CALCULADORA DE ECUACIÓNES MATEMATICAS

MAIRA ALEJANDRA RODRIGUEZ VELASCO

UNIVERSIDAD LIBRE

INGENIERIA DE SISTEMAS

PROGRAMACIÓN WEB

CALI

2025

CALCULADORA DE ECUACIÓNES MATEMATICAS

MAIRA ALEJANDRA RODRIGUEZ VELASCO

INFORME

RAFAEL ALBERTO MORENO PARRA

DOCENTE INVESTIGACIÓN

UNIVERSIDAD LIBRE

INGENIERIA

PTOGRAMACIÓN WEB

CALI

2025

CONTENIDO

[1. INTRODUCCIÓN 4](#_Toc191954829)

[2. OBJETIVOS 5](#_Toc191954830)

[2.1 OBJETIVO GENERAL 5](#_Toc191954831)

[2.2 OBJETIVOS ESPECÍFICOS 5](#_Toc191954832)

[3. DESARROLLO DEL PROYECTO 6](#_Toc191954833)

[3.1 FASE 1: Análisis y diseño 6](#_Toc191954834)

[3.2 FASE 2: Implementación 6](#_Toc191954835)

[3.2.1 Estructura HTML 6](#_Toc191954836)

[3.2.2 Estilos CSS 8](#_Toc191954837)

[3.2.3 Script en JavaScript 12](#_Toc191954838)

[3.3 FASE 3: Pruebas y ajustes 14](#_Toc191954839)

[3.4 FUNCIONALIDADES IMPLEMENTADAS 14](#_Toc191954840)

[4. ANALISIS DEL DESARROLLO DEL PROYECTO 16](#_Toc191954841)

[5. CONCLUSIONES 17](#_Toc191954842)

[BIBLIOGRAFÍA 18](#_Toc191954843)

Pág.

# INTRODUCCIÓN

En el presente informe, tiene como objetivo el desarrollo de una calculadora de ecuaciones con representación gráfica y tabla de valores. Mediante una interfaz web intuitiva, los usuarios pueden ingresar coeficientes para una ecuación específica y observar tanto su gráfica como los valores numéricos correspondientes en un rango determinado.

El desarrollo de esta herramienta se fundamenta en tecnologías web como HTML, CSS y JavaScript, además del uso de la biblioteca Google Charts para la generación de gráficos dinámicos. Se han implementado algoritmos matemáticos que permiten calcular con precisión valores característicos de la función, asegurando así la fiabilidad de los resultados obtenidos.

En este informe se detallan los objetivos del proyecto, la metodología empleada para su desarrollo, los desafíos enfrentados y las soluciones implementadas. Además, se presentan los resultados obtenidos y se analizan posibles mejoras futuras para optimizar la experiencia del usuario y la precisión de los cálculos.

# 2. OBJETIVOS

## 2.1 OBJETIVO GENERAL

Diseñar y desarrollar una aplicación web interactiva que permita el análisis visual y numérico de una ecuación matemática, proporcionando una gráfica y una tabla de valores asociada, con el fin de facilitar la comprensión y el estudio de funciones.

## 2.2 OBJETIVOS ESPECÍFICOS

* Implementar una interfaz web intuitiva que permita el ingreso de coeficientes de manera sencilla y accesible.
* Desarrollar un algoritmo eficiente para calcular los valores de la ecuación en un rango de interés definido por el usuario.
* Determinar y mostrar los valores máximos y mínimos de la función dentro del intervalo establecido.
* Detectar y visualizar los puntos de corte con el eje X, es decir, los valores donde la función se anula.
* Generar automáticamente una tabla de valores con los pares ordenados
* (x, y), facilitando el análisis numérico de la función.
* Implementar una representación gráfica interactiva mediante Google Charts para mejorar la visualización de la ecuación.
* Optimizar el rendimiento del sistema, asegurando una rápida respuesta y precisión en los cálculos.

# 3. DESARROLLO DEL PROYECTO

El desarrollo de este proyecto se llevó a cabo en varias etapas, comenzando con la conceptualización del problema y la identificación de los requisitos funcionales y técnicos. Posteriormente, se procedió con la implementación del código, las pruebas y los ajustes necesarios para garantizar su correcto funcionamiento.

|  |
| --- |
| 3.1 FASE 1: Análisis y diseño |

En esta fase se definió la ecuación a representar y se establecieron los requisitos del sistema. Se decidió que la ecuación tendría la siguiente forma:

Los coeficientes (a, b, c, d, e) pueden ser ingresados por el usuario, permitiendo explorar diversas variaciones de la función. Se establecieron restricciones para evitar errores matemáticos, como divisiones por cero.

## 3.2 FASE 2: Implementación

Se desarrolló la interfaz gráfica utilizando HTML y CSS, asegurando un diseño responsivo y atractivo. La funcionalidad del sistema fue programada en JavaScript, implementando la lógica matemática para calcular los valores de la ecuación y detectar puntos relevantes.

### 3.2.1 Estructura HTML

El código HTML define la estructura de la interfaz de usuario, permitiendo la entrada de datos y la visualización de la gráfica generada.

index.html

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Calculadora de Ecuaciones</title>      <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>      <script type="text/javascript" async src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.7/MathJax.js?config=TeX-MML-AM\_CHTML"></script>      <link rel="stylesheet" href="styles.css">  </head>  <body>      <div class="container">          <header>              <h1>CALCULADORA DE ECUACIONES CON GRÁFICOS Y TABLA DE VALORES</h1>              <h3>AUTOR@: MAIRA ALEJANDRA RODRIGUEZ VELASCO - CÓDIGO: 2216670</h3>          </header>          <section class="equation-section">          <h2>ECUACIÓN</h2>          <img src="Ecuacion.png" alt="Ecuación Matemática" class="ecuation-img">          </section>          <section class="input-section">          <h2>COEFICIENTES</h2>          <h3>Ingrese los coeficientes:</h3>          <div class="input-container">              <label>a: <input type="number" id="a" value="1"></label>              <label>b: <input type="number" id="b" value="1"></label>              <label>c: <input type="number" id="c" value="1"></label>              <label>d: <input type="number" id="d" value="1"></label>              <label>e: <input type="number" id="e" value="1"></label>          </div>          <div class="input-container">              <label>X min: <input type="number" id="xMin" value="-10"></label>              <label>X max: <input type="number" id="xMax" value="10"></label>          </div>          <button id="procesar">Procesar</button>          </section>          <section class="result-section">              <h2>RESULTADOS</h2>              <p class="p-color"><b>MÍNIMO Y </b></p>              <p><span id="minY"></span></p>              <p class="p-color"><b>MÁXIMO Y </b></p>              <p><span id="maxY"></span></p>              <p class="p-color"><b>CORTES EN X </b></p>              <p><span id="cortesX"></span></p>              <p class="p-color"><b>TOTAL DE CORTES EN X</b></p>              <p><span id="totalCortes"></span></p>          </section>          <section class="table-section">              <h2>TABLA DE VALORES</h2>              <table id="tablaValores">                  <thead>                      <tr><th>X</th><th>Y</th></tr>                  </thead>                  <tbody></tbody>              </table>          </section>          <section class="chart-section">              <h2>GRÁFICO</h2>              <div id="chart\_div" class="chart-container"></div>          </section>      </div>      <script src="script.js"></script>  </body>  </html> |

### 3.2.2 Estilos CSS

El código CSS mejora la presentación visual de la interfaz.

style.css

|  |
| --- |
| body {      font-family: Arial, sans-serif;      background-color: #f8f0fc;      text-align: center;      margin: 0;      padding: 0;  }  .container {      width: 90%;      max-width: 1000px;      margin: auto;      background: white;      padding: 15px;      border-radius: 10px;      box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);  }  h1 {      color: #6a0572;      font-size: 50px;      font-family:Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;  }  h2 {      color: #d63384;      font-family: Verdana, Geneva, Tahoma, sans-serif;  }  h3 {      color: #235ea0;  }  .input-container {      display: flex;      justify-content: center;      gap: 10px;      margin: 10px 0;  }  input {      padding: 8px;      border: 2px solid #d63384;      border-radius: 5px;      outline: none;      width: 80px;      text-align: center;  }  button {      background-color: #6a0572;      color: white;      padding: 10px 20px;      border: none;      border-radius: 5px;      font-size: 16px;      cursor: pointer;      margin-top: 10px;  }  button:hover {      background-color: #d63384;  }  button:active {      transform: scale(0.98);  }  table {      border-collapse: collapse;      margin: auto;      background-color: white;      width: 100%;      min-width: 400px;  }  .table-section {      overflow-x: auto; /\* Permite desplazamiento lateral \*/      max-width: 100%;  }  th, td {      border: 1px solid #6a0572;      padding: 10px;      text-align: center;  }  th {      background-color: #d63384;      color: white;  }  .chart-container {      width: 800px;      height: 500px;      margin: auto;      overflow-x: auto;  }  .p-color {      color: palevioletred;     }  .ecuation-img {      max-width: 100%;      height: auto;      display: block;      margin: 10px auto;      border: 3px solid #6a0572;      border-radius: 20px;      background: white;      padding: 20px;      box-sizing: border-box;  }  header, .equation-section, .input-section, .result-section, .table-section, .chart-section {      margin-bottom: 20px;      padding: 15px;      border-radius: 10px;      background-color: #fafafa;      box-shadow: 0px 2px 5px rgba(0, 0, 0, 0.1);  }  @media (max-width: 600px) {      .container {          width: 95%;          padding: 10px;      }      .input-container {          flex-direction: column;          gap: 5px;      }      input {          width: 100%;      }      .chart-container {          width: 100%;          height: auto;      }  } |

### 3.2.3 Script en JavaScript

Este código JavaScript implementa la lógica para calcular los valores de la ecuación y generar la gráfica.

Los cálculos matemáticos se realizaron utilizando funciones estándar de JavaScript para trigonometría y álgebra. Se empleó un bucle para evaluar la ecuación en diferentes valores de “x”, generando así los datos para la tabla y la gráfica.

Para la representación visual, se utilizó la API de Google Charts, que permite trazar gráficos de línea interactivos. La integración con esta biblioteca se realizó de manera dinámica, actualizando el gráfico cada vez que el usuario introduce nuevos valores.

script.js

|  |
| --- |
| document.addEventListener("DOMContentLoaded", function() {      document.getElementById("procesar").addEventListener("click", procesarDatos);  });  function calcularY(x, a, b, c, d, e) {      if (!isFinite(x)) return 0; // Evitar valores inválidos      if (Math.abs(x + 1) < 1e-5) return 0; // Evitar división por cero      let sqrt1 = Math.sqrt(Math.abs(x - c)) + 1;      let sqrt2 = Math.sqrt(Math.abs(x - e)) + 1;      let denom3 = Math.abs(x + 1) < 1e-10 ? 1e-10 : (x + 1);      let term1 = (Math.sin(a \* x) + Math.cos(b \* x)) / sqrt1;      let term2 = (Math.cos(d \* x) - Math.sin(e \* x)) / sqrt2;      let term3 = (Math.sin(x) \* Math.cos(x)) / denom3;      return Math.max(-20, Math.min(20, term1 + term2 + term3));  }  function procesarDatos() {      let a = parseFloat(document.getElementById("a").value) || 0;      let b = parseFloat(document.getElementById("b").value) || 0;      let c = parseFloat(document.getElementById("c").value) || 0;      let d = parseFloat(document.getElementById("d").value) || 0;      let e = parseFloat(document.getElementById("e").value) || 0;      let xMin = parseFloat(document.getElementById("xMin").value) || -10;      let xMax = parseFloat(document.getElementById("xMax").value) || 10;      if (isNaN(xMin) || isNaN(xMax) || xMax <= xMin) {          alert("Ingrese valores válidos para X min y X max (X max debe ser mayor).");          return;      }      let coeficientes = [a, b, c, d, e];      if (coeficientes.some(isNaN)) {      alert("Ingrese valores numéricos válidos para los coeficientes.");      return;      }        let valores = [];      let minY = Infinity, maxY = -Infinity;      let cortesX = [];      let paso = 0.1;      let yAnt = calcularY(xMin, a, b, c, d, e);      for (let x = xMin; x <= xMax; x += paso) {          let y = calcularY(x, a, b, c, d, e);          valores.push({ x, y });          if (y < minY) minY = y;          if (y > maxY) maxY = y;          // Detección de corte en X (cambio de signo)          if (yAnt \* y < 0) {              let xCorte = x - paso / 2; // Aproximación de la raíz              cortesX.push(xCorte.toFixed(4));          }          yAnt = y;      }      document.getElementById("minY").innerText = minY.toFixed(4);      document.getElementById("maxY").innerText = maxY.toFixed(4);      document.getElementById("cortesX").innerText = cortesX.length > 0 ? cortesX.join(", ") : "Ninguno";      document.getElementById("totalCortes").innerText = cortesX.length;      let tabla = document.querySelector("#tablaValores tbody");      tabla.innerHTML = valores.map(v => `<tr><td>${v.x.toFixed(2)}</td><td>${v.y.toFixed(4)}</td></tr>`).join("");      google.charts.load('current', { 'packages': ['corechart'] });      google.charts.setOnLoadCallback(() => {          let data = new google.visualization.DataTable();          data.addColumn('number', 'X');          data.addColumn('number', 'Y');          valores.forEach(v => data.addRow([v.x, v.y]));          let chart = new google.visualization.LineChart(document.getElementById('chart\_div'));          chart.draw(data, { title: 'Gráfico de la función', legend: 'none' });      });  } |

## 3.3 FASE 3: Pruebas y ajustes

Se realizaron pruebas con distintos valores de los coeficientes para verificar la precisión de los cálculos. Se corrigieron errores relacionados con valores indefinidos y divisiones por cero. Además, se optimizó el código para mejorar el rendimiento y la velocidad de respuesta de la aplicación.

## 3.4 FUNCIONALIDADES IMPLEMENTADAS

* **Ingreso de coeficientes y rango de valores**: El usuario puede ingresar los coeficientes de la ecuación y definir los valores mínimo y máximo de “x”.
* **Cálculo automático de valores de la función**: La aplicación evalúa la ecuación en un intervalo definido con un paso determinado.
* **Identificación de valores extremos**: Se determinan el mínimo y el máximo de la función dentro del rango seleccionado.
* **Detección de cortes con el eje X**: Se analiza el cambio de signo en los valores de la función para identificar aproximaciones a las raíces.
* **Generación de tabla de valores**: Se muestran los valores calculados en una tabla organizada.
* **Representación gráfica**: Se visualiza la función en un gráfico interactivo, facilitando la interpretación de la ecuación.

# 4. ANALISIS DEL DESARROLLO DEL PROYECTO

El desarrollo del proyecto presentó diversos retos que fueron abordados mediante soluciones adecuadas:

* **Precisión en los cálculos**: Se utilizó un paso pequeño en la evaluación de la función para mejorar la precisión de los valores obtenidos.
* **Manejo de errores matemáticos**: Se implementaron restricciones para evitar errores como divisiones por cero o raíces de números negativos.
* **Optimización del gráfico**: Se ajustaron los parámetros de Google Charts para mejorar la visualización y la interactividad del gráfico.
* **Mejora en la detección de cortes en “x”**: Se implementó un algoritmo que detecta cambios de signo en la función para identificar puntos de intersección con el eje “x”.

# 5. CONCLUSIONES

El desarrollo de este proyecto ha permitido demostrar la importancia de las herramientas digitales en la visualización y análisis de funciones matemáticas. A través de la implementación de tecnologías web, se ha logrado crear una aplicación interactiva que facilita la comprensión de ecuaciones mediante gráficos y tablas de valores.

El uso de esta herramienta no solo simplifica el proceso de análisis matemático, sino que también ofrece una experiencia más intuitiva y accesible para distintos usuarios. Además, su capacidad para calcular puntos clave, como máximos, mínimos y cortes en el eje X, refuerza su utilidad en la exploración de funciones.

A lo largo del proceso, se superaron diversos desafíos técnicos que contribuyeron al aprendizaje y mejora del sistema, permitiendo optimizar su rendimiento y precisión. Este proyecto sienta las bases para futuras mejoras y expansiones, como la incorporación de nuevas funcionalidades y la adaptación a distintos dispositivos y necesidades.

En conclusión, el desarrollo de esta calculadora reafirma el valor de la tecnología en la educación y el análisis matemático, ofreciendo una herramienta versátil con un gran potencial de crecimiento y aplicación en diferentes ámbitos.

# BIBLIOGRAFÍA

* Asth, R. C. (2024, febrero 20). *Funciones Trigonométricas: Cuáles son, Fórmulas y Gráficas*. Enciclopedia Significados. <https://www.significados.com/funciones-trigonometricas/>
* Boneu, M. S. (2021, enero 8). *Funciones Math en JavaScript explicadas*. freecodecamp.org. <https://www.freecodecamp.org/espanol/news/funciones-math-en-javascript-explicadas/>
* *Charts*. (s/f). Google for Developers. Recuperado el 28 de febrero de 2025, de <https://developers.google.com/chart?hl=es-419>
* CodeCogs, Z. L. (s/f). *Equation Editor for online mathematics*. Codecogs.com. Recuperado el 1 de marzo de 2025, de <https://editor.codecogs.com/>
* *Códigos de Colores HTML*. (2015, septiembre 3). HTML Color Codes. <https://htmlcolorcodes.com/es/>