



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Trabajo Terminal I

**“Desarrollo de una aplicación web para el
cálculo y graficación de series de Fourier”**

2025-A075

QUE PARA OBTENER EL GRADO DE:
Ingeniero en Sistemas Computacionales

PRESENTA:
Sebastián Morales Palacios

DIRECTORES:
Lic. Jesús Alfredo Martínez Nuño
M. en C. Gisela González Albarrán



Ciudad de México
Noviembre 2024

Documento Técnico

“Desarrollo de una aplicación web para el cálculo y graficación de series de Fourier”

Resumen - Dentro del presente trabajo se lleva a cabo el desarrollo de una aplicación web diseñada para calcular y graficar el desarrollo en series de Fourier de funciones periódicas. El desarrollo en serie puede ser tanto en forma trigonométrica como en forma exponencial compleja. Los módulos principales que contiene la aplicación son:

1. **Módulo para introducir la función** de diferentes maneras.
2. **Módulo para el cálculo de coeficientes** mediante integración analítica o por métodos numéricos.
3. **Módulo de graficación de funciones periódicas y coeficientes de la serie**, donde se podrá ver la aproximación de la serie a la función mientras se incrementa el número de términos de la expansión.

Se tienen opciones para calcular extensiones de funciones pares e impares. La aplicación pretende ser de utilidad como herramienta de apoyo para estudiantes y docentes en el tema del análisis de Fourier.

Palabras Clave - aplicación web, análisis de Fourier, cálculo, matemáticas avanzadas para la ingeniería.

Correo de Contacto -
smoralesp1700@alumno.ipn.mx

Índice general

Documento Técnico	III
1. Introducción	1
1.1. Antecedentes	1
1.2. Planteamiento del Problema	1
1.3. Justificación	2
1.4. Solución Propuesta	3
1.5. Objetivos	4
1.5.1. Objetivo General	4
1.5.2. Objetivos Específicos	4
1.6. Metodología de la Investigación	4
1.7. Estructura del documento técnico	5
2. Estado del Arte	7
2.1. Herramientas y Tecnologías Actuales	7
2.1.1. Comparativa del Funcionamiento de las Herramientas	9
2.1.1.1. Cálculo de Serie de Fourier con Wolfram Alpha	10
2.1.1.2. Cálculo de Serie de Fourier con MATLAB	12
2.1.1.3. Cálculo de Serie de Fourier con GeoGebra	12
2.1.1.4. Cálculo de Serie de Fourier con Python (Matplotlib/Manim)	12
2.2. Trabajos y Proyectos Relacionados	12
3. Marco Teórico	13
3.1. Origen e Historia	13
3.1.1. El problema de la cuerda oscilante	13
3.1.2. La ecuación de calor	18
3.2. Series de Fourier	18
3.2.1. Series de Funciones	18
3.2.1.1. Definición de Series de Funciones	18
3.2.1.2. Convergencia de Series de Funciones	18
3.2.2. Definición de las Funciones Trigonométricas de Seno y Coseno	18
3.2.2.1. Definición del Seno	18
3.2.2.2. Definición del Coseno	18
3.2.2.3. Propiedades y Relaciones Fundamentales	18
3.2.3. Funciones Periódicas	18
3.2.3.1. Definición de Funciones Periódicas	18

3.2.3.2.	Propiedades de las Funciones Periódicas	18
3.2.4.	Armónicos	18
3.2.4.1.	Definición de Armónicos	18
3.2.4.2.	Importancia de los Armónicos en Series de Fourier	18
3.2.5.	Funciones Ortogonales	18
3.2.5.1.	Concepto de Ortogonalidad	18
3.2.5.2.	Ejemplos de Funciones Ortogonales	18
3.2.6.	Originalidad del Seno y Coseno	18
3.2.6.1.	Propiedades del Seno y el Coseno	18
3.2.6.2.	Uso del Seno y el Coseno en Series de Fourier	18
3.2.7.	Series de Fourier de Período T o de $-p$ a p	18
3.2.7.1.	Generalización de la Serie de Fourier	18
3.2.7.2.	Coefficientes de Fourier para Período T	18
3.2.8.	Funciones Suaves y Suaves por Partes	18
3.2.8.1.	Definición de Funciones Suaves	18
3.2.8.2.	Definición de Funciones Suaves por Partes	18
3.2.8.3.	Aplicaciones en Series de Fourier	18
3.2.9.	Series de Fourier de Senos y Cosenos	18
3.2.9.1.	Condiciones para la Cancelación de Coeficientes	18
3.2.9.2.	Desarrollo de Series de Fourier con Senos y Cosenos	18
3.2.10.	Funciones Pares e Impares	18
3.2.10.1.	Definición y Propiedades de Funciones Pares	18
3.2.10.2.	Definición y Propiedades de Funciones Impares	18
3.2.11.	Extensiones de Medio Intervalo	18
3.2.11.1.	Extensión Periódica	18
3.2.11.2.	Extensión Par	18
3.2.11.3.	Extensión Impar	18
3.2.12.	Serie Compleja de Fourier de Período T o de $-p$ a p	18
3.2.12.1.	Definición de Serie Compleja de Fourier	18
3.2.12.2.	Coefficientes Complejos para Período T	18
3.2.13.	Serie Compleja de Fourier de 0 a p	18
3.2.13.1.	Definición y Desarrollo	18
3.2.13.2.	Aplicaciones y Ejemplos	19
3.3.	Tecnologías Utilizadas	19
3.3.1.	HTML, CSS y JavaScript	19
3.3.1.1.	HTML	19
3.3.1.2.	CSS	19
3.3.1.3.	JavaScript	19
3.3.2.	NodeJS	19
3.3.2.1.	Introducción a NodeJS	19
3.3.2.2.	Ventajas del uso de NodeJS para aplicaciones web	19
3.3.2.3.	Integración de NodeJS con Maxima	19
3.3.3.	Maxima	19
3.3.3.1.	Qué es Maxima y por qué se usa para cálculos simbólicos	19
3.3.3.2.	Implementación de Maxima en el servidor	19
3.3.4.	Angular (Posible implementación futura)	19
3.3.4.1.	Introducción a Angular	19

3.3.4.2.	Ventajas del uso de Angular para futuras versiones del proyecto	20
3.3.4.3.	Consideraciones para su implementación	20
3.3.5.	Metodologías de Desarrollo de Software	20
3.3.5.1.	Desarrollo Ágil	20
3.3.5.2.	Fases del Proyecto	20
3.3.6.	Usabilidad y Experiencia de Usuario (UX)	20
3.3.6.1.	Principios de Usabilidad	20
3.3.6.2.	Diseño de la Interfaz de Usuario (UI)	20
3.3.7.	Seguridad en Aplicaciones Web	20
3.3.7.1.	Protección contra ataques comunes	20
3.3.7.2.	Autenticación y manejo seguro de datos	20
4.	Análisis y Diseño del Sistema	21
4.1.	Metodología de Desarrollo	21
4.1.1.	Descripción de la Metodología	21
4.1.2.	Ventajas de la Metodología Elegida	21
4.2.	Análisis de Requerimientos	21
4.2.1.	Requerimientos Funcionales	21
4.2.2.	Requerimientos No Funcionales	26
4.3.	Análisis de Riesgos	27
4.3.1.	Identificación de Riesgos	27
4.3.2.	Jerarquización de Riesgos	27
4.4.	Estimación de Costos	27
4.4.1.	Método de Estimación de Costos	27
4.4.2.	Detalle de los Costos Estimados	27
4.5.	Diseño del Sistema	27
4.5.1.	Arquitectura General del Sistema	27
4.5.1.1.	Arquitectura Cliente-Servidor	27
4.5.1.2.	Flujo de Datos	27
4.5.2.	Diseño de la Interfaz de Usuario	27
4.5.2.1.	Estructura de la Interfaz	27
4.5.2.2.	Prototipo de la Interfaz	27
5.	Implementación	
	(Primer Prototipo)	29
A.	Cálculo de los Coeficientes de la Serie de Fourier	31
A.1.	Forma Trigonométrica	31
A.2.	Forma Exponencial Compleja	32

Índice de figuras

2.1.	Gráfica de los coeficientes de Fourier calculados en el Apendice A	10
2.2.	Calculo de una serie trigonométrica de Fourier en WolframAlpha .	11
2.3.	Calculo de una serie trigonométrica de Fourier en WolframAlpha .	11
3.1.	Ilustración del problema de la cuerda vibrante.	14
3.2.	Ilustración de 3 armónicos y sus nodos de la cuerda al vibrar. . .	16

Lista de tablas

2.1. Comparación de software para cálculos matemáticos y visualización de datos	9
4.1. Requerimiento funcional No. 1	22
4.2. Requerimiento funcional No. 2	23
4.3. Requerimiento funcional No. 3	23
4.4. Requerimiento funcional No. 4	24
4.5. Requerimiento funcional No. 5	24
4.6. Requerimiento funcional No. 6	25
4.7. Requerimiento funcional No. 7	25
4.8. Requerimiento funcional No. 8	26
4.9. Tabla de requerimientos no funcionales	27

List of Algorithms

CAPÍTULO 1

Introducción

En este capítulo, examinaremos el problema que se abordará y los objetivos previstos para este proyecto. Además, investigaremos los productos existentes que desempeñan una función similar al proyecto que se llevará a cabo.

1.1. Antecedentes

Aquí voy a buscar los antecedentes

1.2. Planteamiento del Problema

Las series de Fourier, desde su concepción por Jean-Baptiste Joseph Fourier a principios del siglo XIX, han desempeñado un rol crucial en el análisis y la comprensión de señales y fenómenos periódicos. Las series de Fourier son parte esencial en campos como la ingeniería eléctrica, la física teórica y el procesamiento de señales, imágenes y audio, permitiendo descomponer funciones periódicas en sumas de senos y cosenos, lo que facilita su análisis y manipulación. Estas herramientas matemáticas no solo han impulsado avances significativos en la ciencia y tecnología, transformando nuestra interacción con el mundo, sino que también son cruciales en la enseñanza de los fundamentos teóricos de la ingeniería. En el ámbito de las ciencias computacionales, el estudio del análisis de Fourier se aplica a áreas vitales como la adquisición y procesamiento de señales, el procesamiento de imágenes y la robótica. Esta capacidad de simplificar señales complejas en componentes básicos no solo mejora el análisis, sino que también facilita la síntesis de nuevas tecnologías que se adaptan a necesidades y entornos cambiantes [1].

Este amplio espectro de aplicaciones destaca la importancia crítica de las series de Fourier no solo en la investigación avanzada, sino también en la formación académica de futuros ingenieros y científicos. Sin embargo, a pesar de su prevalencia en el currículo educativo, la implementación práctica de este análisis en entornos de aprendizaje a menudo revela áreas de oportunidad dentro de las herramientas disponibles, lo que impacta directamente en la eficacia con la que los estudiantes pueden aplicar y profundizar su comprensión de estos conceptos esenciales.

La necesidad de una herramienta más eficiente se hizo evidente durante los

cursos de Matemáticas Avanzadas para la Ingeniería y Procesamiento de Señales Digitales. Al enfrentar el análisis de Fourier, especialmente al resolver ejercicios sobre series de Fourier, se descubrió la falta de medios óptimos para verificar los resultados de manera directa y eficiente. A diferencia de otros cálculos matemáticos, donde las calculadoras especializadas permiten comprobaciones rápidas y fiables, las series de Fourier requerían un proceso mucho más tedioso. Para asegurar la corrección de los cálculos, era necesario utilizar software de resolución matemática para obtener los coeficientes y, posteriormente, otro software de graficación para visualizar si los resultados correspondían efectivamente a la función dada. Este desafío se ampliaba aún más cuando se presentaban variaciones mínimas en los ejercicios, como cambiar el intervalo de la función o alternar entre series trigonométricas y complejas, o incluso al aplicar extensiones pares o impares. Cada una de estas variaciones obligaba a repetir todos los pasos desde el inicio, lo que no solo consumía tiempo valioso, sino que también complicaba la gestión del tiempo disponible, especialmente cuando se tiene una carga académica intensa. Este tiempo podría utilizarse mejor en comprender los conceptos subyacentes y explorar en profundidad el por qué y el cómo de los fenómenos analizados mediante estas series.

1.3. Justificación

A pesar de la indiscutible importancia del análisis de Fourier en diversas áreas de la ciencia y la ingeniería, el acceso a herramientas que integren de manera eficiente el cálculo y la visualización de las series de Fourier aún presenta importantes oportunidades de mejora. Las herramientas actuales, aunque avanzadas y robustas, suelen fragmentar el proceso, obligando a los usuarios a utilizar distintos programas o plataformas para realizar los cálculos y la representación gráfica de los resultados. Esta división entre herramientas de cálculo y visualización añade una capa de complejidad y tiempo, especialmente para aquellos usuarios que, además de realizar cálculos precisos, requieren visualizar los resultados de manera rápida y efectiva.

En este contexto, surge la propuesta de desarrollar un prototipo de aplicación web que combine ambas funcionalidades en una sola interfaz, facilitando el proceso de resolver y graficar las series de Fourier de manera práctica, intuitiva y accesible desde cualquier dispositivo con acceso a Internet. La creación de una plataforma que integre estos aspectos tiene el potencial de mejorar significativamente la experiencia del usuario al permitirle resolver ecuaciones y obtener representaciones gráficas de las series de Fourier en un solo entorno, eliminando la necesidad de conocimientos técnicos avanzados o el uso de múltiples herramientas.

El desarrollo de esta aplicación tiene el potencial de ser un aporte valioso para diversas comunidades, como estudiantes, ingenieros y profesionales del área de las ciencias exactas, que requieren manipular y visualizar series de Fourier en sus trabajos. La integración de funcionalidades de cálculo simbólico con visualización gráfica interactiva permitirá un mayor grado de experimentación

y aprendizaje, reduciendo el tiempo y esfuerzo necesarios para comprender y analizar las transformaciones de Fourier.

Además, este proyecto representa una oportunidad para integrar y aplicar los conocimientos y habilidades adquiridos a lo largo de mi formación en ingeniería en sistemas. Al abordar tanto aspectos de programación, como de matemáticas avanzadas y principios de ingeniería de software, este trabajo terminal no solo permite abordar una necesidad práctica, sino que también sirve como un medio para demostrar la capacidad de aplicar teoría y técnicas de ingeniería en un contexto real. De esta manera, no solo se busca desarrollar una herramienta útil para otros, sino también consolidar y exhibir competencias en áreas clave de mi carrera.

1.4. Solución Propuesta

La solución que se plantea en este proyecto es el desarrollo de una aplicación web que permita el cálculo y graficación de series de Fourier de manera integrada, eficiente y accesible, cubriendo la brecha existente en las herramientas actuales, que tienden a separar el cálculo matemático de la visualización gráfica en una sola interfaz. La solución propuesta combinará estas dos funcionalidades en una sola plataforma, diseñada para ser simple de usar, práctica y accesible desde cualquier navegador web, para usuarios que deseen analizar y visualizar series de Fourier, eliminando la necesidad de usar múltiples programas para resolver y graficar las series de Fourier.

La aplicación web contará con las siguientes características clave:

- **Cálculo automático de series de Fourier para diferentes tipos de funciones:**
 - La aplicación calculará **series de Fourier** para **funciones continuas** en un intervalo o para **funciones definidas a trozos**. El usuario podrá ingresar funciones matemáticas, incluso aquellas con discontinuidades o que estén definidas por partes en distintos intervalos, y la aplicación manejará estos casos automáticamente.
 - Se implementará tanto la **serie trigonométrica** como la **serie exponencial compleja**. La serie trigonométrica descompondrá la función en términos de senos y cosenos, mientras que la versión compleja lo hará en términos de exponenciales imaginarios, lo cual es útil en el contexto de análisis más avanzado y en aplicaciones de ingeniería.
- **Extensiones de medio rango:**
 - La herramienta permitirá calcular **extensiones de medio rango** para funciones en un medio intervalo. Esto incluirá la posibilidad de obtener:
 - **Serie de senos** para funciones impares.
 - **Serie de cosenos** para funciones pares.

- **Serie completa** para funciones periódicas en un intervalo definido, permitiendo extender la función para construir series de Fourier en un medio rango específico.
- **Visualización gráfica interactiva:**
 - Una vez calculados los coeficientes de Fourier, la aplicación generará una **gráfica interactiva** que mostrará la aproximación de la serie de Fourier a la función original. El usuario podrá ajustar el número de términos de la serie para observar cómo mejora la aproximación conforme se incluyen más términos en la suma.
- **Interfaz intuitiva y amigable:**
 - La aplicación ofrecerá una interfaz sencilla y accesible, en la que el usuario podrá ingresar las funciones, definir los intervalos y elegir el tipo de serie de Fourier que desea calcular. La experiencia estará diseñada para minimizar la curva de aprendizaje, permitiendo a usuarios sin experiencia en programación obtener resultados rápidamente.

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar un prototipo de una aplicación web capaz de calcular las series de Fourier en su forma trigonométrica o exponencial compleja de una función continua en un intervalo o definida a trozos, así como ser capaz de obtener la extensión par o impar de dicha función y, finalmente, graficar tanto la función original como la función expandida como una serie de Fourier y poder añadir o eliminar términos de dicha forma para apreciar como esta se aproxima a la función original.

1.5.2. Objetivos Específicos

- Implementar una interfaz de usuario que permita ingresar la función continua o a trozos, seleccionar intervalos y definir el tipo de serie de Fourier (trigonométrica o exponencial) o tipo de expansión (par o impar).
- Implementar los módulos para el cálculo de coeficientes de la serie trigonométricas y serie exponencial compleja, adaptándose a funciones continuas o definidas a trozos.
- Desarrollar un módulo de visualización que permita graficar tanto la función original como la aproximación de la serie de Fourier. Este módulo debería incluir opciones para añadir o eliminar términos y visualizar cómo estas modificaciones afectan la aproximación a la función original

1.6. Metodología de la Investigación

Para el desarrollo de este proyecto se utilizará la metodología por prototipos...

1.7. Estructura del documento técnico

Al final voy a detallar como es que está estructurado todo el documento del TT

CAPÍTULO 2

Estado del Arte

Para la correcta comprensión del trabajo presente, se necesita conocer el estado actual de las herramientas y tecnologías disponibles para el cálculo y la visualización de series de Fourier, así como los estudios y proyectos previos que abordan la implementación de soluciones similares. En este sentido, se revisarán diversas plataformas de uso común que permiten realizar estos procesos de manera separada, además de destacar trabajos académicos relacionados que aportan al desarrollo de herramientas educativas y matemáticas interactivas. Esta revisión permitirá contextualizar la propuesta de una aplicación web que integre ambas funcionalidades en una sola plataforma.

2.1. Herramientas y Tecnologías Actuales

En esta sección se presentarán las principales herramientas tecnológicas utilizadas para el cálculo de series de Fourier y su visualización gráfica. Podemos verlos en la siguiente tabla:

SOFTWARE	CARACTERÍSTICAS	PRECIO
Wolfram Alpha	Se trata de un potente motor comercial de conocimiento computacional para cálculos matemáticos y gráficos. Contiene módulos que permiten el calculo de los coeficientes de Series de Fourier (Trigonometría y compleja) así como las extensiones pares e impares para funciones simples o a trozos, así como poder expandir la serie y obtener una gráfica estática. [2]	Desde MXN \$1,200.00 anuales para estudiantes, plan gratuito limitado.

Continúa en la siguiente página

Tabla 2.1 – continuación

SOFTWARE	CARACTERÍSTICAS	PRECIO
Geogebra	Herramienta dinámica para construcciones geométricas y gráficas. Podemos graficar cualquier serie o extensión trigonométrica y poder ver como cambia la serie conforme añadimos más coeficientes. Requiere que hagamos los cálculos y armar la expresión de la serie [3]	Software libre y código abierto
Desmos	Similar a Geogebra es una calculadora gráfica en línea para cálculos y gráficos, donde de igual modo podemos graficar cualquier serie o extensión trigonométrica y poder ver como cambia la serie conforme añadimos más coeficientes. Requiere cálculos previos para series de Fourier. [4]	Software libre y código abierto
Python _{Manim}	Librería de animación en Python para visualizaciones matemáticas, incluida la animación de series de Fourier. Podemos graficar cualquier serie ya sea trigonométrica o exponencial y poder animar de infinitas maneras el como se aproxima la serie con sus coeficientes. Requiere hacer los cálculos, armar la expresión de la serie y tener conocimientos en Python, programación orientada a objetos y a librería de Manim [5]	Software libre y código abierto
Python _{SymPy}	Es una biblioteca de Python para realizar matemáticas simbólicas, que incluye el cálculo de series de Fourier. A menudo se usa en combinación con librerías de visualización como Matplotlib para representar gráficamente los resultados, pero de nuevo, esta combinación requiere conocimientos de programación. [6]	Software libre y código abierto

Continúa en la siguiente página

Tabla 2.1 – continuación

SOFTWARE	CARACTERÍSTICAS	PRECIO
Matlab	Es un entorno de programación comercial para cálculos numéricos y visualización de datos, con herramientas específicas para series de Fourier, además de tener funciones para poder graficarlas. Requiere conocimientos en matlab. [7]	Desde USD\$99 (aprox. MXN\$1627.82) anuales para estudiantes.

Tabla 2.1: Comparación de software para cálculos matemáticos y visualización de datos

2.1.1. Comparativa del Funcionamiento de las Herramientas

A continuación, se procederá a calcular la serie de Fourier para la función $f(x) = x$ en el intervalo de $-\pi$ a π utilizando cada una de las herramientas previamente descritas. El cálculo se realizará en su **forma trigonométrica** y, en los casos en que la herramienta lo permita, también se obtendrá la **forma exponencial compleja**. Asimismo, se graficará la serie de Fourier en las plataformas que lo permitan, lo que nos permitirá comparar tanto el proceso como los resultados obtenidos en cada herramienta.

Esta comparación servirá para identificar las capacidades, ventajas y limitaciones de cada una de las plataformas en el contexto del cálculo y la visualización de series de Fourier, evaluando también su facilidad de uso y precisión en la representación gráfica.

Primeramente, calcularemos la serie nosotros para hacer una comparativa con los resultados dados por los softwares:

Los coeficientes de Fourier para la función $f(x) = x$, en el intervalo $[-\pi, \pi]$, son los siguientes (ver Apéndice A para los cálculos completos):

- ****Forma Trigonométrica****:

$$a_0 = 0, \quad a_n = 0, \quad b_n = \frac{2(-1)^{n+1}}{n}$$

Por lo tanto, la serie trigonométrica de Fourier para $f(x) = x$ es:

$$f(x) = 2 \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \sin(nx)$$

- ****Forma Exponencial Compleja****:

$$c_n = \frac{i(-1)^n}{n}, \quad \text{para } n \neq 0.$$

Por lo tanto, la serie exponencial compleja de Fourier para $f(x) = x$ es:

$$f(x) = \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \frac{i(-1)^n}{n} e^{inx}$$

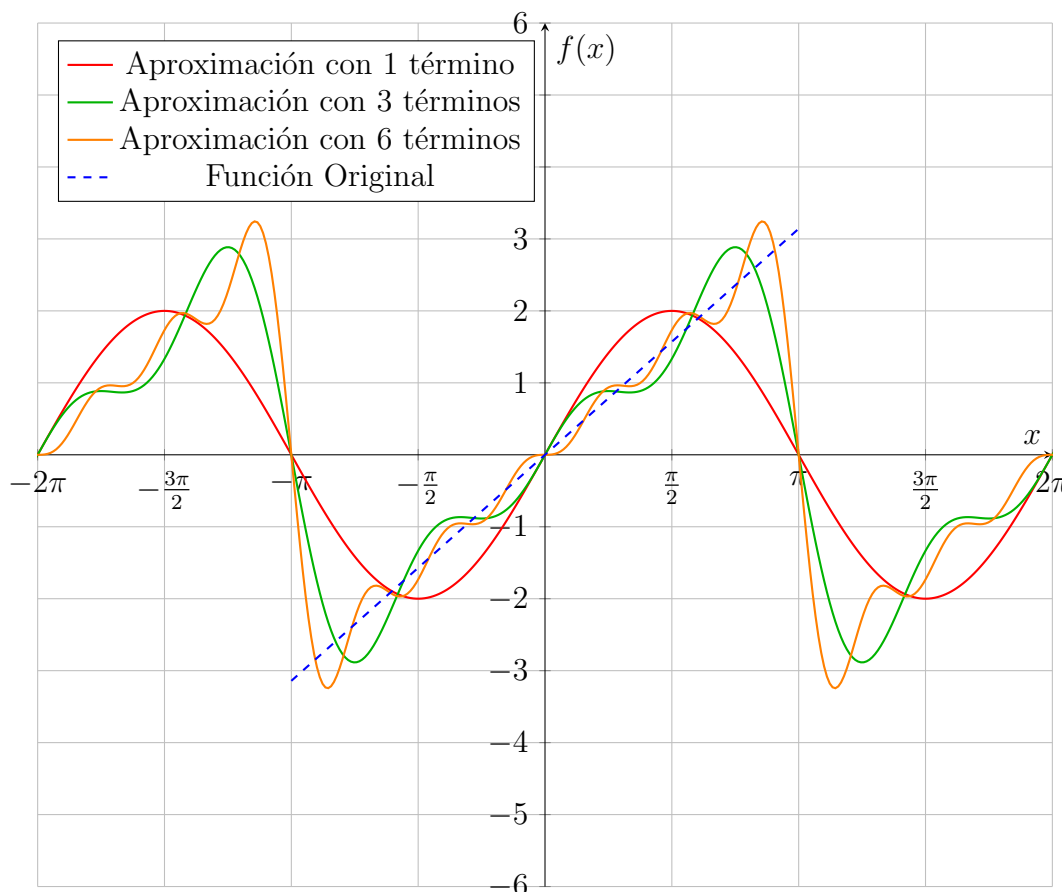


Figura 2.1: Gráfica de los coeficientes de Fourier calculados en el Apendice A

Podemos ver en la Figura 2.1, la gráfica de la función y sus aproximaciones de Fourier, donde estas aproximaciones son equivalentes entre la serie trigonométrica y la serie exponencial

2.1.1.1. Cálculo de Serie de Fourier con Wolfram Alpha

WolframAlpha nos permite calcular series de Fourier trigonométricas y exponenciales complejas, además de que nos permite calcular extensiones de medio rango periódicas pares, impares y periódicas. Para nuestra primer prueba, calcularemos la serie de Fourier trigonométrica para la función $f(x) = x$ en el intervalo de $-\pi$ a π , para hacerlo usaremos la función *FourierTrigSeries(exp, t, n)* de WolframAlpha, que nos permite calcular la serie trigonométrica de Fourier de la función *exp* respecto a la variable *t* obteniendo la serie hasta el término *n*.

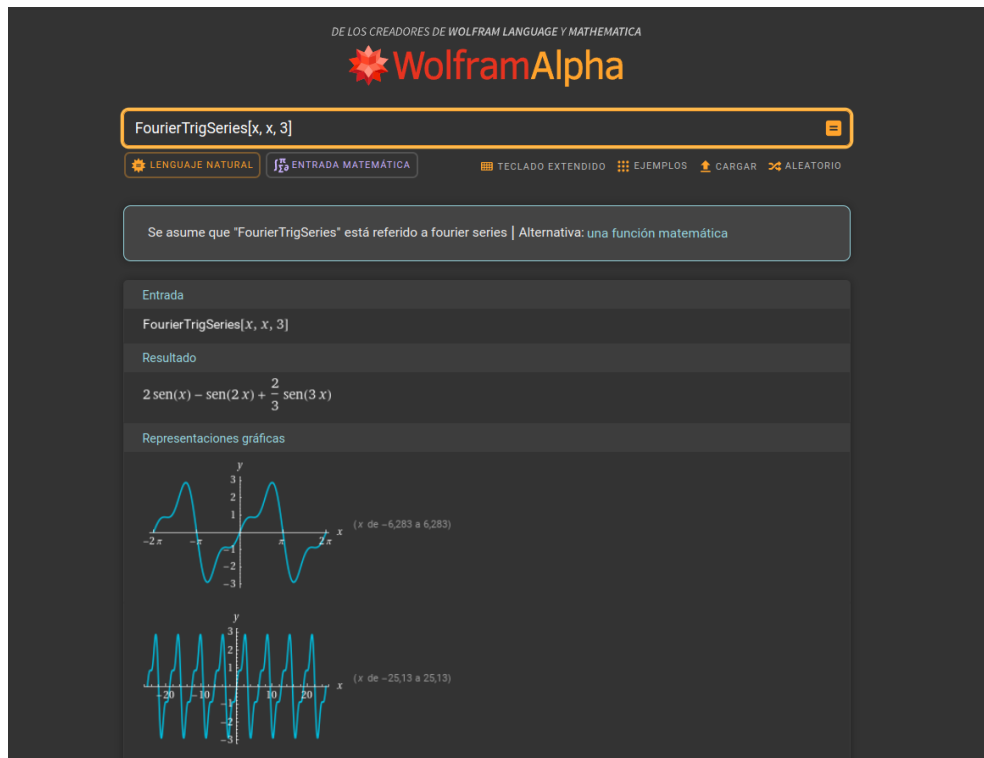


Figura 2.2: Calculo de una serie trigonométrica de Fourier en WolframAlpha

Ahora usaremos la función de $FourierSeries(exp, t, n)$ de WolframAlpha, esta función no permite calcular la serie exponencial compleja de Fourier de la función exp respecto a la variable t obteniendo la serie hasta el término n .

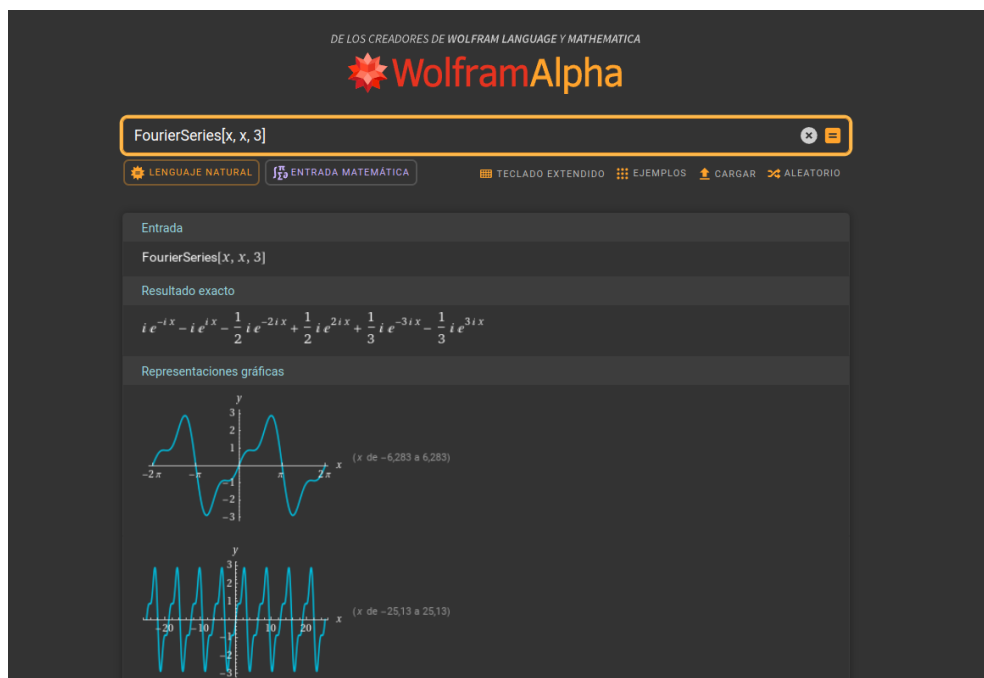


Figura 2.3: Calculo de una serie trigonométrica de Fourier en WolframAlpha

Wolfram nos devuelve la expansión de cada serie en su forma mas reducida, ademas de dos pequeñas gráficas de nuestra aproximación vista desde no nos

devuelve los coeficientes de la serie ni tampoco su expresión final en notación de serie.

2.1.1.2. Cálculo de Serie de Fourier con MATLAB

Matlab

2.1.1.3. Cálculo de Serie de Fourier con GeoGebra

Aquí se describirá cómo se realiza el cálculo y la graficación de la serie de Fourier utilizando GeoGebra...

2.1.1.4. Cálculo de Serie de Fourier con Python (Matplotlib/Manim)

Aquí se describirá cómo se realiza el cálculo y la graficación de la serie de Fourier utilizando Python con las librerías Matplotlib o Manim...

2.2. Trabajos y Proyectos Relacionados

En esta sección se analizarán trabajos académicos y proyectos de investigación que han desarrollado herramientas similares o que abordan la enseñanza de series de Fourier y el uso de plataformas interactivas para la educación matemática.

CAPÍTULO 3

Marco Teórico

En este capítulo...

3.1. Origen e Historia

Uno de los problemas del que se ocuparon los matemáticos del siglo XVIII es el que se conoce con el nombre del “problema de la cuerda vibrante”. Este problema fue estudiado por d’Alembert y Euler (usando el método de propagación de las ondas) y un poco más tarde, concretamente en 1753, por Daniel Bernoulli. La solución dada por este difería de la proporcionada por los anteriores y consistió básicamente en expresar la solución del problema como superposición (en general infinita) de ondas sencillas.

Las ideas de Bernoulli fueron aplicadas y perfeccionadas por Fourier, en 1807, en el estudio de problemas relacionados con la conducción del calor. Quedaron plasmadas por escrito en el libro clásico *Théorie analytique de la chaleur*, publicado en 1822. Los razonamientos realizados por Fourier en este libro plantearon de manera inmediata numerosas controversias y cuestiones que han tenido una influencia significativa en la historia de la Matemática.

3.1.1. El problema de la cuerda oscilante

Uno de los problemas más interesantes que abordaron los científicos del siglo XVIII, y que aparece con frecuencia en problemas físicos relacionados con procesos oscilatorios, es el conocido como “problema de la cuerda vibrante”. Este se puede describir en su forma más elemental de la siguiente manera: Supongamos que tenemos una cuerda flexible y tensa, cuyos extremos están fijos, convenientemente, en los puntos $(0, 0)$ y $(\pi, 0)$ sobre el eje horizontal. Si la cuerda se tira de modo que su forma inicial corresponde a la curva definida por $y = f(x)$, y luego se suelta, la pregunta es: ¿Cuál será el movimiento resultante de la cuerda? Los desplazamientos de la cuerda siempre se encuentran en un mismo plano, y el vector desplazamiento es perpendicular en cualquier momento. Para describir este movimiento se utiliza una función $u(x, t)$, donde $u(x, t)$ representa el desplazamiento vertical de la cuerda en la posición x (con $0 \leq x \leq \pi$) y en el instante t (con $t \geq 0$). El problema es determinar $u(x, t)$ a partir de $f(x)$.

El primer matemático que propuso un modelo adecuado para este problema fue Jean Le Rond d’Alembert. Bajo diversas hipótesis (asumiendo, por ejemplo,

que las vibraciones son "pequeñas"), d'Alembert demostró en 1747 (*Hist. de l'Acad. de Berlin*, 3, 1747, pp. 214-219) que la función $u(x, t)$ debe satisfacer las siguientes condiciones:

$$\begin{aligned} \frac{\partial^2 u(x, t)}{\partial t^2} &= \frac{\partial^2 u(x, t)}{\partial x^2}, & 0 < x < \pi, & t > 0 \\ u(x, 0) &= f(x), & 0 \leq x \leq \pi \\ \frac{\partial u(x, 0)}{\partial t} &= 0, & 0 \leq x \leq \pi \\ u(0, t) &= u(\pi, t) = 0, & t \geq 0. \end{aligned} \quad (3.1)$$

La primera de estas condiciones es una ecuación en derivadas parciales de segundo orden, conocida como la *Ecuación de Ondas*. La segunda ecuación establece la posición inicial de la cuerda, mientras que la tercera indica que la velocidad inicial de la cuerda es cero (recordando que, una vez desplazada a la posición $f(x)$, la cuerda es liberada). La última condición expresa que, para cualquier tiempo, los extremos de la cuerda permanecen fijos. En la figura 3.1 La variable $u = u(x, t)$ mide el desplazamiento sobre la vertical a tiempo $t > 0$ en la posición $x \in [0, \pi]$.

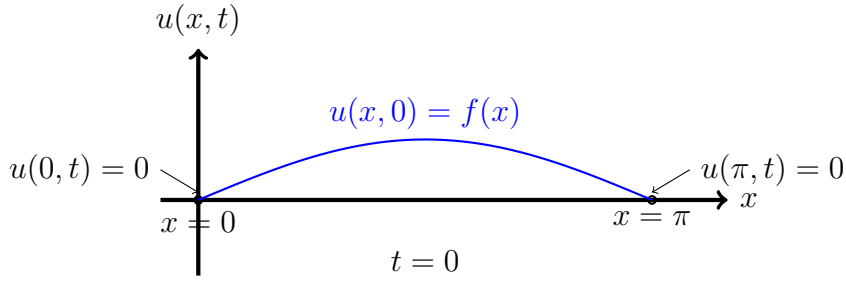


Figura 3.1: Ilustración del problema de la cuerda vibrante.

D'Alembert demostró también que la solución de 3.1 viene dada por

$$u(x, t) = \frac{1}{2} [\tilde{f}(x+t) + \tilde{f}(x-t)] \quad (3.2)$$

donde \tilde{f} es "una extensión conveniente de la función f ."

De manera más precisa, y con el lenguaje de hoy en día, el teorema de d'Alembert ([25], [27]) confirmó que la posición futura de la cuerda está completamente determinada por su posición inicial:

TEOREMA 2.1. *Sea $f \in C^2[0, \pi]$, tal que $f(0) = f(\pi) = f''(0^+) = f''(\pi^-) = 0$. Entonces (1.1) tiene una única solución $u \in C^2(\Omega) \cap C^1(\bar{\Omega})$, donde $\Omega = (0, \pi) \times (0, +\infty)$. Además u viene dada por la fórmula (1.2), donde \tilde{f} es la extensión a \mathbb{R} (conjunto de los números reales), impar y 2π -periódica de la función f .*

No se conoce con exactitud la manera en que d'Alembert llegó a la fórmula 3.2, pero un camino muy probable pudo ser el siguiente: el cambio de variables

$$\xi = x + t, \quad \mu = x - t,$$

transforma la ecuación

$$\frac{\partial^2 u(x, t)}{\partial t^2} = \frac{\partial^2 u(x, t)}{\partial x^2},$$

en otra más simple:

$$\frac{\partial^2 u}{\partial \xi \partial \mu} = 0.$$

Las soluciones de esta última ecuación son trivialmente de la forma

$$u(\xi, \mu) = F(\xi) + G(\mu).$$

Combinando esta última relación con las otras condiciones en 3.1, se puede llegar a intuir la forma (1.2) que tiene la solución de 3.1.

La interpretación física de la solución dada por *d'Alembert* es muy interesante: La función

$$\frac{1}{2}f(x + t)$$

representa una onda (una solución de la ecuación de ondas) que se desplaza hacia la izquierda con velocidad 1. Análogamente, la función

$$\frac{1}{2}f(x - t)$$

representa otra onda que se desplaza hacia la derecha con velocidad 1. De este modo, la solución de 3.1 es la superposición de dos ondas, una de las cuales se desplaza hacia la izquierda y otra hacia la derecha, ambas con velocidad 1. Por este motivo se dice que la fórmula 3.2 se ha obtenido usando el *Método de propagación de las ondas*.

Otra manera de obtener la solución del problema 3.1, completamente distinta de la vista anteriormente, fue propuesta por *Daniel Bernoulli* en 1753 (*Hist. de l'Acad. de Berlin*, 9, 1753, 147-172; 173-195). La idea clave es obtener la solución de 3.1 como superposición de ondas más sencillas, concretamente aquellas que son de la forma

$$u_n(x, t) = \sin(nx) \cos(nt), \quad \forall n \in \mathbb{N}, \quad (3.3)$$

donde \mathbb{N} es el conjunto de los números naturales. Para cada tiempo t fijo, la anterior función es un múltiplo de la función $\sin(nx)$, que se anula exactamente en $n - 1$ puntos del intervalo $(0, \pi)$. Así, si pudiésemos observar la vibración de la cuerda correspondiente a las ondas u_n , tendríamos $n - 1$ puntos, llamados nodos, en los que la cuerda se mantendría constantemente fija en el eje de abscisas (como en los extremos del intervalo $[0, \pi]$). Entre dichos nodos, la cuerda oscilaría de acuerdo con 3.3.

¿Cómo concibió *Bernoulli* la idea anterior? Parece ser que una posibilidad es que usase sus conocimientos musicales. Para ello se basó en que el sonido que emite una cuerda vibrante es, en general, superposición de armónicos, es decir,

superposición de funciones de la forma $u_n(x, t)$. Tales funciones representan, para $n = 1$ el tono fundamental y para $n > 1$ sus armónicos, y desde el punto de vista musical se corresponden con los tonos puros. Así, *Bernoulli* afirmó que cualquier sonido que produjese la vibración de la cuerda debe ser superposición de tonos puros.

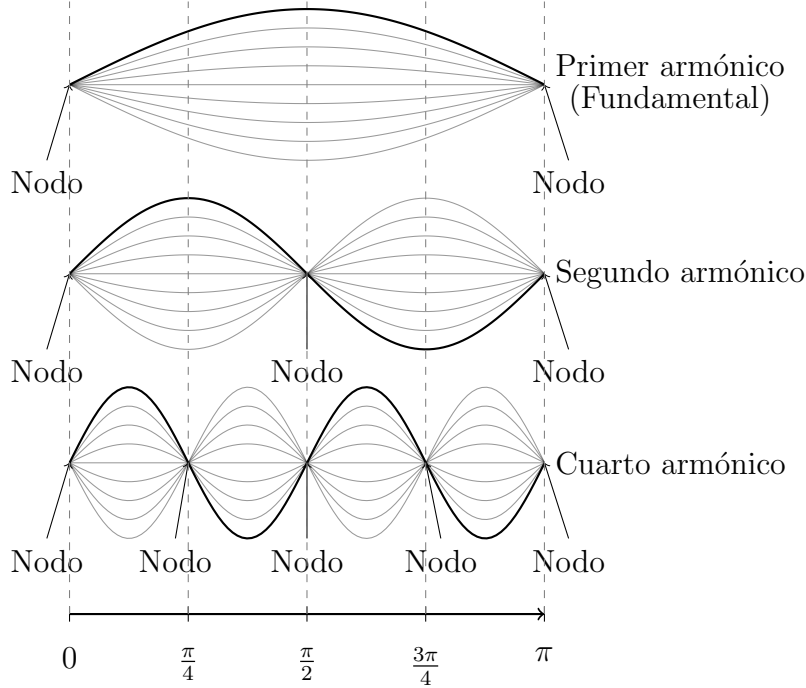


Figura 3.2: Ilustración de 3 armónicos y sus nodos de la cuerda al vibrar.

Desde el punto de vista matemático, ello significa que la solución de 3.1 debe poder representarse de la forma:

$$u(x, t) = \sum_{n=1}^{\infty} a_n \sin(nx) \cos(nt) \quad (3.4)$$

donde los coeficientes a_n han de elegirse adecuadamente para que se satisfaga 3.1.

Ahora bien, f no es, en general de la forma justo mencionada, pero, y ahora viene la pregunta clave: ¿Será posible obtener la solución de 3.1, para cualquier f dada, como superposición de las anteriores soluciones sencillas u_n ? Esto es lo que propuso *Bernoulli*.

Si la solución propuesta por *Bernoulli* fuese correcta, ello obligaría a que

$$u(x, 0) = \sum_{n=1}^{\infty} a_n \sin(nx) \quad (3.5)$$

y por tanto a que

$$f(x) = \sum_{n=1}^{\infty} a_n \sin(nx), \quad \forall x \in [0, \pi], \quad (3.6)$$

para “una adecuada elección de los coeficientes a_n ”.

3.1.2. La ecuación de calor

3.2. Series de Fourier

3.2.1. Series de Funciones

3.2.1.1. Definición de Series de Funciones

3.2.1.2. Convergencia de Series de Funciones

3.2.2. Definición de las Funciones Trigonométricas de Seno y Coseno

3.2.2.1. Definición del Seno

3.2.2.2. Definición del Coseno

3.2.2.3. Propiedades y Relaciones Fundamentales

3.2.3. Funciones Periódicas

3.2.3.1. Definición de Funciones Periódicas

3.2.3.2. Propiedades de las Funciones Periódicas

3.2.4. Armónicos

3.2.4.1. Definición de Armónicos

3.2.4.2. Importancia de los Armónicos en Series de Fourier

3.2.5. Funciones Ortogonales

3.2.5.1. Concepto de Ortogonalidad

3.2.5.2. Ejemplos de Funciones Ortogonales

3.2.6. Originalidad del Seno y Coseno

3.2.6.1. Propiedades del Seno y el Coseno

3.2.6.2. Uso del Seno y el Coseno en Series de Fourier

3.2.7. Series de Fourier de Período T o de $-p$ a $-p$

3.2.7.1. Generalización de la Serie de Fourier

3.2.7.2. Coeficientes de Fourier para Período T

3.2.8. Funciones Suaves y Suaves por Partes

3.2.8.1. Definición de Funciones Suaves

3.2.8.2. Definición de Funciones Suaves por Partes

3.2.8.3. Aplicaciones en Series de Fourier

3.2.9. Series de Fourier de Senos y Cosenos

3.2.9.1. Condiciones para la Cancelación de Coeficientes

3.2.9.2. Desarrollo de Series de Fourier con Senos y Cosenos

3.2.13.2. Aplicaciones y Ejemplos**3.3. Tecnologías Utilizadas****3.3.1. HTML, CSS y JavaScript****3.3.1.1. HTML**

Explicación sobre cómo define la estructura de la aplicación web.

3.3.1.2. CSS

Justificación para su uso en el diseño y estilo visual de la aplicación.

3.3.1.3. JavaScript

Detalle del uso de JavaScript para la interacción dinámica de la interfaz y control de eventos.

3.3.2. NodeJS**3.3.2.1. Introducción a NodeJS**

Explicación básica sobre NodeJS.

3.3.2.2. Ventajas del uso de NodeJS para aplicaciones web

Descripción de por qué se eligió NodeJS para la parte del servidor.

3.3.2.3. Integración de NodeJS con Maxima

Detalle de cómo se usa NodeJS para comunicarse con Maxima y realizar cálculos matemáticos.

3.3.3. Maxima**3.3.3.1. Qué es Maxima y por qué se usa para cálculos simbólicos**

Explicación sobre el software Maxima y su importancia en los cálculos de Series de Fourier.

3.3.3.2. Implementación de Maxima en el servidor

Cómo se utiliza Maxima en el servidor junto con NodeJS para realizar cálculos matemáticos de forma eficiente.

3.3.4. Angular (Posible implementación futura)**3.3.4.1. Introducción a Angular**

Explicación básica sobre el framework Angular.

3.3.4.2. Ventajas del uso de Angular para futuras versiones del proyecto

Justificación sobre la posibilidad de usar Angular en futuras versiones del proyecto.

3.3.4.3. Consideraciones para su implementación

Consideraciones técnicas y cómo Angular podría mejorar la estructura de la aplicación.

3.3.5. Metodologías de Desarrollo de Software

-

3.3.5.1. Desarrollo Ágil

Breve descripción de la metodología ágil aplicada al desarrollo de la aplicación. -

3.3.5.2. Fases del Proyecto

Descripción de las fases de planificación, desarrollo y pruebas que seguiste.

3.3.6. Usabilidad y Experiencia de Usuario (UX)

-

3.3.6.1. Principios de Usabilidad

Breve descripción de los principios que seguiste para garantizar una buena usabilidad de la aplicación. -

3.3.6.2. Diseño de la Interfaz de Usuario (UI)

Descripción de cómo el diseño de la interfaz facilita la interacción de los usuarios con las Series de Fourier.

3.3.7. Seguridad en Aplicaciones Web

-

3.3.7.1. Protección contra ataques comunes

Breve descripción de las medidas tomadas para evitar ataques como la inyección de código y el cross-site scripting (XSS). -

3.3.7.2. Autenticación y manejo seguro de datos

Si tu aplicación maneja datos sensibles o autenticación de usuarios, menciona las técnicas utilizadas para asegurar los datos.

CAPÍTULO 4

Análisis y Diseño del Sistema

En este capítulo se evaluaron los datos del marco teórico para seleccionar los componentes más adecuados para el proyecto. También se realizó un análisis detallado de los requerimientos y riesgos asociados al desarrollo del sistema, asegurando un diseño eficiente y viable para su implementación.

4.1. Metodología de Desarrollo

La metodología a usar

4.1.1. Descripción de la Metodología

aca decimos que rollo con la metodología

4.1.2. Ventajas de la Metodología Elegida

Está perrona

4.2. Análisis de Requerimientos

Los requerimientos funcionales y no funcionales descritos en este documento son fundamentales para el desarrollo y éxito de la aplicación web de cálculo y visualización de Series de Fourier. Estos requerimientos definen no solo las funcionalidades que la aplicación debe cumplir, sino también las expectativas de rendimiento, seguridad y usabilidad. A través de su análisis, se asegura que el sistema propuesto sea capaz de resolver la problemática planteada de manera eficaz, proporcionando a los usuarios una herramienta confiable y eficiente para el análisis de series matemáticas.

4.2.1. Requerimientos Funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Describen las interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúa el sistema.

Requerimiento	Ingreso de Funciones Matemáticas
Identificador	RF1
Prioridad de desarrollo	Alta
Entrada	Funciones matemáticas ingresadas por el usuario usando MathQuill
Salida	Funciones ingresadas en formato compatible con Maxima
Descripción	El usuario debe poder ingresar funciones matemáticas definidas en un solo trozo o en múltiples trozos usando una notación matemática intuitiva. Debe soportar funciones trigonométricas, polinomios, exponenciales y cualquier otra función que Maxima pueda interpretar.
Precondición	El usuario debe estar en el área de ingreso de funciones.
Postcondición	Las funciones son ingresadas correctamente y listas para su procesamiento.

Tabla 4.1: Requerimiento funcional No. 1

Requerimiento	Selección de Parámetros
Identificador	RF2
Prioridad de desarrollo	Alta
Entrada	Parámetros seleccionados por el usuario (tipo de serie, período T)
Salida	Parámetros listos para el cálculo de coeficientes
Descripción	Los usuarios deben poder seleccionar entre las diferentes representaciones de Series de Fourier y especificar el período T de la función, ya sea de $-T/2$ a $T/2$ o de 0 a T .
Precondición	El usuario debe haber ingresado una función matemática y estar en el área de selección de parámetros.
Postcondición	Los parámetros seleccionados están listos para el cálculo de coeficientes.

Tabla 4.2: Requerimiento funcional No. 2

Requerimiento		Cálculo de Coeficientes
Identificador		RF3
Prioridad de desarrollo		Alta
Entrada		Función matemática y parámetros seleccionados por el usuario
Salida		Coeficientes de la serie de Fourier calculados
Descripción		La aplicación debe calcular automáticamente los coeficientes de la serie de Fourier según la función ingresada y los parámetros especificados, enviando la información a Maxima y recibiendo los resultados.
Precondición		El usuario debe haber ingresado una función y seleccionado los parámetros necesarios.
Postcondición		Los coeficientes de la serie de Fourier se calculan y se devuelven al sistema.

Tabla 4.3: Requerimiento funcional No. 3

Requerimiento		Interactividad en la Gráfica
Identificador		RF4
Prioridad de desarrollo		Media
Entrada		Control del usuario sobre la gráfica (acercar, alejar, cantidad de términos n)
Salida		Gráfica interactiva modificada según la selección del usuario
Descripción		El usuario debe poder acercar, alejar y desplazar la gráfica, así como seleccionar la cantidad de términos n de la suma parcial a visualizar.
Precondición		El usuario debe haber ingresado la función y los parámetros.
Postcondición		La gráfica es ajustada interactivamente según la entrada del usuario.

Tabla 4.4: Requerimiento funcional No. 4

Requerimiento	Visualización de las Gráficas
Identificador	RF5
Prioridad de desarrollo	Alta
Entrada	Función original y términos de la Serie de Fourier
Salida	Gráficas de la función original y las sumas parciales de la Serie de Fourier
Descripción	El usuario debe poder visualizar la gráfica de la función original y las sumas parciales de la Serie de Fourier en un lienzo canvas, que se actualizará en tiempo real a medida que se agregan o quitan términos.
Precondición	El usuario debe haber ingresado la función y los parámetros necesarios.
Postcondición	La gráfica se actualiza en tiempo real según los términos seleccionados.

Tabla 4.5: Requerimiento funcional No. 5

Requerimiento	Visualización de resultados
Identificador	RF6
Prioridad de desarrollo	Alta
Entrada	Función matemática y parámetros seleccionados
Salida	Expresión matemática de la serie de Fourier y coeficientes individuales
Descripción	La aplicación debe mostrar la serie de Fourier y los valores individuales de los coeficientes en notación matemática, utilizando MathQuill.
Precondición	El usuario debe haber ingresado la función y los parámetros necesarios.
Postcondición	Se muestran la serie de Fourier y los coeficientes en notación matemática.

Tabla 4.6: Requerimiento funcional No. 6

Requerimiento		Manejo de Errores
Identificador		RF7
Prioridad de desarrollo		Media
Entrada		Función ingresada por el usuario, parámetros seleccionados
Salida		Mensajes de error claros en caso de problemas con la función o los parámetros
Descripción		La aplicación debe detectar errores en la entrada de funciones o parámetros no válidos y mostrar mensajes de error claros. También debe notificar si las integrales no convergen o si Maxima no puede procesar la función.
Precondición		El usuario ha ingresado una función o parámetros y ha intentado procesar los datos.
Postcondición		Se muestra un mensaje de error claro si ocurre algún problema.

Tabla 4.7: Requerimiento funcional No. 7

Requerimiento		Optimización de Resultados
Identificador		RF8
Prioridad de desarrollo		Baja
Entrada		Parámetros y resultados de coeficientes previos
Salida		Optimización de cálculo y presentación de resultados mejorada
Descripción		La aplicación debe optimizar el cálculo de los coeficientes para mejorar el rendimiento y la experiencia del usuario, manejando grandes cantidades de términos de manera eficiente.
Precondición		El usuario ha ingresado parámetros complejos o grandes cantidades de términos.

Postcondición	Los resultados optimizados se calculan y presentan rápidamente.
----------------------	---

Tabla 4.8: Requerimiento funcional No. 8

4.2.2. Requerimientos No Funcionales

Los requerimientos no funcionales describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema. Describen atributos sólo del sistema o del ambiente del sistema que no están relacionados directamente con los requisitos funcionales. Los requisitos no funcionales incluyen restricciones cuantitativas, tipo de plataforma.

Nombre	Descripción
Rendimiento y Escalabilidad	El sistema debe calcular los coeficientes de Fourier en un tiempo razonable, incluso para funciones con múltiples trozos o con un alto número de términos en la serie. La visualización gráfica debe ser fluida y responsiva, incluso al agregar o quitar muchos términos de la serie. Las interacciones en el lienzo canvas deben realizarse sin latencia perceptible.
Interfaz de Usuario Amigable e Intuitiva	La interfaz debe ser intuitiva y fácil de usar, con botones y controles claramente identificados. Los mensajes de error y advertencia deben ser comprensibles, evitando el uso de jerga técnica complicada. La notación matemática debe ser clara y precisa, usando MathQuill para entradas y visualización de resultados.
Manejo de la Concurrency	La aplicación debe soportar múltiples usuarios concurrentes enviando cálculos al backend sin generar bloqueos ni conflictos en las llamadas a Maxima.
Portabilidad y Compatibilidad	La aplicación debe ser compatible con los principales navegadores modernos (Chrome, Firefox, Edge). Aunque no es recomendable usar la aplicación en dispositivos móviles, el diseño debe ser responsivo para ajustarse a diferentes tamaños de pantalla, incluidos dispositivos móviles y tabletas.

Modularidad Extensibilidad Código	y del	El frontend debe estar diseñado para facilitar la migración a frameworks como Angular en el futuro. El código debe ser modular y seguir buenas prácticas de desarrollo para facilitar la adición de nuevas características. Aunque el sistema no requiere una base de datos en su versión actual, el backend debe ser fácilmente extensible para permitir el almacenamiento de sesiones o configuraciones del usuario en el futuro.
Seguridad Aplicación	de la	Las peticiones a la API deben estar protegidas contra inyecciones de código. Implementar seguridad básica en el backend para evitar solicitudes maliciosas o repetitivas que puedan afectar el rendimiento del servicio.
Mantenibilidad Código	de	La aplicación debe tener una guía de usuario clara para explicar cómo ingresar funciones, seleccionar series y usar la interfaz. El código debe estar bien documentado para que futuros desarrolladores puedan entender la lógica y estructura de la aplicación fácilmente.

Tabla 4.9: Tabla de requerimientos no funcionales

4.3. Análisis de Riesgos

4.3.1. Identificación de Riesgos

4.3.2. Jerarquización de Riesgos

4.4. Estimación de Costos

4.4.1. Método de Estimación de Costos

4.4.2. Detalle de los Costos Estimados

4.5. Diseño del Sistema

4.5.1. Arquitectura General del Sistema

4.5.1.1. Arquitectura Cliente-Servidor

4.5.1.2. Flujo de Datos

4.5.2. Diseño de la Interfaz de Usuario

4.5.2.1. Estructura de la Interfaz

4.5.2.2. Prototipo de la Interfaz

CAPÍTULO 5

Implementación (Primer Prototipo)

En este capítulo vamos a juntar todo lo primero

APÉNDICE A

Cálculo de los Coeficientes de la Serie de Fourier

Dada la función $f(x) = x$, vamos a calcular los coeficientes correspondientes a su serie de Fourier en el intervalo $[-\pi, \pi]$.

A.1. Forma Trigonométrica

La serie trigonométrica de Fourier para una función $f(x)$ está dada por la siguiente expresión:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 x) + b_n \sin(n\omega_0 x))$$

Donde los coeficientes a_0 , a_n y b_n se definen como:

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} f(x) dx$$

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(x) \cos(n\omega_0 x) dx$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(x) \sin(n\omega_0 x) dx$$

Donde $\omega_0 = \frac{2\pi}{T}$, y en este caso, $T = 2\pi$, por lo que $\omega_0 = 1$.

Calculamos cada uno de los coeficientes para $f(x) = x$:

- El coeficiente a_0 es:

$$a_0 = \frac{2}{2\pi} \int_{-\pi}^{\pi} x dx = \frac{1}{\pi} \left[\frac{x^2}{2} \right]_{-\pi}^{\pi} = \frac{1}{\pi} \left(\frac{\pi^2}{2} - \frac{(-\pi)^2}{2} \right) = 0$$

Por lo tanto, $a_0 = 0$.

- El coeficiente a_n es:

Dado que $x \cos(nx)$ es una función impar, ya que x es impar y $\cos(nx)$ es par, su integral en un intervalo simétrico es cero:

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} x \cos(nx) dx = 0$$

- El coeficiente b_n es:

Dado que $x \sin(nx)$ es una función par (producto de dos funciones impares), podemos calcular:

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} x \sin(nx) dx = \frac{2}{\pi} \int_0^{\pi} x \sin(nx) dx$$

Aplicamos integración por partes, tomando $u = x$ y $dv = \sin(nx) dx$, lo que nos da $du = dx$ y $v = -\frac{1}{n} \cos(nx)$. Entonces:

$$\int_0^{\pi} x \sin(nx) dx = -\frac{x}{n} \cos(nx) \Big|_0^{\pi} + \frac{1}{n} \int_0^{\pi} \cos(nx) dx$$

Evaluamos el primer término:

$$-\frac{\pi}{n} \cos(n\pi) + 0 = -\frac{\pi}{n} (-1)^n$$

La integral restante es:

$$\frac{1}{n} \int_0^{\pi} \cos(nx) dx = \frac{1}{n} \left[\frac{\sin(nx)}{n} \right]_0^{\pi} = \frac{1}{n^2} (0 - 0) = 0$$

Por lo tanto, el coeficiente b_n es:

$$b_n = \frac{2}{\pi} \left(-\frac{\pi}{n} (-1)^n \right) = \frac{2(-1)^{n+1}}{n}$$

Entonces, la serie trigonométrica de Fourier para $f(x) = x$ es:

$$f(x) = 2 \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \sin(nx)$$

A.2. Forma Exponencial Compleja

La serie exponencial compleja de Fourier está dada por la siguiente expresión:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_0 x}$$

Donde los coeficientes c_n se definen como:

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(x) e^{-in\omega_0 x} dx$$

Calculamos el coeficiente c_n para $n \neq 0$:

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} x e^{-inx} dx$$

Aplicamos integración por partes con $u = x$ y $dv = e^{-inx} dx$, obteniendo $du = dx$ y $v = \frac{e^{-inx}}{-in}$. Entonces:

$$c_n = \frac{1}{2\pi} \left(\left[x \frac{e^{-inx}}{-in} \right]_{-\pi}^{\pi} - \int_{-\pi}^{\pi} \frac{e^{-inx}}{-in} dx \right)$$

El primer término es:

$$\left[x \frac{e^{-inx}}{-in} \right]_{-\pi}^{\pi} = \frac{\pi e^{-in\pi} - (-\pi) e^{in\pi}}{-in} = \frac{2\pi(-1)^n}{-in}$$

El segundo término es cero, ya que:

$$\int_{-\pi}^{\pi} e^{-inx} dx = 0$$

Por lo tanto:

$$c_n = \frac{1}{2\pi} \left(\frac{2\pi(-1)^n}{-in} \right) = \frac{i(-1)^n}{n}$$

Para $n = 0$:

$$c_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} x dx = 0$$

Así que la serie exponencial compleja de Fourier para $f(x) = x$ es:

$$f(x) = \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \frac{i(-1)^n}{n} e^{inx}$$

Ambas representaciones son equivalentes y proporcionan la expansión de Fourier correcta para la función dada.

APÉNDICE B

Simulator

```
% Numero de terminos en la serie de Fourier
N = 10; % Puedes modificar N para mejorar la
      aproximacion

% Vector de tiempo de -pi a pi
t = linspace(-pi, pi, 1000);

% Funcion original x(t) = t
x_original = t;

% Inicializacion de la aproximacion de la serie
      de Fourier
x_aprox = zeros(size(t));

% Calculo de la serie de Fourier
for n = 1:N
    bn = (2 / n) * (-1)^(n + 1);
    x_aprox = x_aprox + bn * sin(n * t);
end

% Grafica de la funcion original y su
      aproximacion
figure;
plot(t, x_original, 'k', 'LineWidth', 1.5); %
      Funcion original en negro
hold on;
plot(t, x_aprox, 'r', 'LineWidth', 1.5); %
      Aproximacion en rojo
legend('Funcion original x(t)', ['Aproximacion
      con N = ', num2str(N), ' terminos']);
xlabel('t');
ylabel('x(t)');
title('Serie de Fourier de x(t) = t');
grid on;
hold off;
```