

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>PROGRAMA 3</title>
7 <!-- Importamos la fuente Poppins de Google Fonts para darle un aspecto moderno -->
8 <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">
9 <style>
10 /* Estilos generales para resetear márgenes y establecer la fuente */
11 * {
12     margin: 0;
13     padding: 0;
14     box-sizing: border-box;
15     font-family: 'Poppins', sans-serif;
16 }
17
18 /* Fondo animado con gradiente que cambia de colores */
19 body {
20     background: linear-gradient(-45deg, #ee7752, #e73c7e, #23a6d5, #23d5ab);
21     background-size: 400% 400%;
22     animation: gradient 15s ease infinite;
23     min-height: 100vh;
24     display: flex;
25     justify-content: center;
26     align-items: center;
27     padding: 20px;
28     color: #333;
29 }
30

```

```

31 /* Animación del gradiente de fondo */
32 @keyframes gradient {
33     0% {
34         background-position: 0% 50%;
35     }
36     50% {
37         background-position: 100% 50%;
38     }
39     100% {
40         background-position: 0% 50%;
41     }
42 }
43
44 /* Contenedor principal con efecto de flotación */
45 .container {
46     background-color: rgba(255, 255, 255, 0.95);
47     border-radius: 15px;
48     box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
49     padding: 30px;
50     width: 100%;
51     max-width: 800px;
52     transform: translateY(0);
53     animation: float 6s ease-in-out infinite;
54 }

```

```
56      /* Animación de flotación para el contenedor */
57      @keyframes float {
58          0% {
59              transform: translateY(0px);
60          }
61          50% {
62              transform: translateY(-10px);
63          }
64          100% {
65              transform: translateY(0px);
66          }
67      }
68
69      /* Estilo para el título principal con línea decorativa */
70      h2 {
71          color: #333;
72          margin-bottom: 25px;
73          text-align: center;
74          font-weight: 600;
75          position: relative;
76          padding-bottom: 10px;
77      }
78
```

```
79      /* Línea decorativa debajo del título */
80      h2:after {
81          content: '';
82          position: absolute;
83          bottom: 0;
84          left: 50%;
85          transform: translateX(-50%);
86          width: 60px;
87          height: 4px;
88          background: linear-gradient(to right, #23a6d5, #23d5ab);
89          border-radius: 2px;
90      }
91
92      /* Contenedor para cada grupo de entrada */
93      .input-group {
94          margin-bottom: 20px;
95          position: relative;
96      }
97
98      /* Estilo para las etiquetas de los campos */
99      label {
100          display: block;
101          margin-bottom: 8px;
102          font-weight: 500;
103          color: #555;
104      }
105
```

```
106      /* Estilo para los campos de entrada */
107      input {
108          width: 100%;
109          padding: 12px 15px;
110          border: 2px solid #ddd;
111          border-radius: 8px;
112          font-size: 16px;
113          transition: all 0.3s ease;
114      }
115
116      /* Efecto al enfocar los campos de entrada */
117      input:focus {
118          border-color: #23a6d5;
119          box-shadow: 0 0 3px rgba(35, 166, 213, 0.2);
120          outline: none;
121      }
122
123      /* Estilo para el botón de calcular */
124      button {
125          background: linear-gradient(to right, #23a6d5, #23d5ab);
126          color: white;
127          border: none;
128          padding: 12px 25px;
129          border-radius: 8px;
130          font-size: 16px;
131          font-weight: 500;
132          cursor: pointer;
133          transition: all 0.3s ease;
134          display: block;
135          width: 100%;
136          margin-bottom: 25px;
137      }
```

```

139      /* Efectos al pasar el mouse sobre el botón */
140      button:hover {
141          transform: translateY(-3px);
142          box-shadow: 0 7px 14px rgba(0, 0, 0, 0.1);
143      }
144
145      /* Efecto al hacer clic en el botón */
146      button:active {
147          transform: translateY(0);
148      }
149
150      /* Estilo para subtítulos */
151      h3 {
152          margin: 25px 0 15px;
153          color: #444;
154      }
155
156      /* Contenedor de la tabla con scroll horizontal */
157      .table-container {
158          overflow-x: auto;
159          border-radius: 8px;
160          box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
161      }
162
163      /* Estilo para la tabla de resultados */
164      table {
165          width: 100%;
166          border-collapse: collapse;
167          background: white;
168          overflow: hidden;
169      }

```

```
171      /* Estilo para celdas de la tabla */
172      th, td {
173          padding: 12px 15px;
174          text-align: center;
175          border: none;
176      }
177
178      /* Estilo para encabezados de la tabla */
179      th {
180          background-color: #23a6d5;
181          color: white;
182          font-weight: 500;
183          text-transform: uppercase;
184          font-size: 14px;
185          letter-spacing: 0.5px;
186      }
187
188      /* Filas alternas con color de fondo diferente */
189      tr:nth-child(even) {
190          background-color: #f8f9fa;
191      }
192
193      /* Animación para las filas de la tabla */
194      tbody tr {
195          transition: all 0.3s ease;
196          animation: fadeIn 0.5s ease-out forwards;
197          opacity: 0;
198      }
199
```

```

200  /* Animación de aparición para las filas */
201  @keyframes fadeIn {
202      from {
203          opacity: 0;
204          transform: translateY(10px);
205      }
206      to {
207          opacity: 1;
208          transform: translateY(0);
209      }
210  }
211
212  /* Efecto al pasar el mouse sobre las filas */
213  tbody tr:hover {
214      background-color: #f1f1f1;
215      transform: scale(1.01);
216  }
217
218  /* Contenedor para el indicador de carga */
219  .loading {
220      display: none;
221      text-align: center;
222      margin: 20px 0;
223  }
224
225  /* Animación del spinner de carga */
226  .loading-spinner {
227      display: inline-block;
228      width: 40px;
229      height: 40px;
230      border: 4px solid rgba(35, 166, 213, 0.3);
231      border-radius: 50%;
232      border-top-color: #23a6d5;
233      animation: spin 1s ease-in-out infinite;
234  }
235
236  /* Animación de rotación para el spinner */
237  @keyframes spin {
238      to {
239          transform: rotate(360deg);
240      }
241  }
242
243  /* Contenedor para la información del resultado */
244  .result-info {
245      margin-top: 20px;
246      padding: 15px;
247      background-color: #f8f9fa;
248      border-radius: 8px;
249      display: none;
250      animation: fadeIn 0.5s ease-out forwards;
251  }
252
253  /* Estilo para mensajes de error */
254  .error {
255      color: #e73c7e;
256      font-size: 14px;
257      margin-top: 5px;
258      display: none;
259  }

```

```

260 </style>
261 </head>
262 <body>
263   <div class="container">
264     <h2>Calculadora de  $e^x$  con Serie de McLaurin</h2>
265
266     <!-- Formulario para ingresar el valor de x -->
267     <div class="input-group">
268       <label for="x">Valor de x:</label>
269       <input type="text" id="x" placeholder="Ingrese un valor">
270       <div id="x-error" class="error">Por favor ingrese un valor válido</div>
271     </div>
272
273     <!-- Formulario para ingresar el margen de error deseado -->
274     <div class="input-group">
275       <label for="error">Margen de error (ej. 0.0001):</label>
276       <input type="text" id="error" placeholder="Ingrese el margen de error deseado">
277       <div id="error-error" class="error">Por favor ingrese un valor válido</div>
278     </div>
279
280     <!-- Botón para iniciar el cálculo -->
281     <button onclick="calcular()">Calcular</button>
282
283     <!-- Indicador de carga mientras se realizan los cálculos -->
284     <div id="loading" class="loading">
285       <div class="loading-spinner"></div>
286       <p>Calculando...</p>
287     </div>
288
289     <!-- Sección que muestra información sobre el resultado -->
290     <div id="result-info" class="result-info">
291       <p>Valor real de  $e^{\sup id="x-value"></sup>}: <span id="real-value"></span></p>
292       <p>Términos necesarios: <span id="terms-needed"></span></p>
293     </div>
294$ 
```



```

295 <!-- Sección de la tabla de resultados -->
296 <h3>Resultado:</h3>
297 <div class="table-container">
298   <table id="tablaResultados">
299     <thead>
300       <tr>
301         <th>Términos</th>
302         <th>Aproximación</th>
303         <th>Error Relativo (%)</th>
304         <th>Error Absoluto (%)</th>
305       </tr>
306     </thead>
307     <tbody></tbody>
308   </table>
309 </div>
310 </div>
311
312 <script>
313   /**
314    * Función para analizar fracciones ingresadas como texto
315    * Permite al usuario ingresar valores como "1/2" y los convierte a decimales (0.5)
316    * @param {string} input - El texto ingresado por el usuario
317    * @return {number} - El valor decimal de la fracción o el número ingresado
318    */
319   function parseFraction(input) {
320     // Verifica si el input contiene una barra de división (/)
321     if (input.includes('/')) {
322       // Divide el texto en numerador y denominador
323       const parts = input.split('/');
324       if (parts.length === 2) {
325         const numerator = parseFloat(parts[0]);
326         const denominator = parseFloat(parts[1]);
327

```

```

328     // Verifica que ambos valores sean números válidos y que el denominador no sea cero
329     if (!isNaN(numerator) && !isNaN(denominator) && denominator !== 0) {
330         return numerator / denominator; // Retorna la división
331     }
332 }
333 return NaN; // Retorna NaN si la fracción no es válida
334 }
335
336 // Si no es una fracción, intenta convertir directamente a número
337 return parseFloat(input);
338 }
339
340 /**
341  * Valida los datos ingresados por el usuario
342  * Muestra mensajes de error si los datos no son válidos
343  * @param {string} inputId - ID del elemento de entrada a validar
344  * @param {string} errorId - ID del elemento donde mostrar el error
345  * @return {number|boolean} - El valor numérico si es válido, o false si no lo es
346  */
347 function validateInput(inputId, errorId) {
348     const input = document.getElementById(inputId).value.trim();
349     const errorElement = document.getElementById(errorId);
350
351     // Verifica si el campo está vacío
352     if (input === '') {
353         errorElement.style.display = 'block';
354         return false;
355     }
356
357     let value;
358     // Para el valor de x, permite fracciones
359     if (inputId === 'x') {
360         value = parseFraction(input);
361     } else {
362         value = parseFloat(input);

```

```

365 // Verifica si el valor es un número válido
366 if (isNaN(value)) {
367     errorElement.style.display = 'block';
368     return false;
369 }
370
371 // Si todo está bien, oculta el mensaje de error y retorna el valor
372 errorElement.style.display = 'none';
373 return value;
374 }
375
376 /**
377  * Calcula el factorial de un número
378  * Necesario para la serie de McLaurin:  $e^x = 1 + x/1! + x^2/2! + x^3/3! + \dots$ 
379  * @param {number} num - El número para calcular su factorial
380  * @return {number} - El factorial del número
381  */
382 function factorial(num) {
383     if (num === 0 || num === 1) return 1; // Caso base:  $0! = 1! = 1$ 
384
385     let fact = 1;
386     // Multiplica todos los números desde 2 hasta num
387     for (let i = 2; i <= num; i++) {
388         fact *= i;
389     }
390     return fact;
391 }
392

```

```

393  /**
394  * Funci3n principal que realiza el c3lculo de e^x usando la serie de McLaurin
395  * Muestra los resultados en una tabla con la aproximaci3n y los errores
396  */
397  function calcular() {
398      // Oculta los mensajes de error previos
399      document.getElementById('x-error').style.display = 'none';
400      document.getElementById('error-error').style.display = 'none';
401
402      // Valida los datos ingresados
403      const x = validateInput('x', 'x-error');
404      const margen = validateInput('error', 'error-error');
405
406      // Si alg3n dato no es v3lido, detiene la ejecuci3n
407      if (x === false || margen === false) {
408          return;
409      }
410
411      // Muestra el indicador de carga
412      document.getElementById('loading').style.display = 'block';
413
414      // Limpia la tabla de resultados anteriores
415      const tabla = document.getElementById("tablaResultados").querySelector("tbody");
416      tabla.innerHTML = "";
417
418      // Usa setTimeout para no bloquear la interfaz durante los c3lculos
419      setTimeout(() => {
420          // Calcula el valor real de e^x usando la funci3n Math.exp
421          const resultadoReal = Math.exp(x);
422
423          let suma = 1;    // Primer t3rmino de la serie (x^0/0! = 1)
424          let anterior = 0; // No hay t3rmino anterior para el primer t3rmino
425          let n = 0;       // Contador de t3rminos (comenzamos desde el primer t3rmino)
426          let delay = 0;    // Retraso para la animaci3n de las filas
427

```

```

428 // Para el primer término, calculamos solo el error absoluto
429 let errorAbsolutoPorc = (suma / resultadoReal) * 100;
430 let errorAbsolutoPorcFinal = 100 - errorAbsolutoPorc;
431
432 // Crea la primera fila en la tabla
433 const primeraFila = tabla.insertRow();
434 primeraFila.style.animationDelay = `${delay}ms`;
435 delay += 50;
436
437 primeraFila.innerHTML = `
438   <td>${n + 1}</td>
439   <td>${suma.toFixed(10)}</td>
440   <td>N/A</td>
441   <td>${errorAbsolutoPorcFinal.toFixed(6)}%</td>
442 `;
443
444 // Incrementa el contador para el siguiente término
445 n++;
446
447 // Bucle principal para calcular la serie de McLaurin
448 while (true) {
449   anterior = suma; // Guardamos el valor anterior
450
451   // Calcula el siguiente término:  $x^n/n!$ 
452   let nuevoTermino = Math.pow(x, n) / factorial(n);
453   suma += nuevoTermino; // Añade el término a la suma
454
455   // Calcula los errores
456   // Error relativo:  $(\text{Aproximación actual} - \text{Aproximación previa}) / \text{Aproximación previa} * 100$ 
457   let errorRelativoPorc = Math.abs((suma - anterior) / anterior) * 100;
458
459   // Error absoluto:  $(\text{Aproximación} / \text{valor calculado de } e^x) * 100 = 100 - \text{resultado}$ 
460   errorAbsolutoPorc = (suma / resultadoReal) * 100;
461   errorAbsolutoPorcFinal = 100 - errorAbsolutoPorc;
462
463   // Crea una nueva fila en la tabla para este término
464   const fila = tabla.insertRow();
465   fila.style.animationDelay = `${delay}ms`; // Retraso para efecto escalonado
466   delay += 50; // Incrementa el retraso para la siguiente fila
467
468   // Llena la fila con los datos calculados
469   fila.innerHTML = `
470     <td>${n + 1}</td>
471     <td>${suma.toFixed(10)}</td>
472     <td>${errorRelativoPorc.toFixed(6)}%</td>
473     <td>${errorAbsolutoPorcFinal.toFixed(6)}%</td>
474   `;
475
476   // Si el error absoluto es menor o igual al margen deseado, termina
477   if (Math.abs(errorAbsolutoPorcFinal) <= margen) break;
478
479   // Incrementa el contador para el siguiente término
480   n++;
481   // Límite de seguridad para evitar bucles infinitos
482   if (n > 100) break;
483 }
484
485 // Actualiza la información del resultado
486 document.getElementById('x-value').textContent = x;
487 document.getElementById('real-value').textContent = resultadoReal.toFixed(10);
488 document.getElementById('terms-needed').textContent = n + 1;
489 document.getElementById('result-info').style.display = 'block';
490
491 // Oculta el indicador de carga
492 document.getElementById('loading').style.display = 'none';

```

```
493     }, 300); // Pequeño retraso para mostrar la animación de carga
494 }
495
496 // Evento para permitir navegar entre campos con la tecla Enter
497 document.getElementById('x').addEventListener('keypress', function(e) {
498     if (e.key === 'Enter') {
499         document.getElementById('error').focus(); // Pasa al siguiente campo
500     }
501 });
502
503 // Evento para iniciar el cálculo al presionar Enter en el último campo
504 document.getElementById('error').addEventListener('keypress', function(e) {
505     if (e.key === 'Enter') {
506         calcular(); // Inicia el cálculo
507     }
508 });
509 </script>
510 </body>
511 </html>
```