Verifica que los valores sean válidos
    if (
      isNaN(mes1_rcv) || isNaN(mes1_bancos) ||
      isNaN(mes2_rcv) || isNaN(mes2_bancos) ||
      isNaN(mes3_rcv) || isNaN(mes3_bancos) ||
      isNaN(months)
    ) {
      throw new Error("Por favor, ingrese valores válidos en
todos los campos.");
    }

    // Calcula el promedio de los valores ingresados
    const promedio = (
      mes1_rcv + mes1_bancos +
      mes2_rcv + mes2_bancos +
      mes3_rcv + mes3_bancos
    ) / 6;

    // Calcula el total con el promedio y los meses
    const total = promedio * months;

    // Tabla de precios
    const priceTable = [
      { min: 0, max: 100, cost: 1.0 },
      { min: 101, max: 200, cost: 2.2 },
      { min: 201, max: 300, cost: 3.0 },
      { min: 301, max: 400, cost: 3.5 },
      { min: 401, max: 500, cost: 4.4 },
    ];
  }
}

```

```

        { min: 501, max: 600, cost: 5.0 },
        { min: 601, max: 700, cost: 5.6 },
        { min: 701, max: 800, cost: 6.1 },
        { min: 801, max: 900, cost: 7.0 },
        { min: 901, max: 1000, cost: 8.0 },
        { min: 1001, max: 2000, cost: 13.0 },
        { min: 2001, max: 3000, cost: 16.0 },
        { min: 3001, max: 4000, cost: 18.3 },
        { min: 4001, max: 5000, cost: 20.3 },
        { min: 5001, max: 10000, cost: 26.0 },
        { min: 10001, max: 20000, cost: 30.0 },
        { min: 20001, max: 30000, cost: 33.0 }
    ];

    // Encuentra el rango correspondiente en la tabla
    const range = priceTable.find(r => total >= r.min && total <=
r.max);

    // Determina el costo basado en el rango
    const cost = range ? range.cost : 0;

    // Muestra el resultado en la página
    resultDiv.innerHTML = `
        Promedio: <strong>${promedio.toFixed(2)}</strong><br>
        Total: <strong>${total.toFixed(2)}</strong><br>
        Costo estimado: <strong>${cost.toFixed(2)} UF</strong>
    `;
    } catch (error) {
        // Muestra mensajes de error al usuario
        resultDiv.innerHTML = `<span style="color: red;">${
error.message}</span>`;
    }
}

```