Instituto Tecnológico de Chihuahua II

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

APLICACIÓN MÓVIL PARA DESARROLLAR Y GRAFICAR SERIES DE FOURIER

TESIS

PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA

JESÚS FRANCISCO DUARTE MARTÍNEZ

DIRECTOR DE TESIS

DR. ALBERTO CAMACHO RÍOS

CODIRECTOR DE TESIS

DR. JESÚS HUMBERTO CUEVAS ACOSTA

CHIHUAHUA, CHIH., JUNIO 2020

Dictamen

Chihuahua, Chihuahua, 05 de junio del 2020

M.C. MARÍA ELENA MARTÍNEZ CASTELLANOS
COORDINADORA DE POSGRADO
Presente. -

Por medio de este conducto el comité tutorial revisor de la tesis para obtención de grado de Maestro en Sistemas Computacionales, que lleva por nombre **"APLICACIÓN MÓVIL PARA DESARROLLAR Y GRAFICAR SERIES DE FOURIER"**, que presenta el (la) **C. JESÚS FRANCISCO DUARTE MARTÍNEZ**, hace de su conocimiento que después de ser revisado ha dictaminado la **APROBACIÓN** del mismo.

Sin otro particular de momento, queda de Usted.

Atentamente

La Comisión de Revisión de Tesis.

DR. ALBERTO CAMACHO RIOS	**DR. JESÚS HUMBERTO CUEVAS ACOSTA**
Director de Tesis	**Co-Director**
Leonardo Nevárez Chávez	
M.C. LEONARDO NEVÁREZ CHÁVEZ	**M.C. ARTURO LEGARDA SÁENZ**
Revisor	**Revisor**

EDUCACIÓN
INSTITUTO TECNOLÓGICO DE CHIHUAHUA II
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

DEDICATORIA

Dedico este trabajo a mis hijos Víctor y Daniel
y a mi esposa Judith.
por siempre estar a mi lado apoyándome.

AGRADECIMIENTOS

Al M.C. José Rivera ex director del Instituto Tecnológico de Chihuahua por confiar en mi persona y darme el aval para realizar estudios de Posgrado

Al Instituto Tecnológico de Chihuahua II, por abrirme las puertas del conocimiento y cultura que habitan en su interior.

Al Dr. Alberto Camacho Ríos quien trabajó a mi lado y enriqueció en gran medida este proyecto.

A los miembros del jurado Dr. Alberto Camacho Ríos, Dr. Jesús Humberto Cuevas Acosta, M.C. Leonardo Nevárez Chávez y M.C. Arturo Legarda Sáenz por su interés en ser sinodales y formar parte de este momento en mi vida.

A los profesores que tuve durante la carrera, el estudiante es el resultado de la suma de todos los maestros que tuvo.

A mis compañeros de carrera con quienes compartí mayormente buenos momentos.

RESUMEN

En la actualidad existen varias alternativas de aplicaciones móviles para el trabajo de Series de Fourier, tanto gratuitas como de paga, pero no existe alguna que cumpla con todos los requisitos planteados para este proyecto. El objetivo fue desarrollar una aplicación móvil que desarrollara y graficara series de Fourier de funciones continuas, de dos y tres trozos, que fuera gratis, sin publicidad, que representara los pasos simbólicamente con alto grado de fidelidad con el salón de clase y que no necesitara Internet para ejecutar los procesos necesarios

La implementación fue desarrollada para teléfonos inteligentes con sistema operativo Android. El desarrollo de la aplicación se implementó en Python con la librería SymPy para utilizar cálculo simbólico. Los resultados mostrados por la aplicación están basados en la fidelidad con el desarrollo en el salón de clase. La aplicación realiza todos los cálculos de manera local, gracias a la librería Chaquopy, el código Python se puede enlazar con Java, por lo que no es necesario utilizar un servidor en la red de Internet para este proceso.

La aplicación está disponible en la tienda de Google Play, con más de ocho mil descargas y con una puntuación de 4.25 puntos.

ABSTRACT

Currently, there are several mobile application alternatives for the Fourier Series work, both free and paid, but there is none that meets all the requirements for this project. The objective was to develop a mobile application that would develop and graph Fourier series of continuous functions, of two and three pieces, that was free, without advertising, that represented the steps symbolically with a high degree of fidelity with the classroom and that did not need Internet to execute the necessary processes

The implementation was developed for smartphones with Android operating system. The application development was implemented in Python with the SymPy library to use symbolic calculation. The results shown by the application are based on fidelity with development in the classroom. The application performs all the calculations locally, thanks to the Chaquopy library, the Python code can be linked to Java, so it is not necessary to use a server on the Internet for this process.

The application is available in the Google Play store, with more than eight thousand downloads and a score of 4.25 points.

Contenido

Capítulo I. Introducción

1.1. Introducción.

En la actualidad, el uso del software matemático o aplicaciones matemáticas, permite a docentes y alumnos mejoren los procesos de enseñanza – aprendizaje de forma eficaz en la adquisición del conocimiento matemático. Dicho software se apoya en representaciones geométricas y gráficas para clarificar las cuestiones algorítmicas en los desarrollos y procedimientos en la resolución de problemas (Mosquera & Vivas, 2017).

Según Flores, Valadez y Atencio (2016), diferentes estudios demuestran que los alumnos al utilizar tecnología en su proceso de aprendizaje, tendrán más tiempo para descubrir, entender y aplicar conceptos y, por lo tanto, llegar a la resolución de problemas, y coadyuvando en el proceso de aprendizaje del estudiante.

La tecnología no se puede entender de manera independiente de la sociedad, por el contrario, se debe considerar una dialéctica entre ambas: “la tecnología influye en la sociedad, y esta la retroalimenta, modificándola permanentemente, llegando, incluso, a provocar en determinados momentos históricos un cambio tecnológico de gran impacto en el hombre y en su forma de percibir el mundo” (Assum, Guil, & Malet, 2014).

La tecnología debe usarse como un instrumento capaz de potenciar las actividades de las personas, con la cual se dinamice la adquisición de conocimiento, de perseguir nuevas metas, de corroborar hipótesis, etc. “La tecnología como un puente entre distintos conocimientos, introducida con un propósito y una aplicación concreta”, (Salomon, 1991, citado en Assum, et al., 2014).

El Tecnológico Nacional de México (TecNM), promueve en sus distintos programas de licenciatura el uso de la tecnología para lograr las competencias, tanto profesionales como genéricas sugeridas en los mismos. Es importante considerar que, en los programas de estudio de Ecuaciones Diferenciales, así como en los libros de texto utilizados, se sugiere el uso de las

TIC's (software educativo en forma de Sistemas Algebraicos Computarizados (SAC)) por ejemplo: Mathematica, Matlab, Maple, Derive, Mathcad, Geogebra, entre otros.

En este trabajo, primero se identifican las herramientas de software, ya diseñadas, utilizadas como apoyo didáctico en la graficación de Series de Fourier, dirigido a estudiantes de la materia de ED que se imparte en el TecNM. En la sección correspondiente al Estado del Arte, se menciona la revisión del uso de aplicaciones móviles para la enseñanza de las matemáticas, específicamente aquellas que grafican Series de Fourier. En la práctica escolar, este tema no siempre se aborda con profundidad, debido a la cantidad de operaciones necesarias y a la falta de tecnología para graficar dichas soluciones.

Posteriormente, se analizan y proponen los criterios de inclusión para la selección de la herramienta, seleccionando siete de las aplicaciones móviles para realizar un análisis comparativo de sus funcionalidades, orientadas a desarrollar las competencias tecnológicas establecidas en los programas de estudio.

Finalmente, se desarrolló la aplicación para dispositivos móviles que contiene los criterios planteados.

1.2. Planteamiento del problema

Actualmente, existe software especializado que permite tratar casos de estudio que implican el análisis y representación gráfica de Series de Fourier. No obstante, dicho software carece de atributos necesarios para su utilización en el aula de clase, a saber:

Algunos no son de uso libre. Son de uso privativo, es necesario pagar un costo alto para tener una licencia. En el caso de las aplicaciones que sí lo son, incluyen publicidad en la pantalla que dificulta la visualización de la información.

No son capaces de presentar el desarrollo el análisis matemático, al menos no con el grado de detalle necesario en un entorno educativo.

Las aplicaciones de uso libre, en general, fueron diseñadas como meras calculadoras, valiosas, pero insuficientes en un entorno de enseñanza, cuyo propósito central es aprender desde la génesis misma de problema bajo estudio, pasando por la lógica del tratamiento matemático, la generación de resultados numéricos y su representación gráfica. Ya que los procesos son matemáticos y no de cálculo simbólico, lo anterior, dificulta su recomendación para su implementación en un curso de la naturaleza descrita en la sección anterior. Las aplicaciones de costo, si muestran sus procesos con cálculo simbólico, pero hay que hacerlo mediante instrucciones propias.

Las aplicaciones de uso libre no permiten su utilización de manera autónoma para su ejecución (compilación-interpretación); exigen una conexión a Internet para tal efecto.

1.3. Objetivo

Diseñar y crear una aplicación móvil que permita desarrollar y representar gráficamente Series de Fourier de funciones de entrada continuas, de dos y tres trozos en un intervalo determinado, pudiendo ser estas con simetría Par, Impar o sin simetría, la disponibilidad será de uso libre y gratuito con ausencia de publicidad, su orientación a la enseñanza y la posibilidad de mostrar el desarrollo algorítmico del proceso de solución con una alta fidelidad al proceso del salón de clase, los cálculos se harán en el dispositivo móvil, por lo que no necesitará de una conexión a Internet, constituyen las características más representativas y distintivas respecto de otras alternativas.

1.4. Alcances y Limitaciones

Alcances:

Las operaciones matemáticas para el desarrollo y graficación de la serie de Fourier se desarrollarán en el dispositivo móvil, sin necesidad de tener una conexión a Internet para la obtención de resultados.

- Se ejecutará en dispositivos móviles con sistema operativo Android 4.4 y superiores.

- Aceptará funciones continuas o, incluso, de hasta 3 trozos en un intervalo dado.
- Podrá desarrollar series de Fourier de funciones en las cuales se presenten indeterminaciones en alguno de sus primeros tres términos.
- Utilizará lenguaje matemático simbólico.
- Las funciones matemáticas podrán incluir las siguientes funciones: sen, cos, abs, y exp.

Limitaciones:

- No estará disponible para otros sistemas operativos.
- El tamaño de la aplicación es de 33 Mb aproximadamente.

1.5. Justificación

Actualmente, los planes de estudio de los cursos de matemáticas que se ofrecen en las carreras del nivel de ingeniería del Tecnológico Nacional de México (TecNM), destacan una problemática importante, poco atendida, que afecta especialmente la práctica escolar.

Aun cuando los planes de estudio sugieren el uso de las TICs, software educativo en forma de Sistemas Algebraicos Computarizados (SAC), por ejemplo: Mathematica, Matlab, Maple, Derive, Mathcad, Geogebra, etc., según Aparisi & Pochulu (2013), se han identificado algunos inconvenientes a que se enfrentan los docentes, cuando se presentan escenarios de modelamiento matemático para la resolución de problemas, como los que se enlistan a continuación:

1) Tiempo de clase. Los tiempos sugeridos en los programas no son suficientes para la incorporación de las TIC’s.
2) La licencia de uso. La licencia del software comercial sugerido es por demás cara, lo cual dificulta su adquisición y uso.

3) El método de enseñanza. El desconocimiento que se tiene por parte de los profesores de matemáticas de la utilidad del software sugerido, puede ubicar al docente en una zona de inseguridad.
4) Acceso a tecnología. En muchas ocasiones no se cuenta con la infraestructura tecnológica necesaria para implementar las TIC´s en el aula.
5) Alto número de estudiantes. Incide directamente en el logro de los objetivos. En ocasiones se dispersa la atención y es complicado trabajar con un software matemático especializado, como Mathematica o Matlab.
6) Tiempo de aprendizaje. El estudiante debe de tener conocimientos previos en el uso de software matemático, para aprovechar el tiempo en resolución de problemas y no en el aprendizaje de su interfaz.

Los componentes anteriores son algunos de los que llevan a que el aprendizaje de los conceptos de la matemática en este nivel de enseñanza sea deficiente, dificultando además el propio aprendizaje que se vincula entre las asignaturas, sin que por ello se cumpla con la adquisición de capacidades por parte de los estudiantes, que los lleven a utilizar las TICs en la resolución de problemas.

El proyecto se centra en las necesidades de contar con software que permita el desarrollo y la graficación de series de Fourier, centrando esta problemática en la unidad V del curso de Ecuaciones Diferenciales Ordinarias (EDO), que se lleva en las carreras de Ingeniería en Sistemas Computacionales, Ingeniería Eléctrica, Ingeniería Electrónica, entre otras, del TecNM.

La competencia específica se describe de la siguiente manera:

- Unidad V. Utilizar las TICs para graficar la solución de problemas de valor inicial resueltas en series de Fourier e interpretar sus resultados.

En la unidad V del programa, se estudian las definiciones básicas de ortogonalidad de funciones para construir una serie de Fourier en un intervalo arbitrario centrado y en medio intervalo. Dentro de las competencias genéricas, se busca propiciar el uso de nuevas tecnologías en la resolución de ese tipo de problemas. Para ello, el plan de estudios propone fincar un entorno propicio en el aula o laboratorio que promueva en el estudiante el uso de las TIC´s: Matcad, Mathematica, Maple e incluso calculadoras gráfico-simbólicas, de manera que se ahorre el trabajo operativo y se experimente con la situación en estudio bajo distintas condiciones. Dentro del aula, los profesores sugieren el uso de software, aun cuando no es claro si ellos mismos tienen un control de las actividades que con el mismo desarrollan.

Por otro lado, los autores de libros de texto de ecuaciones diferenciales más comunes en el nivel de ingeniería, sugieren el uso de software. Por ejemplo, Kreyszig (2011) propone, de manera opcional, el uso de Sistemas Asistidos por Computadora (SAC) como el Mathematica y Maple, para la resolución de problemas. Por su lado Zill (2008) sugiere los comandos del Mathematica para resolver ecuaciones diferenciales.

Tanto el programa del curso de Ecuaciones Diferenciales como los textos y el profesor, sugieren la incorporación de las TICs y SAC, en el salón de clases, sin que ello tenga un peso específico efectivo en la resolución de los problemas que se sugieren para cada unidad. Esta deficiencia hace que el curso se desarrolle como un compendio de algoritmia con el que solamente se resuelven EDO, dejando de lado la parte fundamental del objetivo del mismo, que es la Resolución de Problemas.

Capítulo II. Estado del Arte

En la actualidad existen diversos software que ayudan a resolver y graficar problemas matemáticos de diversa índole, desde programas computacionales de prestigio y de costo económico, así como software de prestigio y licencia libre. Ante ello, el objetivo en este apartado es seleccionar el más adecuado para la resolución y graficación de Series de Fourier. Para este fin, se emplean diversas condiciones de clasificar instrumentos para la evaluación de aprendizajes. Entre ellas puede utilizarse una evaluación referida a criterios, que a partir de comparaciones permite el establecimiento de un puntaje relacionado con el resultado de puntaje individual orientado a evaluar conocimientos, competencias o habilidades, (Covacevich, 2014).

El software matemático puede obtenerse para ser utilizado en sistemas basados en la Web, aplicaciones de escritorio y/o dispositivos móviles. Las aplicaciones que interesan son aquellas que se descargan al dispositivo móvil (teléfonos inteligentes en sistema operativo Android), ya que, por su popularidad, más de un 90% de los estudiantes tienen uno.

Como primera etapa metodológica, se inició la investigación con las palabras clave utilizadas en la tienda de Google Play Store, la cual arrojó diversas aplicaciones móviles relacionadas con las Series de Fourier. Se encontraron 13 aplicaciones.

Para determinar las tres mejores aplicaciones, Fallas & Chavarría (2010) explican que se requiere un proceso de validación que asegure la calidad y pertinencia de las mismas. Se pueden aplicar modelos de evaluación de software educativo que comprenda criterios de inclusión de características técnicas, así como de competencias matemáticas requeridas para la resolución de EDO en Series de Fourier (Mosquera & Vivas, 2017).

Según Abud (2005), antes de llevar a cabo alguna evaluación, se deberán definir y jerarquizar los factores a evaluar de acuerdo a la importancia tanto en niveles técnicos como educativos. Siguiendo a Fallas & Chavarría (2010), se emplearon cuestionarios de valoración como instrumentos de evaluación.

En lo que se refiere a esta investigación, para evaluar la calidad del software se asignó un peso de 40% al cuestionario de competencias matemáticas, un 36% para el factor de interfaz y usabilidad y un 24% en el aspecto técnico, siguiendo, de esta manera, la propuesta de Abud (2005) y Mosquera & Vivas (2017).

Según Abud (2005), la calidad del software se evalúa mediante un modelo de atributos múltiples a partir de la función:

$$U = \left[\sum_{k=1}^{n} w_k u_k\right] / 100 \qquad 2.1$$

donde: U es el valor de calidad global, w_k es el peso para el factor de calidad y u_k es el puntaje obtenido para la alternativa k

Dada la gran variedad de software matemático que existe, según Assum et, al. (2014) proponen los siguientes criterios para evaluar un software matemático:

- Software Libre. Desde el tipo de licencia o acceso al programa sea libre y por tiempo ilimitado.
- Sistema operativo. Se trate de un software diseñado para ejecutarse en los sistemas operativos más usados (Android en este caso).
- Idioma. De preferencia tenga la interface en español.
- Usabilidad. Que permita al usuario concentrarse en su tarea y no en ver cómo se maneja la aplicación.

Según Mosquera & Vivas (2017), al momento de evaluar un software, los criterios de inclusión técnicos asociados a la calidad del software y requerimientos de funcionamiento, son:

- Portabilidad del software. Que el software se pueda instalar en dispositivos móviles como los celulares.

- Requisitos del sistema operativo. El sistema operativo Android es el de mayor facilidad de adquisición por parte de estudiantes y docentes.
- Software libre. Que se tenga acceso ilimitado a la aplicación.
- Facilidad de instalación y operación del mismo sin manual.
- Idioma.
- Presentación y funcionamiento.

Para la realización de la presente investigación, se decidió trabajar con los siguientes criterios:

- Portabilidad del software: la aplicación debe ejecutarse en dispositivos móviles
- Sistema operativo. Se eligió el sistema operativo Android.
- Software libre.
- Idioma.
- Interfaz gráfica, que no requiera desarrollo de código de programación por el estudiante.
- Usabilidad.
- Que acepte funciones de hasta 3 trozos.
- Desarrolle la Serie de Fourier.
- Grafique la Serie de Fourier.

En la siguiente etapa de la metodología, se aplicaron los cuatro primeros criterios de inclusión técnica, se filtró el software propuesto obteniendo un total de 7 de los 13 software encontrados. Finalmente, se evaluó el software por criterios de inclusión de acorde a las competencias matemáticas, tomando en consideración los objetivos que se desean alcanzar con su utilidad.

El total de cada componente se obtiene de la suma de los valores para cada pregunta de los cuestionarios consignados en las Tablas 2.1, 2.2 y 2.3. El cálculo de la utilidad por componentes se obtiene al multiplicar el peso porcentual de la componente por la suma total de los ítems evaluados.

Para la escala de evaluación se tomaron en consideración los siguientes valores: (Mosquera & Vivas, 2017)

- Bajo: Se asigna el valor de 1 para indicar que el software no cumplió con dicho criterio.
- Medio: Se asigna el valor 3 para indicar que cumple con el criterio, pero no completamente, puesto que faltó dentro del mismo criterio el cumplimiento de alguno de los conceptos evaluados.
- Alto: Se asigna el valor 5 para indicar que el software cumplió a cabalidad con el criterio evaluado

Tabla 2.1. Cuestionario de Evaluación de las funcionalidades matemáticas asociadas a los objetivos y competencias de las Series de Fourier que permite desarrollar cada aplicación.

Funcionalidades Matemáticas	**Serie de Fourier**	**FouSE-Fourier Series Expansion**	**Calculadora de integrales**	**Desmos Calculadora Graficadora**	**Symbolab**	**GeoGebra Clásico**	**Excel**
Permite graficación de funciones	1	1	5	5	1	5	5
Acepta funciones por trozos	1	1	5	5	1	3	3
Obtiene los coeficientes a_0, a_n y b_n.	5	1	3	1	1	1	1
Desarrolla la Serie de Fourier	3	1	1	1	1	1	1
Grafica la Serie de Fourier	1	3	5	5	1	3	3
TOTAL	**11**	**7**	**19**	**17**	**5**	**13**	**13**
Cálculo de la utilidad componente de funcionalidades matemáticas	4.4	2.8	7.6	6.8	2	5.2	5.2

Fuente: Elaboración propia

Tabla 2.2. Cuestionario de Evaluación de los criterios técnicos que posee cada software.

Técnico	**Serie de Fourier**	**FouSE-Fourier Series Expansion**	**Calculadora de integrales**	**Desmos Calculadora Graficadora**	**Symbolab**	**GeoGebra Clásico**	**Excel**
Portabilidad del software	5	5	5	5	5	5	5
Sistema operativo Android	5	5	5	5	5	5	5
Software libre	5	5	5	5	3	5	3
Idioma español	5	5	1	3	5	5	5
Que no requiera desarrollo de código	5	5	5	5	5	1	1
Usabilidad	3	3	5	5	5	3	3
Resuelva de manera local	1	5	1	1	1	5	5
Exprese los resultados en forma matemática	5	1	1	1	1	1	1

TOTAL	**34**	**34**	**28**	**30**	**30**	**30**	**28**
Cálculo de la utilidad componente técnico	8.16	8.16	6.72	7.2	7.2	7.2	6.72

Fuente: Elaboración propia

Tabla 2.3. Cuestionario de Evaluación de la Interfaz gráfica y usabilidad de cada software.

Interfaz y Usabilidad	Serie de Fourier	FouSE-Fourier Series Expansion	Calculadora de integrales	Desmos Calculadora Graficadora	Symb olab	GeoGebr a Clásico	Excel
El diseño de la interfaz de usuario permite flexibilidad de navegación por las actividades (Marquès 2002)	5	3	5	5	5	3	3
La terminología utilizada por el software, coincide con la terminología usada por el docente (Floría, 2001)	5	5	5	5	5	3	3
Mensajes de error (Floría, 2001)	3	1	1	5	5	3	3
El software trae ejemplos ilustrativos de funcionamiento (Marquès 2002)	5	3	1	3	5	1	1
Realiza los cálculos paso a paso	5	1	1	3	3	1	1
Facilidad para entender qué datos se deben ingresar y los resultados proporcionados (Ferrari & Mariño, 2014)	5	1	5	5	5	1	1
El acceso a la ayuda está en una zona visible y reconocible (Ferrari & Mariño, 2014)	1	1	5	1	1	3	3
La herramienta está diseñada para necesitar el mínimo de ayuda y de instrucciones (Ferrari & Mariño, 2014)	5	3	5	5	5	1	1
La apariencia de la herramienta es agradable y sencilla (Ferrari & Mariño, 2014)	5	3	5	5	5	3	3
TOTAL	**39**	**21**	**33**	**37**	**39**	**19**	**19**
Cálculo de la utilidad componente interfaz y usabilidad	14.04	7.56	11.88	13.32	14.0 4	6.84	6.84

Fuente: Elaboración propia

Después de evaluar los criterios de los cuestionarios propuestos en las tablas 2.1, 2.2 y 2.3, se procedió a aplicar la función 2.1, con la cual se obtuvo la calidad global de cada software. Los resultados se presentan en la tabla 2.4.

Tabla 2.4. Evaluación con la calidad global de cada software.

Calidad global de cada software						
Serie de Fourier	FouSE-Fourier Series Expansion	Calculadora de integrales	Desmos Calculadora Graficadora	Symbolab	GeoGebra Clásico	Excel
26.60	18.52	26.20	27.32	23.24	19.24	18.76

Fuente: Elaboración propia

La puntuación máxima de la calidad global adquirida por un software es de 35.8 puntos, tomando como base que en cada factor su valor más alto es de 5. Este puntaje corresponde al 100% de la calidad global. En la tabla 2.5 se muestran los porcentajes que cada software obtuvo en los tres cuestionarios aplicados.

Tabla 2.5. Porcentaje de la calidad global de cada software.

Porcentaje de la calidad global de cada software						
Serie de Fourier	FouSE-Fourier Series Expansion	Calculadora de integrales	Desmos Calculadora Graficadora	Symbolab	GeoGebra Clásico	Excel
74.30%	51.73%	73.18%	76.31%	64.92%	53.74%	52.40%

Fuente: Elaboración propia

De los resultados que se muestran en la tabla 4 se observa que son tres aplicaciones de software las que están por encima de los 25 puntos, ya que cumplen con la mayoría de los criterios evaluados. Se observa también, que cuatro software están por debajo de los 25 puntos los cuales no cumplen con la mayoría de los criterios establecidos. Entre ellos se encuentra GeoGebra Clásico, aun cuando esta aplicación es suficientemente potente para diversos cálculos de matemáticas e, incluso, determina buenas gráficas. Se observó que el software no puede graficar una función en series de Fourier de manera directa y sólo puede hacerlo utilizando secuencias de funciones intermedias. En este sentido no es software intuitivo y lleva algo de tiempo dominarlo.

La aplicación de Excel, aunque no es precisamente para cálculo matemático, elabora gráficas de funciones en series de Fourier, pero presenta el mismo problema de GeoGebra, son necesarios una buena cantidad de pasos para poder lograr la gráfica.

En el caso de Symbolab, resulta una aplicación útil para la resolución de ecuaciones diferenciales de forma simbólica y cuenta con buena interfaz y usabilidad. Sin embargo, no llega a graficar funciones en serie de Fourier. En el caso de la aplicación FouSE-Fourier Series Expansion, presenta tipos de funciones ya prestablecidas y sólo se puede modificar el número de términos con los que se desea construir la gráfica.

En lo que se refiere a las aplicaciones que estuvieron arriba del 70% de calidad global, se cuentan tres. Serie de Fourier presenta un buen diseño y una interfaz intuitiva, da como resultado los coeficientes a_0, a_n y b_n de la serie de Fourier, pero no está diseñada para obtener su gráfica. En el caso de Calculadora de Integrales, tiene una interfaz limitada, la cual acepta solo una función de dos trozos así como el número de términos con los que se quiere graficar. Da por resultado los valores numéricos de los coeficientes de a_0, a_n y b_n de cada término y, por último, devuelve la gráfica solo en el intervalo en el cual es periódica.

Por último, la aplicación que obtuvo mejor porcentaje de calidad global fue Desmos Calculadora Graficadora, la cual está diseñada para mostrar gráficas de cualquier función, incluyendo la sumatoria de una Serie de Fourier. Puede resolver gran cantidad de problemas matemáticos, pero no una función en Serie de Fourier. Otra consideración es que sólo está disponible para versiones de Android 5.0 en adelante.

Capítulo III. Marco Teórico

3.1 Serie de Fourier

La idea básica y central de las series de Fourier es que toda función periódica de período T, puede ser expresada como una suma trigonométrica de senos y cosenos dentro del mismo período T.

Una serie de Fourier es una serie infinita que converge puntualmente a una función periódica y continua a trozos (o por partes). Las series de Fourier constituyen la herramienta matemática básica del análisis de Fourier empleada para analizar funciones periódicas a través de la descomposición de dicha función en una suma infinita de funciones sinusoidales mucho más simples (como combinación de senos y cosenos con frecuencias enteras). El nombre se debe al matemático francés Jean-Baptiste Joseph Fourier, que desarrolló la teoría cuando estudiaba la ecuación del calor. Fue el primero que estudió tales series sistemáticamente y publicó sus resultados iniciales en 1807 y 1811.

Es una expresión utilizada en varias ramas de la ingeniería, además de ser una herramienta sumamente útil en la teoría matemática abstracta. Áreas en las que se aplican series de Fourier incluyen análisis vibratorio, acústica, óptica, procesamiento de imágenes y señales, y compresión de datos. En ingeniería, para el caso de los sistemas de telecomunicaciones y a través del uso de los componentes espectrales de frecuencia de una señal dada, se puede optimizar el diseño de un sistema para la señal portadora del mismo. Refiérase al uso de un analizador de espectros.

Serie Trigonométrica.

Se parte de que el siguiente conjunto de funciones trigonométricas

$$\left\{1, \cos\frac{\pi}{p}x, \cos\frac{2\pi}{p}x, \cos\frac{3\pi}{p}x, \dots, \operatorname{sen}\frac{\pi}{p}x, \operatorname{sen}\frac{2\pi}{p}x, \operatorname{sen}\frac{3\pi}{p}x, \dots\right\} \tag{3.1}$$

Es ortogonal en el intervalo [-p, p]. Surge la necesidad de desarrollar una función *f* definida sobre [-p, p] en una serie ortogonal que consista en las funciones trigonométricas dadas en (3.1), es decir,

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\alpha} \left(a_n \cos\left(\frac{n\pi}{p}x\right) + b_n \operatorname{sen}\left(\frac{n\pi}{p}x\right) \right) \tag{3.2}$$

Al integrar ambos lados de (3.2) desde -p a p resulta:

$$\int_{-p}^{p} f(x) = \frac{a_0}{2} \int_{-p}^{p} dx + \sum_{n=1}^{\alpha} \left(a_n \int_{-p}^{p} \cos\left(\frac{n\pi}{p}x\right) dx + b_n \int_{-p}^{p} \operatorname{sen}\left(\frac{n\pi}{p}x\right) dx \right) \tag{3.3}$$

Puesto que $\cos\left(\frac{n\pi}{p}x\right)$ y $\operatorname{sen}\left(\frac{n\pi}{p}x\right)$, $n \geq 1$, son ortogonales a 1 en el intervalo, el segundo miembro de (3.3) se reduce a un solo término:

$$\int_{-p}^{p} f(x) = \frac{a_0}{2} \int_{-p}^{p} dx = \frac{a_0}{2} x \Big|_{-p}^{p} = p a_0 \tag{3.4}$$

Al despejar a_0 se obtiene:

$$a_0 = \frac{1}{p} \int_{-p}^{p} f(x)\, dx \tag{3.5}$$

Multiplicando (3.2) por $\cos\left(\frac{n\pi}{p}x\right)$ e integrando:

$$\int_{-p}^{p} f(x) \cos\left(\frac{m\pi}{p}x\right) dx = \frac{a_0}{2} \int_{-p}^{p} \cos\left(\frac{m\pi}{p}x\right) dx$$

$$+ \sum_{n=1}^{\alpha} \left(a_n \int_{-p}^{p} \cos\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx + b_n \int_{-p}^{p} \operatorname{sen}\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx \right) \tag{3.6}$$

Mediante la ortogonalidad, se tiene:

$$\int_{-p}^{p} \cos\left(\frac{m\pi}{p}x\right) dx = 0, \;\; m > 0, \qquad \int_{-p}^{p} \cos\left(\frac{m\pi}{p}x\right) \operatorname{sen}\left(\frac{n\pi}{p}x\right) dx = 0$$

y

$$\int_{-p}^{p} \cos\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx = \begin{cases} 0, & m \neq n \\ p, & m = n \end{cases}$$

Por lo tanto, (3.6) se puede simplificar a

$$\int_{-p}^{p} f(x) \cos\left(\frac{n\pi}{p}x\right) dx = a_n p$$

y así

$$a_n = \frac{1}{p} \int_{-p}^{p} f(x) \cos\left(\frac{n\pi}{p}x\right) dx \tag{3.7}$$

Por último, multiplicando (3.2) por sen $\left(\frac{n\pi}{p}x\right)$, integrando y utilizando los resultados:

$$\int_{-p}^{p} \operatorname{sen}\left(\frac{m\pi}{p}x\right) dx = 0, \;\; m > 0, \qquad \int_{-p}^{p} \operatorname{sen}\left(\frac{m\pi}{p}x\right) \operatorname{sen}\left(\frac{n\pi}{p}x\right) dx = 0$$

y

$$\int_{-p}^{p} \operatorname{sen}\left(\frac{m\pi}{p}x\right) \cos\left(\frac{n\pi}{p}x\right) dx = \begin{cases} 0, & m \neq n \\ p, & m = n \end{cases}$$

Se encuentra que:

$$b_n = \frac{1}{p} \int_{-p}^{p} f(x) \operatorname{sen}\left(\frac{n\pi}{p}x\right) dx \tag{3.8}$$

Se dice que la serie trigonométrica (2) con coeficientes a_0, a_n y b_n definidos por (3.5), (3.7) y (3.8), respectivamente, se conoce como serie de Fourier de la función *f*. Los coeficientes obtenidos a partir de (3.5), (3.7) y (3.8) se conocen como coeficientes de Fourier de *f*.

Para calcular los coeficientes a_0, a_n y b_n, se supone que *f* es integrable en el intervalo y que (3.2), así como la serie obtenida al multiplicar (3.2) por $\cos\left(\frac{n\pi}{p}x\right)$, converge de tal manera que permite la integración término a término. Hasta que se demuestre que (3.2) es convergente para una función f dada, el signo de igualdad no se tomará en sentido estricto o literal. En algunos textos se utiliza el símbolo ~ en lugar de =. En vista de que la mayoría de las funciones incluidas en las aplicaciones son de un tipo que garantiza la convergencia de la serie, aquí utilizaremos el símbolo de igualdad.

Definición de la Serie de Fourier (Zill & Warren, 2015):
La serie de Fourier de una función f definida en el intervalo (-p, p), está dada por:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\alpha}\left(a_n \cos\left(\frac{n\pi}{p}t\right) + b_n \sin\left(\frac{n\pi}{p}t\right)\right) \tag{3.9}$$

donde

$$a_0 = \frac{1}{p}\int_{-p}^{p} f(x)\,dx \tag{3.10}$$

$$a_n = \frac{1}{p}\int_{-p}^{p} f(x)\cos\left(\frac{n\pi}{p}x\right)dx \tag{3.11}$$

$$b_n = \frac{1}{p}\int_{-p}^{p} f(x)\,\mathrm{sen}\left(\frac{n\pi}{p}x\right)dx \tag{3.12}$$

La forma canónica de las series de Fourier es la que se ha estado utilizando hasta el momento, donde la función en cuestión está definida sobre el intervalo [-π, π].

Extensión periódica. Se observa que cada una de las funciones incluidas en el conjunto básico (3.1) tiene un periodo fundamental diferente, es decir, $\frac{2\pi}{n}$, n ≥ 1; sin embargo, puesto que un múltiplo entero positivo de un periodo es también un periodo, se puede ver que todas las funciones tienen en común el periodo 2p. En consecuencia, el lado derecho de (3.2) tiene periodo 2p; de hecho, 2p es el periodo fundamental de la suma. Se concluye que una serie de Fourier no solo representa la función en el intervalo (-p, p), sino también proporciona la extensión periódica de *f* fuera de este intervalo. Se puede suponer desde el principio que la función dada es periódica con periodo T = 2p; esto es, f (x + T) = f(x).

En ocasiones es deseable adaptar la forma de una serie de Fourier a funciones periódicas de periodo T = 2L > 0 en el intervalo [-L, L]. Esto se consigue gracias a un cambio de variable.

Se puede reescribir la serie de Fourier y los coeficientes a_0, a_n y b_n considerando lo anteriormente expuesto de la siguiente manera:

La serie de Fourier de una función f definida en el periodo T = 2L en el intervalo (-L, L), está dada por:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\alpha} \left(a_n \cos\left(\frac{2n\pi}{T} x\right) + b_n \sin\left(\frac{2n\pi}{T} x\right) \right) \tag{3.13}$$

donde

$$a_0 = \frac{2}{T} \int_{-L}^{L} f(x)\, dx \tag{3.14}$$

$$a_n = \frac{2}{T} \int_{-L}^{L} f(x) \cos\left(\frac{2n\pi}{T} x\right) dx \tag{3.15}$$

$$b_n = \frac{2}{T} \int_{-L}^{L} f(x) \operatorname{sen}\left(\frac{2n\pi}{T} x\right) dx \tag{3.16}$$

Puesto que elaborar un código computacional hace necesario contar con generalidades y símbolos durante la resolución de problemas, la optimización del código es fundamental. Ante ello, se decidió utilizar el tipo de nomenclatura que se ha presentado en (3.13), (3.14), (3.15) y (3.16), ya que soporta intervalos simétricos y asimétricos de cualquier periodo.

Series de Fourier de cosenos y senos

El esfuerzo que se lleva a cabo en la evaluación de los coeficientes a_0, a_n y b_n al desarrollar una función *f* en una serie de Fourier se reduce significativamente cuando *f* es una función par o impar.

Se dice que una función *f* es:

Par si $f(-x) = f(x)$ e **Impar** si $f(-x) = -f(x)$.

En un intervalo simétrico (-L, L), la gráfica de una función par cuenta con simetría respecto al eje *y*, mientras que la gráfica de una función impar tiene simetría con relación al origen.

La serie de Fourier de una función Par en el intervalo (-L, L) es la serie de cosenos:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\alpha} a_n \cos\left(\frac{2n\pi}{T} x\right)$$

donde

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

$$a_n = \frac{2}{T} \int_{-L}^{L} f(x) \cos\left(\frac{2n\pi}{T} x\right) dx$$

La serie de Fourier de una función Impar en el intervalo (-L, L) es la serie de senos:

$$f(x) = \sum_{n=1}^{\alpha} b_n \sin\left(\frac{2n\pi}{T} x\right)$$

donde

$$b_n = \frac{2}{T}\int_{-L}^{L} f(x)\,\mathrm{sen}\left(\frac{2n\pi}{T}x\right)dx$$

3.2 Fidelidad.

Con el desarrollo de la aplicación se busca que los resultados que se obtengan sean lo más parecidos a la resolución que se elabora en clase. Esto con la finalidad de que los alumnos usuarios sean conscientes de los pasos de resolución en el desarrollo de Series de Fourier.

"Transposición Informática", frase acuñada por Balacheff (1994) citado en Camacho et. al., (2019), habla del fenómeno que vivimos cotidianamente en las escuelas, en donde se van introduciendo Software en el ámbito de la enseñanza matemática, pudiendo generar conflictos cognitivos, ya que se están reproduciendo conocimientos matemáticos en contextos diferentes a los que fue producida. Esto es que, la representación matemática de las posibles soluciones dependerá de la conceptualización e interpretación del autor del software matemático y no necesariamente de las bases metodológicas del alumno.

Camacho et. al, (2019) nos dicen que: "*El concepto de fidelidad es una norma con la que pretendemos que las distorsiones y perturbaciones epistémicas provocadas por la asociación del conocimiento matemático y el software sea mínimo*". Para generar el menor conflicto posible, el software matemático desarrollado, se tiene que adaptar lo suficiente a la resolución en el cuaderno que pueda hacer un estudiante.

Capítulo IV. Desarrollo

Para desarrollar esta aplicación, se revisaron las diferentes opciones de lenguajes de programación que pudieran cumplir con las especificaciones ya descritas, como son, el que pueda ejecutarse en un dispositivo móvil, que pueda utilizar cálculo simbólico y que pueda hacer todos los cálculos en el mismo dispositivo móvil. Según Rackauckas (2018), el lenguaje de programación Julia es el que cumplió más requisitos en su investigación de lenguajes de programación para resolver Ecuaciones Diferenciales Ordinarias. De los lenguajes revisados Matlab, R, Julia, Phyton, C, Mathematica, Maple y Fortran; R y Python son los otros dos lenguajes que cumplen requisitos para este objetivo.

4.1 Computación simbólica.

La computación simbólica se ocupa de la computación de objetos matemáticos simbólicamente. Esto significa que los objetos matemáticos se representan exactamente, no aproximadamente, y las expresiones matemáticas con variables no evaluadas se dejan en forma simbólica.

El poder real de un sistema de cálculo simbólico como la biblioteca SymPy es la capacidad de hacer todo tipo de cálculos simbólicamente. SymPy puede simplificar expresiones, calcular derivadas, integrales y límites, resolver ecuaciones, trabajar con matrices y mucho más, y hacerlo todo simbólicamente. Incluye módulos para trazar, imprimir (como la salida impresa en 2D de fórmulas matemáticas, o LATEX), generación de código, física, estadística, combinatoria, teoría de números, geometría, lógica y más.

Un requisito importante es que la aplicación pueda representar de manera simbólica los pasos intermedios y los resultados finales. Para este punto, se decidió utilizar SymPy. Una ventaja de esta librería es que se puede ejecutar en R, Julia y Python.

4.2 Software utilizado para el desarrollo

4.2.1 Sympy.

SymPy es una biblioteca de Python para matemática simbólica. Su objetivo es convertirse en un sistema de álgebra computacional (CAS) con todas las funciones, manteniendo el código lo más simple posible para que sea comprensible y fácilmente extensible. SymPy está escrito completamente en Python (SymPy Development Team, 2020).

¿Qué hace que SymPy sea una mejor opción que las alternativas? se puede afirmar lo siguiente:

En primer lugar, SymPy es completamente gratis. Es de código abierto y tiene licencia bajo la licencia BSD liberal, por lo que puede modificar el código fuente e incluso venderlo si lo desea. Esto contrasta con los sistemas comerciales populares como Maple o Mathematica que cuestan cientos de dólares en licencias.

En segundo lugar, SymPy usa Python. La mayoría de los sistemas de álgebra computacional inventan su propio lenguaje. SymPy está escrito completamente en Python y se ejecuta completamente en Python. Esto significa que, si ya conoce Python, es mucho más fácil comenzar a usar SymPy, porque ya conoce la sintaxis (y si no conoce Python, es realmente fácil de aprender). Ya sabemos que Python es un lenguaje bien diseñado y probado en batalla. Los desarrolladores de SymPy confían en sus habilidades para escribir software matemático, pero el diseño del lenguaje de programación es algo completamente diferente. Al reutilizar un lenguaje existente, podemos centrarnos en aquellas cosas que importan: las matemáticas.

Una ventaja de SymPy es que es liviano. Además de ser relativamente pequeño, no tiene otras dependencias que Python, por lo que se puede usar en casi cualquier lugar fácilmente. SymPy, por otro lado, pretende ser un sistema independiente, con todas las características implementadas en SymPy.

Una última característica importante de SymPy es que se puede usar como una biblioteca. Muchos sistemas de álgebra computacional se centran en ser utilizables en entornos interactivos, pero si desea automatizarlos o ampliarlos, es difícil hacerlo. Con SymPy, puede usarlo fácilmente en un entorno interactivo de Python o importarlo en su propia aplicación Python. SymPy también proporciona API para facilitar la extensión con sus propias funciones personalizadas (SymPy Development Team, 2020).

4.2.2 Numpy.

Según la página web oficial numpy (Numpy, 2020), se define como sigue: NumPy es el paquete fundamental para la computación científica con Python. Contiene entre otras cosas:

- Un poderoso objeto de matriz N-dimensional.
- Funciones sofisticadas (de transmisión).
- Herramientas para integrar código C / C ++ y Fortran.
- Álgebra lineal útil, transformada de Fourier y capacidades de números aleatorios.

Además de sus usos científicos obvios, NumPy también se puede utilizar como un contenedor eficiente multidimensional de datos genéricos. Se pueden definir tipos de datos arbitrarios. Esto permite que NumPy se integre sin problemas y rápidamente con una amplia variedad de bases de datos.

NumPy tiene licencia bajo la licencia BSD, lo que permite su reutilización con pocas restricciones.

4.2.3 Python.

Según Python, (2020): “Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Y define este como un lenguaje multiparadigma, debido a que soporta orientación a objetos, programación imperativa y en menor medida programación funcional. Es interpretado de tipado dinámico y multiplataforma”.

Python es un lenguaje de propósito general, eso significa que no está orientado a un fin concreto. Con Python se pueden crear páginas web sin tener un alto conocimiento (con Javascript como un poderoso aliado), pero también hacer scripts o software para el sistema operativo Windows.

Python aún no se puede ejecutar en dispositivos móviles, pero se puede usar Kivy o Chaquopy para este propósito.

Comentando un poco más las características:

- Python es un lenguaje multiparadigma, porque soporta más de un paradigma (modelo de desarrollo de la programación).
- Es interpretado porque el intérprete va a traducir el código a medida que se necesita.
- Es de tipado dinámico porque permite la mutación (transformación) de variables.
- Es multiplataforma porque se puede ejecutar en cualquier sistema operativo.

Otras funciones o elementos propios de Python:

- De libre distribución.
- Gracias a su popularidad, existen una gran cantidad de librerías y funciones ya hechas, que se pueden utilizar gracias a su extensa biblioteca.
- Tiene soporte para múltiple variedad de bases de datos.
- Tiene un gran soporte gracias a su comunidad. Por ejemplo, la última versión de Visual Studio permite desarrollar en Python, o la comunidad de la página oficial de Python, dónde se pueden ver todas las actividades que hacen en el mundo.

(Pythones, 2020)

4.2.4 Spyder.

Para trabajar con el lenguaje Python con las librerías NumPy y SymPy, se necesita de un IDE. Se decidió utilizar el entorno de Spyder que viene dentro de la suite Anaconda. Este IDE muestra simplicidad en su interface sin sacrificar su potencia, aunque la suite de Anaconda es grande, el IDE de Spyder es rápido en cargar y ligero en compilar.

En la página web oficial (https://www.spyder-ide.org/) nos da la siguiente definición: “Spyder es un poderoso entorno científico escrito en Python, para Python, y diseñado por y para científicos, ingenieros y analistas de datos. Ofrece una combinación única de la funcionalidad avanzada de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, ejecución interactiva, inspección profunda y hermosas capacidades de visualización de un paquete científico. Más allá de sus muchas funciones integradas, sus capacidades se pueden ampliar aún más a través de su sistema de complementos y API. Además, Spyder también se puede utilizar como una biblioteca de extensión PyQt5, lo que permite a los desarrolladores desplegar su funcionalidad e integrar sus componentes, como la consola interactiva, en su propio software PyQt.”.

4.2.5 Android.

Android es un sistema operativo móvil desarrollado por Google, basado en Kernel de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes (Wear OS), automóviles (Android Auto) y televisores (Android TV).

Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y que adquirió en 2005. Android fue presentado en 2007 junto con la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El código fuente principal de Android se conoce como Android Open Source Project (AOSP), que se licencia principalmente bajo la Licencia Apache. Android es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado superior al 80% al año 2017, muy por encima de IOS.

Aplicaciones

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios marcos de aplicaciones basadas en la web multiteléfono.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o SDK provisto por Google el cual se puede descargar gratuitamente.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

Para el caso de Python, hasta ahora no se puede ejecutar directamente en Android. Existen otras herramientas para poder trabajar con lenguaje Python, como es el caso del Framework de Kivy y Chaquopy que se distribuye como un complemento para el sistema de compilación basado en Gradle de Android.

4.26 Android Studio.

Para desarrollar la aplicación móvil, se utilizó el IDE de Android Studio. De esta manera se obtuvo el archivo APK, que es el que acepta la Play Store para poder distribuir la aplicación móvil.

La definición que se encuentra en la página web oficial (https://developer.android.com/studio/intro) es la siguiente: “Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de ejemplo.

- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros.
- Compatibilidad con C++ y NDK.
- Compatibilidad integrada para Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine".

El lenguaje nativo que se utiliza es Java, aunque Kotlin está tomando más fuerza día con día. En el caso de la aplicación desarrollada, se utilizó el lenguaje Java, por ser más popular y tener una comunidad más grande de desarrolladores, así con un gran repositorio de librerías.

4.2.7 Java.

Según la definición de su página oficial (https://www.java.com/es/download/faq/whatis_java.xml), se define como: "Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes".

4.2.8 MathView.

Hasta aquí, tenemos lenguajes y librerías para resolver problemas matemáticos y mostrarlos en la pantalla del móvil, sin embargo, los resultados están en LaTex, por lo que se tiene que usar alguna biblioteca que pueda renderizar estos resultados en modo texto y mostrarlos de forma matemática.

MathView una biblioteca de vista de terceros, que nos ayuda a mostrar las fórmulas matemáticas en las aplicaciones de Android más fácilmente. Tiene dos motores de renderizado disponibles: MathJax y KaTeX. Admite desde la versión 4.1 de Android (Jelly Bean) y más reciente.
(https://github.com/jianzhongli/MathView)

4.2.9 GraphView.

Cuando se obtiene el desarrollo de una Serie de Fourier, la aplicación genera los datos necesarios para representar su gráfica. Necesitamos hacer uso de una biblioteca para poder generar un espacio donde se represente su gráfica.

GraphView es una biblioteca para Android para crear mediante programación diagramas flexibles y atractivos. Es fácil de entender, integrar y personalizar. Cree gráficos de líneas, gráficos de barras, gráficos de puntos o implemente sus propios tipos personalizados. (https://github.com/jjoe64/GraphView).

4.2.10 Chaquopy.

Para poder ejecutar el código de Python en un dispositivo Android con Java, se debe utilizar un SDK que funcione como intérprete del lenguaje Python y a la vez como intermediario entre Python y Java y el sistema operativo Android.

El SDK de Chaquopy, es una forma de usar Python en las aplicaciones de Android. Se distribuye como un complemento para el sistema de compilación estándar de Android. La descarga y la instalación se automatizan a través de Gradle. El núcleo de Chaquopy es una interfaz de lenguaje Python/Java ligera pero flexible, que permite acceder a Java desde Python o Python desde Java. Es la mejor forma de entremezclar libremente los dos lenguajes en una aplicación, utilizando el que sea mejor para cada situación. La mayoría de los paquetes de PyPI también se pueden descargar automáticamente e incorporar a la aplicación (Chaquopy Ltd., 2020).

Interfaz Gráfica.

Esta aplicación móvil tiene una interfaz principal en la que se despliegan las tres opciones de funciones que se pueden introducir, además de un botón para obtener ayuda.

Esta app está diseñada para trabajar con los siguientes tipos de funciones:

Función continua definida en un intervalo [a, b]:

$$F(x) = \{\, f(x) \;\; si \quad a \leq x \leq b$$

Función compuesta por dos trozos definida de [a, c], donde f1(x) está definida en [a, b] y f2(x) está definida en (b, c]:

$$F(x) = \begin{cases} f1(x), & si \quad a \leq x \leq b \\ f2(x), & si \quad b < x \leq c \end{cases}$$

Función compuesta por tres trozos definida de [a, d], donde f1(x) está definida en [a, b], f2(x) está definida en (b, c] y f3(x) está definida en (c, d]:

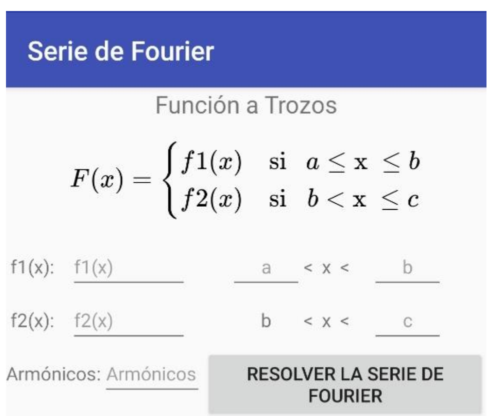
$$F(x) = \begin{cases} f1(x), & si \quad a \leq x \leq b \\ f2(x), & si \quad b < x \leq c \\ f3(x), & si \quad c < x \leq d \end{cases}$$

Una segunda vista es, la interfaz para introducir los datos con los que se quiere trabajar. Se pide lo siguiente:

- Función(es) f(x).
- Límites del intervalo donde la función f(x) es válida.
- Número de armónicos para poder realizar la gráfica.

La figura 4.1 muestra la pantalla para una función de dos trozos.

Fig. 4.1. Pantalla de la app para una función de dos trozos



Una vez introducidos los valores necesarios, el botón Resolver la serie de Fourier nos muestra el resultado en otra ventana.

La tercera ventana muestra los siguientes elementos:

- La función original.
- La grafica de la función original.
- El periodo.
- La simetría.
- Planteamiento del coeficiente a_{0}.
- Pasos de resolución del coeficiente a_{0}.
- Planteamiento del coeficiente a_{n}.
- Pasos de resolución del coeficiente a_{n}.
- Planteamiento del coeficiente b_{n}.
- Pasos de resolución del coeficiente b_{n}.
- Planteamiento de la serie de Fourier.
- Serie de Fourier obtenida.
- Gráfica de la serie de Fourier obtenida con el número de armónicos dado.

Estas interfaces están alojadas como Actividades (Activity) en Android. La comunicación se da entre actividades.

4.3 Descripción de las Librerías.

4.3.1 MathView.

Con respecto a la visualización de ecuaciones, Android no posee atributos o métodos. Para lograr este objetivo es necesario usar un lenguaje especializado como Latex. La inclusión de este lenguaje trae consigo diversos problemas ya que debe realizarse a través de una librería externa que tenga la capacidad de renderizar Latex.

El proceso de renderizado depende de un motor, dicho motor también es una librería y determina que símbolos se pueden mostrar y el tiempo que tarda en mostrar los resultados.

Existen varios motores dependiendo de la librería que se escoja, como pueden ser:

- jqMath
- KaTex

- MathJax

La librería que se utilizó fue MathView, la cual puede usar los motores KaTex y MathJax.

El comportamiento de MathView es casi el mismo que TextView, excepto que representará automáticamente el código TeX (o el código MathML si se procesa con MathJax) en una fórmula matemática. El resultado de las ecuaciones se dio bien con los dos motores. Se decidió utilizar el motor KaTex, ya que es más rápido en la renderización y acepta Latex sin ningún problema.

Al momento de renderizar el código Latex, se debe tomar en cuenta lo siguiente:

- Debe incluir la fórmula en \(...\) para que esta se muestre en línea con el texto.
- Debe incluir la fórmula en $$...$$ para que las fórmulas se muestren en una línea sola.
- Debe escapar de los caracteres espaciales como la barra diagonal inversa, las comillas, etc.

A continuación, se muestra un segmento de código con esta implementación.

Se define MathView en el archivo de diseño xml:

```
<LinearLayout ...>
         …
         …
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Formula: de Java String con KaTeX"
        android:textStyle="bold"/>

    <io.github.kexanie.library.MathView
        android:id="@+id/formula"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        auto:engine="KaTeX" >
    </io.github.kexanie.library.MathView>
         …
         …

</LinearLayout>
```

Obtener una instancia de la Actividad.

```
public class MainActivity extends AppCompatActivity {
    MathView formula;
    String tex = "Esto es una cadena de texto. Tú puedes insertar un formula en línea con el texto:" + " \\(ax^2 + bx + c = 0\\) " +
```

```
            "o mostrar una formula en una línea aparte: $$\\sum_{i=0}^n i^2 =
\\frac{(n^2+n)(2n+1)}{6}$$";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    protected void onResume() {
        super.onResume();

        formula = (MathView) findViewById(R.id.formula);
        formula.setText(tex);
    }
}
```

4.3.2 GraphView.

GraphView es una biblioteca para Android para crear mediante programación diagramas flexibles y atractivos. Es fácil de entender, integrar y personalizar. Esta librería se utilizó para crear las gráficas de las funciones de entrada y la serie de Fourier resultante.

A continuación, se muestra un ejemplo de la implementación de esta librería.

Se define el elemento GraphView en el archivo de diseño xml:

```
< com .jjoe64.graphview.GraphView
         android : layout_width = " match_parent "
         android : layout_height = " 200dip "
         android : id = " @ + id / graph " />
```

Obtener una instancia de la Actividad.

```
GraphView graph = (GraphView) findViewById(R.id.graph);
LineGraphSeries<DataPoint> series = new LineGraphSeries<>(new DataPoint[] {
          new DataPoint(0, 1),
          new DataPoint(1, 5),
          new DataPoint(2, 3)
});
graph.addSeries(series);
```

4.3.3 Teclado.

En el caso del teclado, es necesario reducir el número de teclas disponibles para la introducción de expresiones a desarrollar. Para eso, se escondió el teclado predeterminado de Android y se desarrolló otro teclado personalizado, el cual tiene solo las operaciones básicas permitidas por el algoritmo propuesto, como son la suma, resta, multiplicación, división y potencia, Valor Absoluto, las funciones trigonométricas de seno y coseno, y la función exponencial, los paréntesis y el punto decimal. Así como el símbolo de PI. Este teclado se desarrolló para minimizar los errores de sintaxis en las expresiones matemáticas introducidas.

4.3.4 Comunicación Python – Android.

Como ya se dijo, la librería Chaquopy fue la que se seleccionó para lograr ejecutar los archivos Python en un dispositivo Android y que hubiera una comunicación bidireccional de datos de entrada y de salida. La instalación y configuración de esta librería se encuentra documentada en su página web oficial.

El algoritmo que resuelve el problema de desarrollar la serie de Fourier, está codificado en el lenguaje de programación Python utilizando la librería Sympy. Como ya se ha dicho, el sistema operativo Android no cuenta con la capacidad de ejecutar archivos de Python, por lo que se tuvo que utilizar la librería de Chaquopy.

Primeramente, se tiene que crear una clase en Java, de donde se invoque el archivo Python que se requiera ejecutar, esto debe hacer de modo que no se invoque a la consola de Python, sino que se ejecute el archivo Python sin mostrar la consola.

```java
package edu.fduarte_Fourier.python.console;
import android.app.Application;
import edu.fduarte_Fourier.python.utils.PythonConsoleActivity;

public class EjecutarSerieFourier extends PythonConsoleActivity {

    @Override protected Class<? extends Task> getTaskClass() {
        return Task.class;
    }

    public static class Task extends PythonConsoleActivity.Task {
        public Task(Application app) {
            super(app);
        }

        @Override public void run() {
            py.getModule("seriefourier").callAttr("seriefourier");
        }
    }
}
```

A través de esta librería, las variables en las actividades de Java, se pueden comunicar con las variables de los archivos de Python. Esto se logra, creando una clase en Java y declarando las variables globales que se utilizaran para llevar y traer valores entre estas dos plataformas.
A continuación, se muestra una parte del código de la clase creada para este fin:

```
package edu.fduarte_Fourier.python.console;

public class DatosGraficaSF {

    //SELECCION DEL MENU
    public static int f1t = 0;
    public static int f2t = 0;
    public static int f3t = 0;
    public static int ED = 0;


    //FUNCION A TROZOS A INTRODUCIR
    public static String fx1;
    public static String fx2;
    public static String limI;
    public static String limM;
    public static String limS;
    public static String Armonicos;

   …
}
```

En el archivo Python, se escribe la siguiente línea, la cual, importa las variables de la clase Java para que estas tengan efecto en el código Python, y de esta manera, puedan proveer los datos que le manda Java, trabajarlos en Python y después regresar otros valores a Java.

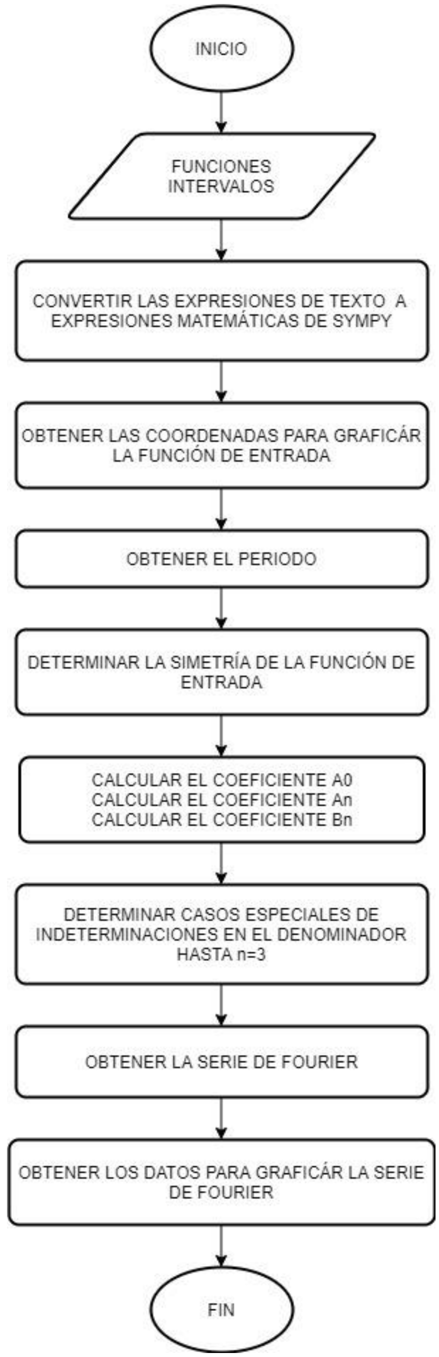
```
from edu.fduarte_Fourier.python.console import DatosGraficaSF as SF
fdx1=SF.fx1
```

Para regresar un valor a Java se utiliza la siguiente estructura:

```
SF.txtan01 = "Sustituyendo los límites de las integrales definidas:"
SF.an01 = a00
```

4.4 Etapas del desarrollo

Fig. 4.1. Etapas en la resolución del problema:



4.4.1 Entrada de datos.

En la interfaz del teléfono, el usuario teclea su expresión matemática para indicar una función matemática, así como los límites del intervalo donde es válida dicha función. Estos datos se capturan en Java y se almacenan en las variables globales de la clase creada para funcionar como enlace con Python.

Estas expresiones se tienen que mandar en forma de texto plano para no tener inconvenientes en Java. Al recibir estos datos en Python, lo primero es estructurar estas expresiones de tipo *string* en expresiones matemáticas pero, en este caso, deben de ser entendibles para SymPy.
La instrucción que se utilizó para convertir texto en expresión matemática fue: *sympify*.

"*Convierte una expresión arbitraria en un tipo que se puede usar dentro de SymPy.*
Por ejemplo, convertirá Python ints en instancias de sympy.Integer, floats en instancias de sympy.Float, etc. También puede forzar expresiones simbólicas que heredan de Basic. Esto puede ser útil en cooperación con SAGE." (SymPy Development Team, 2020).

La sintaxis es:

sympify(a, locals=None, convert_xor=True, strict=False, rational=False, evaluate=None)

Para la aplicación solo se utilizó el argumento evaluate=None ya que queremos trabajar con calculo simbólico y no con resultados numéricos.
Var = sympify (VarTexto, evaluate=False)

Las expresiones que se convirtieron a expresión SymPy fueron: las funciones de entrada, los límites de los intervalos y el número de armónicos.

En el caso de los límites de los intervalos, se creó una segunda variable para hacer la sustitución del valor de PI, en caso de que lo tuviera. Esto para hacer más fácil la generación de puntos para graficar la(s) función(es) de entrada.

4.4.2 Graficar la función de entrada

Para graficar la función de entrada, a cada trozo de la función se le asignan 30 valores como dominio en el intervalo en que es válido, luego se procede a calcular f(x) en cada punto dado. Como se está trabajando con expresiones simbólicas, no se pueden hacer cálculos matemáticos directos, y lo que se quiere hacer, es, obtener 30 valores de f(x), uno para cada uno del dominio asignado. La opción que maneja SymPy para hacer sustituciones de números por variables y evaluar la expresión, es la instrucción subs. Esto sería muy lento, y si la expresión tuviera el símbolo PI, se tendría que hacer otra sustitución, lo que generaría más código y más lentitud. Por este motivo, se decidió utilizar la función *lamdify*.

El propósito principal de esta función es proporcionar un puente desde las expresiones SymPy a las bibliotecas numéricas como NumPy, SciPy, NumExpr, mpmath y tensorflow. En general, las funciones SymPy no funcionan con objetos de otras bibliotecas, como matrices NumPy y las funciones de bibliotecas numéricas como NumPy o mpmath no funcionan en expresiones SymPy. *Lambdify* une los dos al convertir una expresión SymPy en una función numérica equivalente, como en este caso, una expresión numérica en NumPy.

Esta función tiene una gran ventaja, que es mucho más rápida que un ciclo for.

```
lam_f = lambdify(x, fdex1, 'numpy')
y_t1 = lam_f(x_t1)
```

En las funciones de entrada para desarrollar una Serie de Fourier, es normal utilizar funciones constantes. En estos casos, la función *lamdify* falla, ya que está construida para trabajar con valores de *x*. En este caso, antes de generar los valores de *f(x)*, primero se tiene que saber si la función es constante, en caso de que así lo sea, no se utiliza *lamdify* y a cada valor del dominio se le asigna la constante.

Se generan los pares ordenados de las funciones a trozos utilizadas en el mismo número de variables, para, de esta manera, devolverlas a la clase Java y elaborar la gráfica en la actividad de resultados.

4.4.3 Obtener el período de la función de entrada.

El período donde es válida la función de entrada, ya sea continua en un intervalo, o discontinua en varios intervalos, es muy importante, ya que lo vamos a utilizar en las fórmulas para obtener el desarrollo de la Serie de Fourier.

En este caso, al límite inferior se le cambia de signo y se le suma al límite superior.

4.4.4 Determinar la simetría de la función de entrada.

Determinar la simetría de la función ayuda para ahorrar cálculos innecesarios, como ya se vio anteriormente, si la función es Par entonces la serie de senos será igual a cero, por lo que solo se calculan los coeficientes a_0 y a_n, si la función es impar, la serie de cosenos es igual a cero, por lo que solo se calcula el coeficiente b_n. Solo en caso de que la función no tenga simetría par o impar, se calculan los tres coeficientes.

Para obtener la simetría de la función se parte de la definición de simetría:

$$\begin{aligned} Simetría\ Par&: \quad f(-x) = f(x) \\ Simetría\ Impar&: \quad f(-x) = -f(x) \end{aligned}$$

Antes de evaluar y comparar cualquier par ordenado, se hace la validación de la diferencia entre el límite superior y el inferior. Si la diferencia es cero, entonces procedemos a verificar la simetría, en caso contrario, sabemos de antemano que la función o funciones de entrada no tienen simetría, ya que el intervalo dado no es simétrico.

Si la diferencia es cero, se procede a evaluar la función de entrada en 15 valores definidos dentro del intervalo válido de la función de entrada, esto para asegurar la validación en las funciones de entrada de tres trozos.

Se obtienen los pares ordenados y se procede a aplicar las definiciones de simetría, para esto, se toman dos valores en los que se asegura que se apliquen las definiciones de simetría.

En el caso de la función continua, se utiliza el mismo procedimiento para obtener los valores para la gráfica. En el caso de las funciones a trozos, la función *Lambdify* se acompaña con la función *Piecewise*, esto es para obtener los pares ordenados de los diferentes intervalos de cada trozo de función en una solo instrucción, por ejemplo, para una función de dos trozos, tenemos las siguientes instrucciones:

```
x_tpp = linspace(lii, lss, 11)
lam_fpp = lambdify(x, Piecewise((fdex1, x < lmm), (fdex2, x >= lmm)), "numpy")
y_tpp = lam_fpp(x_tpp)
```

Para comprobar su simetría Par, tenemos la siguiente instrucción:

```
if round(y_tpp[2],2) == round(y_tpp[8],2):
…
```

En este caso, para comprobar su simetría Impar, tenemos la siguiente instrucción:

```
elif round(y_tpp[2],2) == -round(y_tpp[8],2):
…
```

Las posiciones 2 y 8, son valores de *x*, iguales en magnitud y de diferente signo, esto hace posible evaluar la simetría.

4.4.5 Cálculo de los coeficientes.

Cálculo del coeficiente a_0.

Para calcular este coeficiente, se valida que la función tenga simetría Par o que no tenga simetría. Esta parte del algoritmo se divide en 3 partes, para que, como ya se ha comentado, tenga fidelidad con el trabajo del estudiante y del docente en el salón de clase. Estas partes son las siguientes:

1. "Planteando el coeficiente a_0". En esta sección, únicamente se convierten a Látex las variables que intervienen en el cálculo de a_0, como se muestra a continuación:

```
a0 = "$$a_0 = \\frac{2}{" + latex(p) + "} \\int\\limits_{" + latex(a) + "}^{" + latex(b) + "} " +
latex(fdex1) + "\\dx$$"
```

El plantearlo en Latex, es para que la librería MathView pueda representarlo en la interfaz gráfica del móvil.

2. "Integrando…". En esta sección, integramos de manera indefinida, para hacer el planteamiento de la integral y de los limites en que debe ser evaluada.
Aquí, se utiliza la función *integrate* de *SymPy*, la cual nos devuelve una expresión matemática simbólica del resultado.
Su sintaxis es:

```
sympy.integrate(expression, reference variable)
```

En este caso, se hizo la integración indefinida de la siguiente integral:

$$a0 = \frac{2}{p}\int fd1\,dx$$

La instrucción utilizada es:

```
i1 = integrate(fd1,x, conds='none')
```

Se muestra la integral indefinida con los límites a evaluar en Latex:

```
i1 = "$$a_0 = \\frac{2}{" + latex(p) + "} \\left[" + latex(i1) + "\\right]_{" + latex(a) + "}^{" + latex(b) +
"$$"
```

3. "Sustituyendo los límites de las integrales definidas". Como siguiente paso, se vuelve a evaluar la función, pero ahora aplicando los límites establecidos por el usuario, después de integrar se simplifica el resultado para hacerlo más presentable.

Continuando con la función *integrate*, modificamos su sintaxis de la siguiente manera:

```
integrate(expression, (reference variable, lower limit, upper limit))
```

En este caso, se hizo la integración definida de la siguiente integral:

$$a_0 = \frac{2}{p}\int_a^b fd1\ dx$$

La instrucción utilizada es:

```
a0 = together(simplify(integrate((2 / p) * fd1, (x, a, b), conds='none')))
```

Se muestra la integral definida

```
a0 = "$$a_{0} = " + latex(a0) + "$$"
```

Simplify.

Una de las características más útiles de un sistema de manipulación simbólica es la capacidad de simplificar expresiones matemáticas. *SymPy* tiene varias funciones para realizar varios tipos de simplificación. También hay una función general llamada *simplify*() que intenta aplicar todas estas funciones de manera inteligente para llegar a la forma más simple de una expresión.

Su sintaxis es:

Simplify(expresión)

Together

Esta función toma una expresión o un conjunto de expresiones y las junta (combinando subexpresiones racionales). *Together*() puede preservar tanto como sea posible la estructura de la expresión de entrada en la salida.

Sintaxis:

```
together(expr, deep=False, fraction=True)
```

Cálculo del coeficiente a_n.

Para calcular este coeficiente, se valida que la función tenga simetría Par o que no tenga simetría.

Esta parte del algoritmo se divide en 5 partes, para que, como ya se ha comentado, tenga fidelidad con el trabajo del estudiante y del docente en el salón de clase. Estas partes son las siguientes:

Las primeras tres partes son las mismas que para calcular el coeficiente a_0, por lo que solo se nombrarán:

1. “Planteando el coeficiente a_n”.
2. “Integrando…”.
3. "Sustituyendo los límites de las integrales definidas".

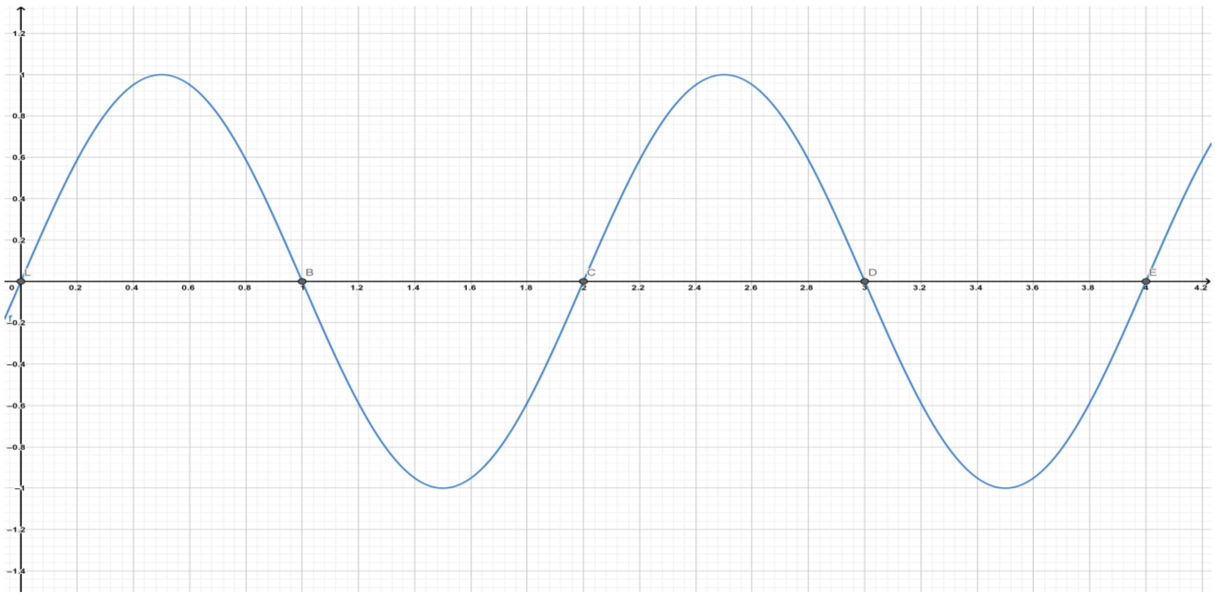
En este caso, se hizo la integración definida de la siguiente integral:

$$a_n = \frac{2}{p}\int_a^b \cos\left(\frac{2\pi n x}{p}\right) fd1\, dx$$

4. “Simplificando senos”. El resultado de esta integración, en muchos casos, vamos a tener senos en la expresión. Estos senos pueden tener en su argumento la constante PI, la cual nos da un comportamiento predictivo. Como se sabe, la función seno es una función senoidal con periodo repetitivo, según se puede observar en la gráfica 1, la cual tiene como característica la intersección con el eje x, en donde el valor del seno es igual a 0. Para simplificar la expresión que nos resulta de la integración, se procede a sustituir con 0 aquellos senos en donde se dé la intersección con el eje x. Los cuales serían:

sen(π) = 0, sen(2π) = 0, sen(3π) = 0, sen(4π) = 0, sen(5π) = 0, etc.

Gráfica 1. Función sen(nπ).



Fuente: Elaboración propia

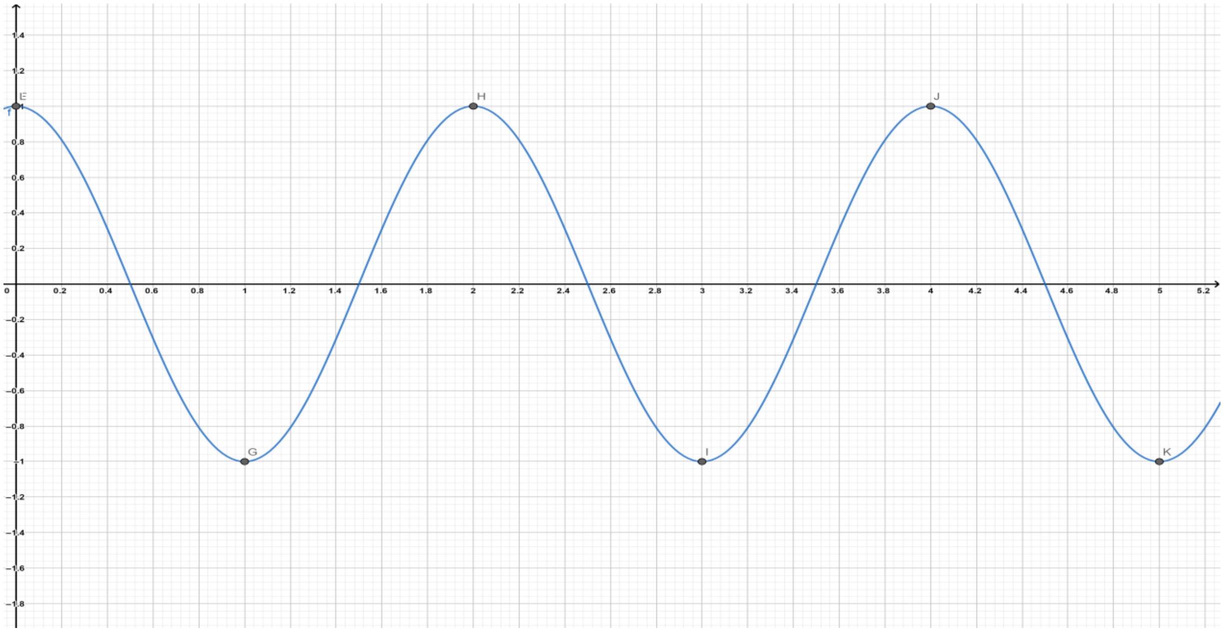
5. “Simplificando cosenos”. El resultado de esta integración, en muchos casos, vamos a tener cosenos en la expresión. Estos cosenos pueden tener en su argumento la constante PI, la cual nos da un comportamiento predictivo. Como se sabe, la función coseno es una función senoidal con periodo repetitivo, según se puede observar en la gráfica 2, la cual tiene como característica que la máxima amplitud obtenida es 1 o -1. Para simplificar la expresión que nos resulta de la integración, se procede a sustituir aquellos cosenos en donde se obtenga dicho valor. Los cuales serían:

$\cos(\pi) = -1$, $\cos(3\pi) = -1$, $\cos(5\pi) = -1$, $\cos(7\pi) = -1$, etc.

Por otro lado:

$\cos(2\pi) = 1$, $\cos(4\pi) = 1$, $\cos(6\pi) = 1$, $\cos(8\pi) = 1$, etc.

Gráfica 2. Función cos(nπ)



Fuente: elaboración propia

Por último, se vuelve a simplificar la expresión final.

Cálculo del coeficiente b_n.

Para calcular este coeficiente, se valida que la función tenga simetría Impar o que no tenga simetría.

Esta parte del algoritmo se divide en 5 partes, para que, como ya se ha hablado, tenga fidelidad con el trabajo del estudiante y del docente en el salón de clase.

Estas partes son las mismas que para calcular el coeficiente a_n, por lo que solo se nombrarán:

1. "Planteando el coeficiente b_n".
2. "Integrando…".
3. "Sustituyendo los límites de las integrales definidas".

En este caso, se hizo la integración definida de la siguiente integral:

$$b_n = \frac{2}{p}\int_a^b \sin\left(\frac{2\pi n x}{p}\right) fd1\; dx$$

4. “Simplificando senos”.
5. “Simplificando cosenos”.

Valor Absoluto.

Cuando la función de entrada es un valor absoluto, *SymPy* no puede hacer la integración definida, solo la indefinida. Por lo que al momento de evaluar la función en los siguientes pasos o produce error o da el valor de cero, haciendo con esto que el resultado de la serie sea equivocado.

Para resolver esto último, se hizo un cambio de la variable *x* por *w* en la integral a trabajar, esta nueva variable *w* es de tipo real, solo de esta manera, *SymPy* puede realizar los cálculos del valor absoluto, dando como resultado una función definida por partes, la cual puede ser evaluada en los siguientes pasos.

Después de efectuar la integración, se procede a cambiar la variable auxiliar *w* por la variable *x*.

Hasta aquí, hemos planteado el desarrollo de una serie de Fourier de una función dada. Este algoritmo es considerado para resolver de manera general la serie de Fourier, considerando que no tenga excepciones

A continuación, se presenta un ejemplo de desarrollo de serie de Fourier. Este ejercicio está tomado de (Zill & Warren, 2015) de la página 421, ejercicio 3. Su solución se encuentra en la página RES-18.

$$f(x) = \begin{cases} 1, & -1 \leq x \leq 0 \\ x, & 0 < x \leq 1 \end{cases}$$

Los valores en la aplicación se introducen de forma separada:

F1(x) = 1

F2(x) = x

Límite a = -1

Límite b = 0

Límite c = 1

Armónicos = 40

El algoritmo convierte los datos de entrada en una expresión látex para que se pueda mostrar la función a trozos de entrada como inicio del desarrollo de la serie de Fourier.

```
$$F(x) = \begin{cases}1 & \text{si } & -1\leq \text{x } \leq0\\x & \text{si } & 0\leq \text{x } \leq1\end{cases}$$
```

$$f(x) = \begin{cases} 1, & -1 \leq x \leq 0 \\ x, & 0 < x \leq 1 \end{cases}$$

El algoritmo calcula los pares ordenados para la gráfica de la función 1.

[-1. -0.96551724 -0.93103448 -0.89655172 -0.86206897 -0.82758621 -0.79310345 -0.75862069 -0.72413793 -0.68965517 -0.65517241 -0.62068966 -0.5862069 -0.55172414 -0.51724138 -0.48275862 -0.44827586 -0.4137931 -0.37931034 -0.34482759 -0.31034483 -0.27586207 -0.24137931 -0.20689655 -0.17241379 -0.13793103 -0.10344828 -0.06896552 -0.03448276 0.]

[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

El algoritmo calcula los pares ordenados para la gráfica de la función 2.

[0. 0.03448276 0.06896552 0.10344828 0.13793103 0.17241379 0.20689655 0.24137931 0.27586207 0.31034483 0.34482759 0.37931034 0.4137931 0.44827586 0.48275862 0.51724138 0.55172414 0.5862069 0.62068966 0.65517241 0.68965517 0.72413793 0.75862069 0.79310345 0.82758621 0.86206897 0.89655172 0.93103448 0.96551724 1.]

[0. 0.03448276 0.06896552 0.10344828 0.13793103 0.17241379 0.20689655 0.24137931 0.27586207 0.31034483 0.34482759 0.37931034 0.4137931 0.44827586 0.48275862 0.51724138 0.55172414 0.5862069 0.62068966 0.65517241 0.68965517 0.72413793 0.75862069 0.79310345 0.82758621 0.86206897 0.89655172 0.93103448 0.96551724 1.]

Con estos datos se grafica la función a trozos para confirmar visualmente que estamos trabajando con la función de entrada correcta

Calcula el periodo:

Periodo = 2

Después hace una diferencia entre los límites establecidos de los intervalos en el dominio de las funciones de entrada. Si la diferencia es cero, se procede a establecer si tiene simetría o no. Si la diferencia es diferente de cero, se da por hecho que no tiene simetría.

Diferencia de límites = 0

Se procede a detectar si tiene simetría.

Se generan 11 números equidistantes entre los límites, y después se evalúan en las funciones de entrada para generar una tabla de coordenadas.

Num	*x*	*f(x)*
0	-1.0	1.0
1	-0.8	1.0
2	-0.6	1.0
3	-0.4	1.0
4	-0.2	1.0
5	0.0	0.0
6	0.2	0.2
7	0.4	0.4
8	0.6	0.6
9	0.8	0.8
10	1.0	1.0

Se procede a hacer la comparación de dos coordenadas de la tabla para comprobar mediante la definición de simetría par y simetría impar si es que existe alguna de ellas. En este caso se tiene que:

La Función f(x) no tiene simetría

“Una función f(x) no tiene simetría si no cumple con la simetría Par o Impar. Al tener una función sin simetría, es necesario calcular todos los coeficientes: a0, an y bn.”

Planteando el coeficiente a_0.

Se plantea la integral.

Se genera el siguiente código látex:

```
$$a_0 = \frac{2}{2} \int\limits_{-1}^0 1\,dx + \frac{2}{2} \int\limits_{0}^1 x\,dx$$
```

Para mostrar lo siguiente:

$$a_0 = \frac{2}{2} \int_{-1}^{0} 1\,dx + \frac{2}{2} \int_{0}^{1} x\,dx$$

Se plantea el resultado de la integral indefinida con los valores de sustitución:

```
$$a_0 = \frac{2}{2} \left[x\right]_{-1}^{0} + \frac{2}{2} \left[\frac{x^{2}}{2}\right]_{0}^{1}$$
```

Donde se muestra la siguiente:

$$a_0 = \frac{2}{2} [x]_{-1}^{0} + \frac{2}{2} \left[\frac{x^{2}}{2}\right]_{0}^{1}$$

Sustituyendo el límite de las integrales:

```
$$a_0 = \left[1\right] + \left[\frac{1}{2}\right]$$
```

Donde se muestra lo siguiente:

$$a_0 = 1 + \frac{1}{2}$$

Se muestra el resultado de a_0:

```
$$a_0 = \frac{3}{2} $$
```

$$a_0 = \frac{3}{2}$$

Planteando el coeficiente a_n.

Se plantea la integral.

Se genera el siguiente código látex:

```
$$a_n = \frac{2}{2} \int\limits_{-1}^0 [1]\cos\left(\frac{2*n*x*\pi}{p}\right)\,dx + \frac{2}{2}
\int\limits_{0}^1 [x]\cos\left(\frac{2*n*x*\pi}{p}\right)\,dx$$
```

Para mostrar lo siguiente:

$$a_n = \frac{2}{2}\int_{-1}^{0}[1]\cos\left(\frac{2n\pi x}{p}\right)\,dx + \frac{2}{2}\int_{0}^{1}[x]\cos\left(\frac{2n\pi x}{p}\right)\,dx$$

Se plantea el resultado de la integral indefinida con los valores de sustitución:

$$a_n = \frac{2}{2} \left[\frac{\sin{\left(\pi n x \right)}}{\pi n}\right]_{-1}^{0} + \frac{2}{2} \left[\frac{x \sin{\left(\pi n x \right)}}{\pi n} + \frac{\cos{\left(\pi n x \right)}}{\pi^{2} n^{2}}\right]_{0}^{1}$$

Donde se muestra la siguiente:

$$a_n = \frac{2}{2}\left[\frac{\sin(\pi n x)}{\pi n}\right]_{-1}^{0} + \frac{2}{2}\left[\frac{x\sin(\pi n x)}{\pi n} + \frac{\cos(\pi n x)}{\pi^2 n^2}\right]_{0}^{1}$$

Sustituyendo el límite de las integrales:

$$a_n = \left[\frac{\sin{\left(\pi n \right)}}{\pi n}\right] + \left[\frac{\pi n \sin{\left(\pi n \right)} + \cos{\left(\pi n \right)} - 1}{\pi^{2} n^{2}}\right]$$

Donde se muestra lo siguiente:

$$a_n = \left[\frac{\sin \pi n}{\pi n}\right] + \left[\frac{\pi n \sin(\pi n) + \cos(\pi n) - 1}{\pi^2 n^2}\right]$$

Este sería el resultado final del coeficiente a_n, pero aplicamos los valores de seno conocidos en donde sabemos que es igual a cero.

sen(π) = 0, sen(2π) = 0, sen(3π) = 0, …, por lo tanto, podemos asumir que sen(nπ) = 0, haciendo la sustitución:

$$a_n = \left[0\right] + \left[\frac{\cos{\left(\pi n \right)} - 1}{\pi^{2} n^{2}}\right]$$

$$a_n = [0] + \left[\frac{\cos(\pi n) - 1}{\pi^2 n^2}\right]$$

Ahora toca el caso de sustituir los cosenos en donde se iguala a 1 o -1 según sea el caso cos(π) = -1, cos(2π) = 1, cos(3π) = -1, ..., por lo tanto, podemos asumir que: cos(nπ) = $(-1)^n$, haciendo la sustitución:

$$a_n = \left[0\right] + \left[\frac{\left(-1\right)^{n} - 1}{\pi^{2} n^{2}}\right]$$

$$a_n = [0] + \left[\frac{(-1)^n - 1}{\pi^2 n^2}\right]$$

Se muestra el resultado de a_n:

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

$$a_n = \frac{(-1)^n - 1}{\pi^2 n^2}$$

Planteando el coeficiente b_n.

Se plantea la integral.

Se genera el siguiente código látex:

$$b_n = \frac{2}{2} \int\limits_{-1}^0 \left[1\right]\sin\left(\frac{2 \pi nx}{p}\right)\,dx + \frac{2}{2} \int\limits_{0}^1 \left[x\right]\sin\left(\frac{2 \pi nx}{p}\right)\,dx$$

Para mostrar lo siguiente:

$$b_n = \frac{2}{2}\int_{-1}^{0}[1]\sin\left(\frac{2n\pi x}{p}\right)\,dx + \frac{2}{2}\int_{0}^{1}[x]\sin\left(\frac{2n\pi x}{p}\right)\,dx$$

Se plantea el resultado de la integral indefinida con los valores de sustitución:

$$b_n = \frac{2}{2} \left[- \frac{\cos{\left(\pi n x \right)}}{\pi n}\right]_{-1}^{0} + \frac{2}{2} \left[- \frac{x \cos{\left(\pi n x \right)}}{\pi n} + \frac{\sin{\left(\pi n x \right)}}{\pi^{2} n^{2}}\right]_{0}^{1}$$

Donde se muestra la siguiente:

$$b_n = \frac{2}{2}\left[-\frac{\cos(\pi n x)}{\pi n}\right]_{-1}^{0} + \frac{2}{2}\left[\frac{x\cos(\pi n x)}{\pi n} + \frac{\sin(\pi n x)}{\pi^2 n^2}\right]_{0}^{1}$$

Sustituyendo el límite de las integrales:

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

Donde se muestra lo siguiente:

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

Este sería el resultado final del coeficiente b_n, pero aplicamos los valores de seno conocidos en donde sabemos que es igual a cero.

sen(π) = 0, sen(2π) = 0, sen(3π) = 0, …, por lo tanto, podemos asumir que sen(nπ) = 0, haciendo la sustitución:

$$b_n = \left[\frac{\cos{\left(\pi n \right)} - 1}{\pi n}\right] + \left[- \frac{\cos{\left(\pi n \right)}}{\pi n}\right]$$

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

Ahora toca el caso de sustituir los cosenos en donde se iguala a 1 o -1 según sea el caso cos(π) = -1, cos(2π) = 1, cos(3π) = -1, ..., por lo tanto, podemos asumir que: cos(nπ) = $(-1)^n$, haciendo la sustitución:

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

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

Se muestra el resultado de a_n:

$$b_n = -\frac{1}{\pi n}$$

$$b_n = -\frac{1}{\pi n}$$

Planteando la serie de Fourier:

$$F(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos \left(\frac{n \pi}{p}x \right) + b_n sin \left(\frac{n \pi}{p}x \right) $$

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\alpha} \left[a_n \cos\left(\frac{n\pi}{p} x\right) + b_n \sin\left(\frac{n\pi}{p} x\right)\right]$$

Se obtiene la serie de Fourier:

$$F(x) = \frac{3}{4} + \sum_{n=1}^{\infty} \frac{\left(\left(-1\right)^{n} - 1\right) \cos{\left(\pi n x \right)}}{\pi^{2} n^{2}} + - \frac{\sin{\left(\pi n x \right)}}{\pi n}$$

$$f(x) = \frac{3}{4} + \sum_{n=1}^{\alpha} \left(\frac{((-1)^n - 1)\ \cos(n\pi x)}{\pi^2 n^2} - \frac{\sin(n\pi x)}{\pi n} \right)$$

La respuesta a este ejercicio mostrada en el libro de Ecuaciones Diferenciales, es la siguiente:

$$f(x) = \frac{3}{4} + \sum_{n=1}^{\alpha} \left\{ \frac{((-1)^n - 1)}{\pi^2 n^2} \cos(n\pi x) - \frac{1}{\pi n} \sin(n\pi x) \right\}$$

4.4.6 Casos especiales

En ocasiones, el denominador de alguno de los coeficientes de la serie de Fourier, a_n o b_n, pueden ser igual a cero cuando se sustituye algún número de la sumatoria de Fourier.

Si se intenta graficar esta sumatoria, nos daría un error de división entre cero.

En estos casos se procede a identificar en cual numero de la sumatoria se indetermina el coeficiente de Fourier, para obtener ese número y los menores a él de la sumatoria, integrarlos y presentarlos junto con el coeficiente a_0, fuera de la sumatoria.

Como ejemplo, se toma el ejercicio 23 de la serie de ejercicios 11.3 de la página 427 de Zill & Warren, (2015)

En el cual, se quiere desarrollar la serie de Fourier de la siguiente función:

$$f(x) = |\sin(x)| \quad si \quad -\pi \leq x \leq \pi$$

Empezando con el algoritmo, se tiene que:

- El periodo de la función es: 2π.
- La función tiene simetría par.

Con estos datos, se procede a obtener los coeficientes a_0 y a_n.

El coeficiente a_0 es igual a:

$$a_0 = \frac{4}{\pi}$$

El coeficiente a_n es igual a:

$$a_n = -\frac{2 - 2(-1)^{n+1}}{\pi(n^2 - 1)}$$

Al evaluar este coeficiente para n=1 hasta infinito, se tiene que:

Para n pares:

$$a_n = -\frac{2+2}{\pi(c1_n)} = -\frac{4}{c1_n} = -c2_n$$

Para n impares, excepto n = 1:

$$a_n = -\frac{2-2}{\pi(c_n)} = -\frac{0}{c_n} = 0$$

Se observa que el denominador es igual a cero cuando n es igual a 1

$para\ n = 1, tenemos\ que$:

$$a_1 = -\frac{2 - 2(-1)^{1+1}}{\pi(1^2 - 1)}$$

Haciendo los cálculos, se tiene que:

$$a_1 = -\frac{2 - 2}{\pi(0)} = -\frac{0}{0}$$

En este caso, se ve que, cuando n toma el valor de 1, el coeficiente se indetermina, ya que su denominador es igual a 0

Una vez que se detecta esta indeterminación, el algoritmo separa este coeficiente para n = 1 de la sumatoria de Fourier, y lo evalúa de manera independiente, para agregarlo en la serie de Fourier, pero fuera de la sumatoria.

Evaluando la función para n = 1, se tiene:

$$a_1 = \frac{1}{\pi}\int_{-\pi}^{\pi} |\sin(x)| \cos(x)\, dx$$

$$a_1 = 0$$

Por lo que la serie de Fourier se plantea de la siguiente manera:

$$f(x) = \frac{a_0}{2} + a_1 + \sum_{n=2}^{\alpha}(a_n \cos(nx))$$

Se sustituye el valor de $a_1 = 0$

$$f(x) = \frac{2}{\pi} + 0 + \sum_{n=2}^{\alpha}\left(-\frac{(2 - 2(-1)^{n+1}) \cos(nx)}{\pi(n^2 - 1)}\right)$$

En este caso, como $a_1 = 0$, el algoritmo no lo imprime y la serie de Fourier queda de la siguiente manera:

$$f(x) = \frac{2}{\pi} + \sum_{n=2}^{\alpha}\left(-\frac{(2 - 2(-1)^{n+1}) \cos(nx)}{\pi(n^2 - 1)}\right)$$

El resultado mostrado en la página RES-19 de la sección de soluciones de Zill & Warren, (2015) es el siguiente:

$$f(x) = \frac{2}{\pi} + \frac{2}{\pi}\sum_{n=2}^{\alpha} \frac{1+(-1)^n}{1-n^2}\cos(nx)$$

Como se puede apreciar, es el mismo resultado.

En otro ejemplo, se toma el ejercicio 9 de la serie de ejercicios 11.2 de la página 421 del libro de Zill & Warren, (2015).

En el cual, se quiere desarrollar la serie de Fourier de la siguiente función:

$$f(x) = \begin{cases} 0, & -\pi \leq x \leq 0 \\ sen(x), & 0 < x \leq \pi \end{cases}$$

Empezando con el algoritmo, se tiene que:

- El periodo de la función es: 2π.
- La función no tiene simetría.

Con estos datos, se procede a obtener los coeficientes $a_{0,}$ a_n, y b_n

El coeficiente a_0 es igual a:

$$a_0 = \frac{2}{\pi}$$

El coeficiente a_n es igual a:

$$a_n = -\frac{1-(-1)^{n+1}}{\pi(n^2-1)}$$

Al evaluar este coeficiente para n=1 hasta infinito, se tiene que:

Para n pares:

$$a_n = -\frac{1+1}{\pi(c1_n)} = -\frac{2}{c1_n} = -c2_n$$

Para n impares, excepto n = 1:

$$a_n = -\frac{1-1}{\pi(c_n)} = -\frac{0}{c_n} = 0$$

Se observa que el denominador es igual a cero cuando n es igual a 1, por lo que, para n=1, tenemos que:

$$a_1 = -\frac{1-(-1)^{1+1}}{\pi(1^2-1)}$$

Haciendo los cálculos, se tiene que:

$$a_1 = -\frac{1-1}{\pi(0)} = -\frac{0}{0}$$

Por último, el coeficiente b_n es igual a:

$$b_n = 0$$

En este caso, se ve que cuando n toma el valor de 1, el coeficiente a_1 se indetermina, ya que su denominador es igual a 0.

Una vez que se detecta esta indeterminación, el algoritmo separa este coeficiente para n = 1 de la sumatoria de Fourier, y lo evalúa de forma independiente, para agregarlo en la serie de Fourier, pero fuera de la sumatoria.

Evaluando la función para n = 1, se tiene:

$$a_1 = \frac{1}{\pi}\int_{-\pi}^{0}[0]\cos(x)\,dx + \frac{1}{\pi}\int_{0}^{\pi}[\sin(x)]\cos(x)\,dx$$

$$a_1 = 0$$

También se evalúa el coeficiente b_1, ya que la función de entrada no tiene simetría, aunque el coeficiente $b_n = 0$, se tiene que:

$$b_1 = \frac{1}{\pi}\int_{-\pi}^{0}[0]\sin(x)\,dx + \frac{1}{\pi}\int_{0}^{\pi}[\sin(x)]\sin(x)\,dx$$

$$b_1 = \frac{1}{2}$$

Por lo que la serie de Fourier se plantea de la siguiente manera:

$$f(x) = \frac{a_0}{2} + a_1\cos(x) + b_1\sin(x) + \sum_{n=2}^{\alpha}(a_n \cos(nx))$$

Se sustituye el valor de $a_1 = 0$ y $b_1 = \frac{1}{2}$

$$f(x) = \frac{2}{\pi} + 0 + \frac{\sin(x)}{2} + \sum_{n=2}^{\alpha}\left(-\frac{(1-(-1)^{1+1})\cos(x)}{\pi(1^2-1)}\right)$$

En este caso, como $a_1 = 0$, el algoritmo no lo imprime y la serie de Fourier queda de la siguiente manera:

$$f(x) = \frac{1}{\pi} + \frac{\sin(x)}{2} + \sum_{n=2}^{\alpha}\left(-\frac{(1-(-1)^{n+1})\ \cos(nx)}{\pi(n^2-1)}\right)$$

El resultado mostrado en la página RES-18 de la sección de soluciones del libro de Ecuaciones Diferenciales es el siguiente:

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

Como se puede apreciar, es el mismo resultado.

Otro ejemplo, tomado de la página web de (Bourne, 2018).

En el cual, se pide desarrollar la serie de Fourier de la siguiente función:

$$f(x) = \begin{cases} 0, & -1 \le x \le -\frac{1}{2} \\ \cos(3\pi x) & -\frac{1}{2} \le x \le \frac{1}{2} \\ 0, & \frac{1}{2} \le x \le 1 \end{cases}$$

Empezando con el algoritmo, se tiene que:

- El periodo de la función es: 2.
- La función tiene simetría par.

Con estos datos, se procede a obtener los coeficientes a_0 y a_n

El coeficiente a_0 es igual a:

$$a_0 = -\frac{2}{3\pi}$$

El coeficiente a_n es igual a:

$$a_n = \frac{6\cos\left(\frac{n\pi}{2}\right)}{\pi(n^2-9)}$$

Al evaluar este coeficiente para n=1 hasta infinito, se tiene que:

Para n pares, desde k = 1 hasta infinito:

$$a_n = \frac{6(-1)^k}{\pi(c1_k)}$$

Para n impares, excepto n = 3:

$$a_n = \frac{6(0)}{\pi(c_n)} = \frac{0}{c_n} = 0$$

Se observa que el denominador es igual a cero cuando n es igual a 3, por lo que, para n = 3 tenemos que:

$$a_3 = \frac{6(0)}{\pi(3^2 - 9)}$$

Haciendo los cálculos, se tiene que:

$$a_3 = \frac{0}{\pi(0)} = \frac{0}{0}$$

En este caso, se ve que, cuando n toma el valor de 3, el coeficiente a_1 se indetermina, ya que su denominador es igual a 0.

Una vez que se detecta esta indeterminación, el algoritmo separa los coeficientes para n = 1, n = 2 y n = 3 de la sumatoria de Fourier, y los evalúa de forma independiente, para agregarlos en la serie de Fourier, pero fuera de la sumatoria.

Evaluando la función para n = 1, se tiene:

$$a_1 = \frac{2}{2}\int_{-1}^{-\frac{1}{2}} [0]\cos(\pi x)\,dx + \frac{2}{2}\int_{-\frac{1}{2}}^{\frac{1}{2}} [\cos(3\pi x)]\cos(\pi x)\,dx + \frac{2}{2}\int_{\frac{1}{2}}^{1} [0]\cos(\pi x)\,dx$$

$$a_1 = 0$$

Evaluando la función para n = 2, se tiene:

$$a_2 = \frac{2}{2}\int_{-1}^{-\frac{1}{2}} [0]\cos(2\pi x)\,dx + \frac{2}{2}\int_{-\frac{1}{2}}^{\frac{1}{2}} [\cos(3\pi x)]\cos(2\pi x)\,dx + \frac{2}{2}\int_{\frac{1}{2}}^{1} [0]\cos(2\pi x)\,dx$$

$$a_2 = \frac{6}{5\pi}$$

Evaluando la función para n = 3, se tiene:

$$a_3 = \frac{2}{2}\int_{-1}^{-\frac{1}{2}} [0]\cos(3\pi x)\,dx + \frac{2}{2}\int_{-\frac{1}{2}}^{\frac{1}{2}} [\cos(3\pi x)]\cos(3\pi x)\,dx + \frac{2}{2}\int_{\frac{1}{2}}^{1} [0]\cos(3\pi x)\,dx$$

$$a_3 = \frac{1}{2}$$

Por lo que la serie de Fourier se plantea de la siguiente manera:

$$f(x) = \frac{a_0}{2} + a_1\cos(\pi x) + a_2\cos(2\pi x) + a_3\cos(3\pi x) + \sum_{n=4}^{\alpha}(a_n \cos(n\pi x))$$

Se sustituye el valor de $a_1 = 0$, $a_2 = \frac{6}{5\pi}$ y $a_3 = \frac{1}{2}$

$$f(x) = -\frac{1}{3\pi} + 0 + \frac{6\cos(2\pi x)}{5\pi} + \frac{\cos(3\pi x)}{2} + \sum_{n=4}^{\alpha}\left(\frac{6\cos\left(\frac{n\pi}{2}\right)\cos(n\pi x)}{\pi(n^2-9)}\right)$$

En este caso, como $a_1 = 0$, el algoritmo no lo imprime y la serie de Fourier queda de la siguiente manera:

$$f(x) = -\frac{1}{3\pi} + \frac{6\cos(2\pi x)}{5\pi} + \frac{\cos(3\pi x)}{2} + \sum_{n=4}^{\alpha}\left(\frac{6\cos\left(\frac{n\pi}{2}\right)\cos(n\pi x)}{\pi(n^2-9)}\right)$$

El resultado mostrado en la página web antes mencionada, es el siguiente:

$$f(x) = -\frac{1}{3\pi} + \frac{6\cos(2\pi x)}{5\pi} + \frac{\cos(3\pi x)}{2} + \sum_{n=4}^{\alpha} 6\frac{\cos\left(\frac{n\pi}{2}\right)}{\pi(-9+n^2)}\cos(n\pi x)$$

Como se puede apreciar, es el mismo resultado.

Este algoritmo está diseñado para detectar indeterminaciones hasta en n = 3, para hacer una buena presentación.

Capítulo V. Resultados y Discusión

Después de probar la aplicación en diferentes dispositivos móviles en el salón de clases de la materia de Ecuaciones Diferenciales del Instituto Tecnológico de Chihuahua II y de los datos que nos proporciona la tienda de Google Play, se obtuvieron los resultados esperados, como se describe a continuación.

1. La aplicación se ejecuta en teléfonos inteligentes con sistema operativo Android.
2. La aplicación está desarrollada en dos lenguajes:
    - El núcleo de la aplicación se ejecuta en Python
    - La interface de grafica de usuario, las gráficas, la adquisición de datos está desarrollada en Java para Android.

3. La aplicación no necesita de acceso a Internet para resolver los problemas que se le encomiendan gracias a la librería Chaquopy, la cual nos permite escribir y ejecutar código Python y enlazarlo con Java para Android de manera local en el dispositivo móvil.
4. La aplicación obtiene el desarrollo de Serie de Fourier con una fidelidad de casi el 100% de acuerdo a como se resuelve por un docente en clase
5. Ejecuta satisfactoriamente los diferentes casos que se le presentan.
6. La presentación de los coeficientes de la serie de Fourier, se presentan de forma sencilla con una gran fidelidad a los obtenidos en clase
7. La gráfica presentada se ajusta en un 100% a la gráfica esperada para cada caso propuesto.

Al ejecutar la aplicación, se muestra un menú con las tres opciones de funciones de entrada y un botón de ayuda para las instrucciones, figura 5.1.

Al escoger una opción del menú, en este caso la función continua, se muestra otra pantalla donde se piden los siguientes datos: función de entrada, límite inferior de intervalo, límite superior de intervalo y el número de armónicos para construir la gráfica, figura 5.2.

Figura 5.1. Menú de la aplicación

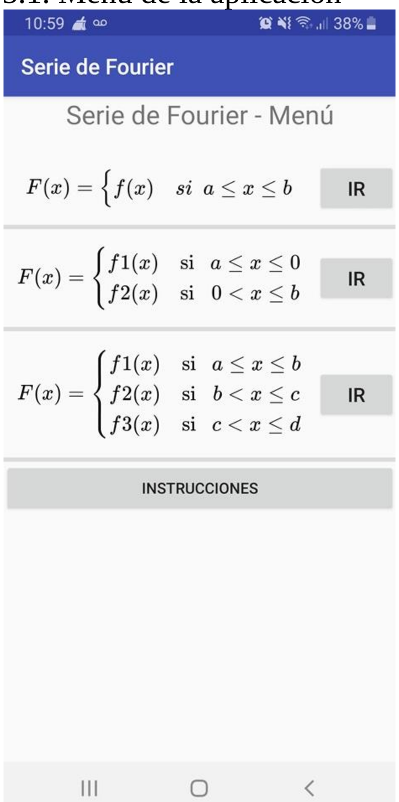


Figura 5.2. Ingresando datos

Al ingresar los datos y presionar el botón Resolver la serie de Fourier, aparece el mensaje: “Calculando…Sea paciente, puede tardar varios segundos”, figura 5.3.

Cuando la aplicación ya hizo los cálculos y está lista para mostrar los resultados, aparece el siguiente mensaje: “Listo para mostrar!!!”, figura 5.4.

Figura 5.3. Mensaje de espera

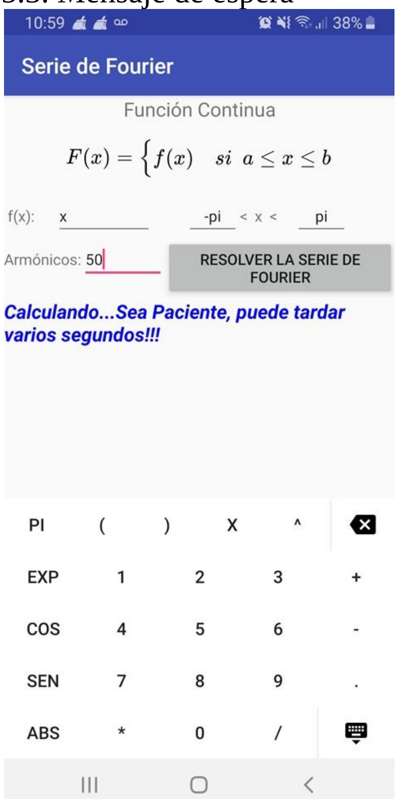
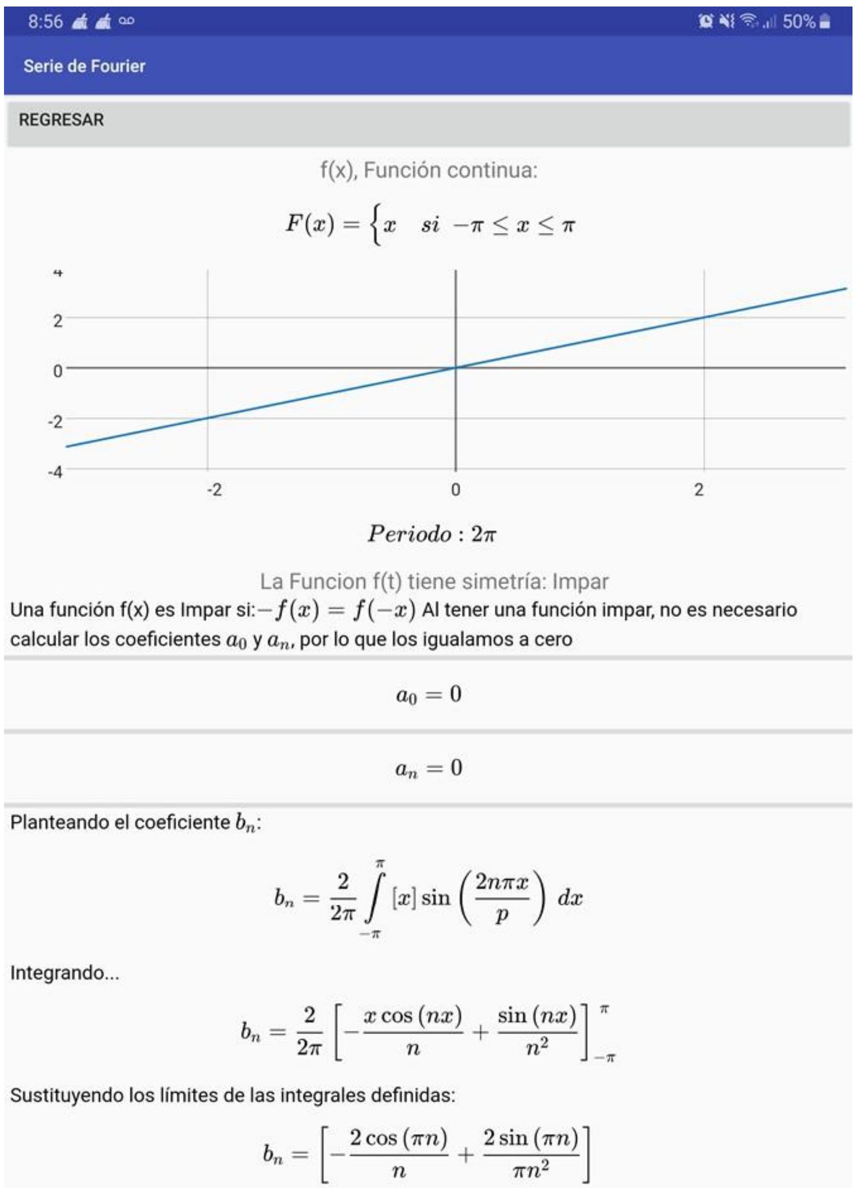


Figura 5.4. Mensaje de cálculos terminados

La aplicación muestra la función de entrada escrita por el usuario con lo finalidad de que el mismo esté seguro de que la función que tecleó es la misma que la aplicación está desarrollando. Muestra el periodo de la función y comunica el tipo de simetría que cuenta.

Empieza a desarrollar los coeficientes de la serie de Fourier, en este caso, como la función es Impar, los coeficientes de a no los calcula y los iguala a cero, figura 5.5.

Figura 5.5. Presentación de los resultados



Después de calcular los coeficientes, nos muestra la serie de Fourier obtenida para esta función. La gráfica se muestra con el número de armónicos que se pidió en el inicio y con un intervalo de 18 unidades, suficiente para mostrar la periodicidad de la función de entrada, figura 5.6.

Figura 5.6. Continuación de la presentación

sin(π) = 0, sin(2π) = 0, sin(3π) = 0, ..., por lo tanto, podemos asumir que: sin(nπ) = 0, haciendo la sustitución:

$$b_n = \left[-\frac{2\cos(\pi n)}{n}\right]$$

cos(π) = -1, cos(2π) = 1, cos(3π) = -1, ..., por lo tanto, podemos asumir que: cos(nπ) = $(-1)^n$, haciendo la sustitución:

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

Se muestra el resultado de b_n:

$$b_n = -\frac{2(-1)^n}{n}$$

La Serie de Fourier:

$$F(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos\left(\frac{n\pi}{p}x\right) + b_n sin\left(\frac{n\pi}{p}x\right)$$

Se obtiene la Serie de Fourier:

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

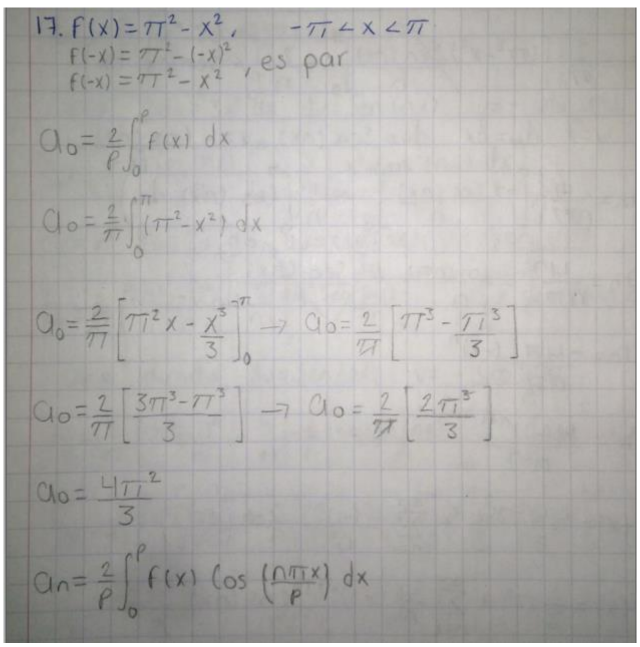
Se grafica la Serie de Fourier con 50 armónicos:

Para probar la veracidad de los resultados obtenidos por la aplicación y mostrar la fidelidad con el procedimiento que se desarrolla en el salón de clase, se presentan los siguientes casos:

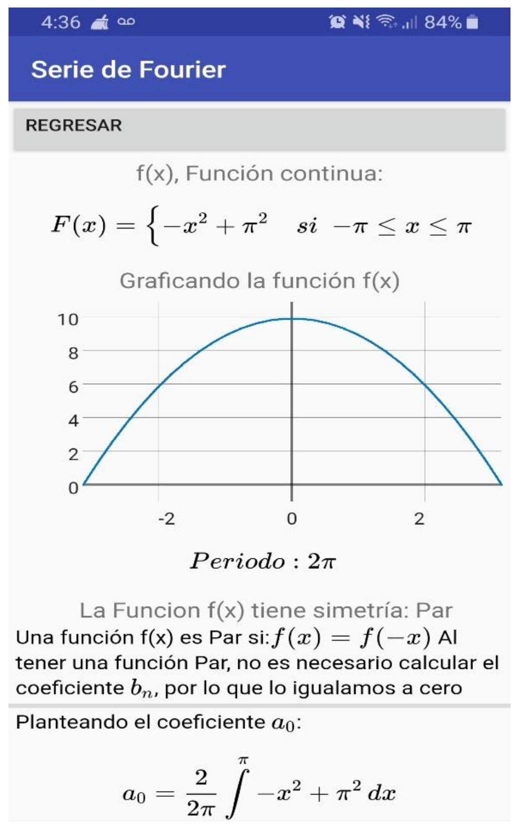
Caso 1. Función continua, figura 5.7.

Figura 5.7. Comparación de cálculos (Fidelidad)

a) Resolución en hoja del alumno



b) Resolución en pantalla de la aplicación



$$a_n = \frac{2}{\pi}\int_0^{\pi} (\pi^2 - x^2)\cos(nx)\,dx$$

$u = \pi^2 - x^2 \quad du = -2x\,dx \quad dv = \cos(nx)\,dx$

$$v = \frac{\operatorname{sen}(nx)}{n}$$

$$a_n = \frac{2}{\pi}\left(\left[\cancel{(\pi^2 - x^2)\frac{\operatorname{sen}(nx)}{n}}^{\,0}\right]_0^{\pi} + \frac{2}{n}\int_0^{\pi} x \operatorname{sen}(nx)\,dx\right)$$

$u = x \quad du = dx \quad dv = \operatorname{sen}(nx) \quad v = -\frac{\cos(nx)}{n}$

$$a_n = \frac{4}{n\pi}\left(\left[-x\frac{\cos(nx)}{n}\right]_0^{\pi} + \frac{1}{n}\int_0^{\pi} \cos(nx)\,dx\right)$$

$$a_n = \frac{4}{n\pi}\left(\frac{-\pi\cos(n\pi)}{n} + \left[\cancel{\frac{\operatorname{sen}(nx)}{n^2}}^{\,0}\right]_0^{\pi}\right)$$

$$a_n = \frac{-4\pi}{n^2\pi}(-1)^n$$

$$a_n = \frac{4(-1)^{n+1}}{n^2}$$

$$f(x) = \frac{4\pi^2}{3}\cdot\frac{1}{2} + \sum_{n=1}^{\infty}\frac{4(-1)^{n+1}}{n^2}\cos\left(\frac{n\pi x}{\pi}\right)$$

$$f(x) = \frac{2}{3}\pi^2 + \sum_{n=1}^{\infty}\frac{4}{n^2}(-1)^{n+1}\cos(nx)$$

Integrando...

$$a_0 = \frac{2}{2\pi}\left[-\frac{x^3}{3} + \pi^2 x\right]_{-\pi}^{\pi}$$

Sustituyendo los límites de las integrales definidas:

$$a_0 = \left[\frac{4\pi^2}{3}\right]$$

Se muestra el resultado de a_0:

$$a_0 = \frac{4\pi^2}{3}$$

Planteando el coeficiente a_n:

$$a_n = \frac{2}{2\pi}\int_{-\pi}^{\pi}[-x^2 + \pi^2]\cos\left(\frac{2n\pi x}{p}\right)\,dx$$

Integrando...

$$a_n = \frac{2}{2\pi}\left[-\frac{x^2\sin(nx)}{n} + \frac{\pi^2\sin(nx)}{n} - \frac{2x\,\cdots}{}\right.$$

Sustituyendo los límites de las integrales definidas:

$$a_n = \left[\frac{4(-\pi n\cos(\pi n) + \sin(\pi n))}{\pi n^3}\right]$$

sin(π) = 0, sin(2π) = 0, sin(3π) = 0, ..., por lo tanto, podemos asumir que: sin(nπ) = 0, haciendo la sustitución:

$$a_n = \left[-\frac{4\cos(\pi n)}{n^2}\right]$$

cos(π) = -1, cos(2π) = 1, cos(3π) = -1, ..., por lo tanto, podemos asumir que: cos(nπ) = $(-1)^n$, haciendo la sustitución:

$$a_n = \left[-\frac{4(-1)^n}{n^2}\right]$$

Se muestra el resultado de a_n:

$$a_n = -\frac{4(-1)^n}{n^2}$$

$$b_n = 0$$

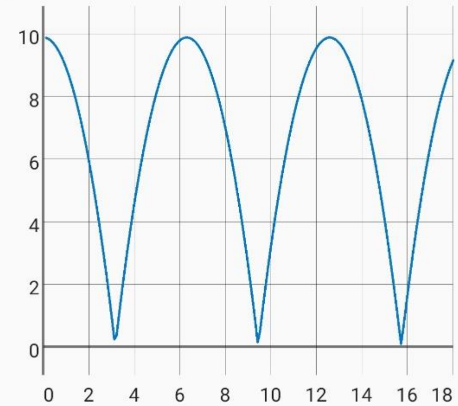
La Serie de Fourier:

$$F(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos\left(\frac{n\pi}{p}x\right) + b_n sin\left(\frac{n}{1}\right.$$

Se obtiene la Serie de Fourier:

$$F(x) = \frac{2\pi^2}{3} + \sum_{n=1}^{\infty} - \frac{4\,(-1)^n \cos(nx)}{n^2}$$

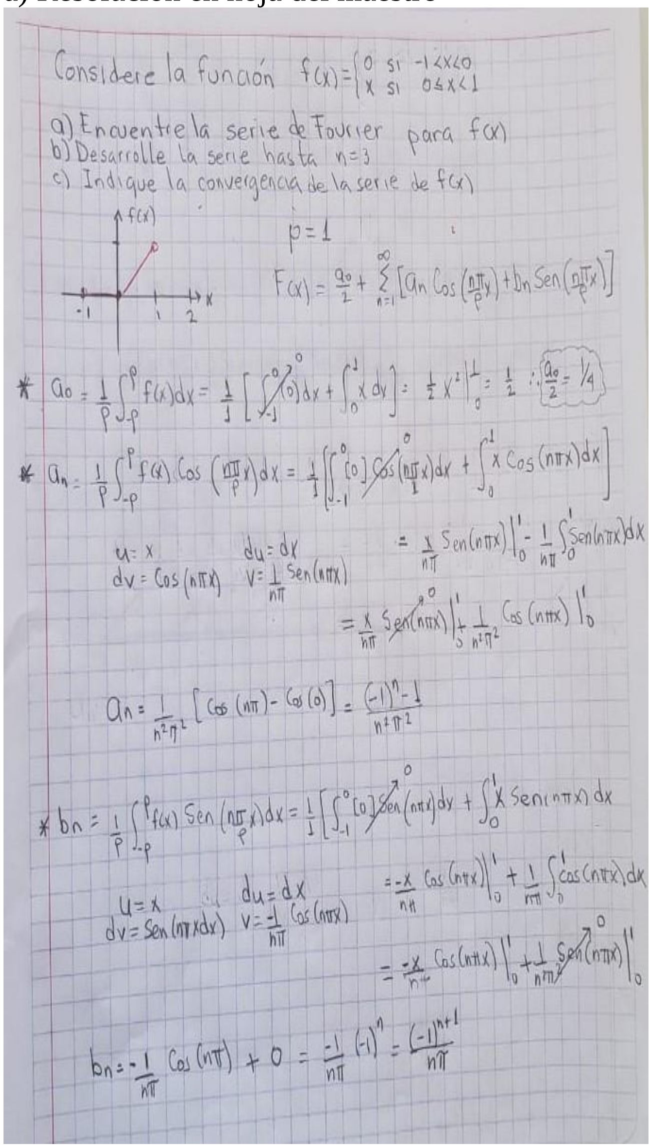
Se grafica la Serie de Fourier con 50 armónicos:



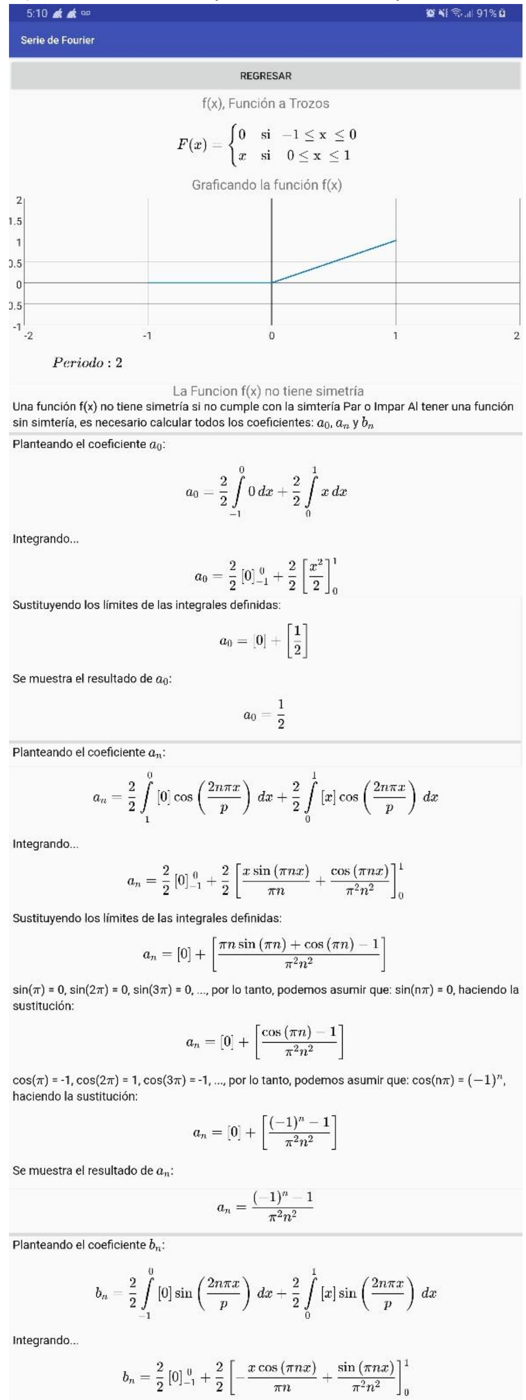
El siguiente caso es para una función de 2 trozos, figura 5.8:

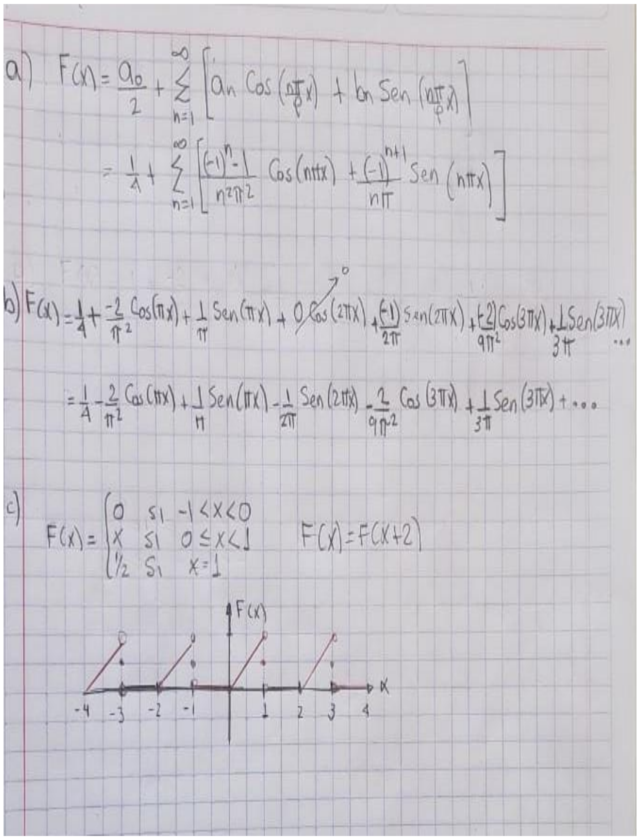
Figura 5.8. Comparación de cálculos (Fidelidad)

a) Resolución en hoja del maestro



b) Resolución en pantalla de la aplicación





Sustituyendo los límites de las integrales definidas:

$$b_n = [0] + \left[\frac{-\pi n \cos(\pi n) + \sin(\pi n)}{\pi^2 n^2}\right]$$

sin(π) = 0, sin(2π) = 0, sin(3π) = 0, ..., por lo tanto, podemos asumir que: sin(nπ) = 0, haciendo la sustitución:

$$b_n = [0] + \left[-\frac{\cos(\pi n)}{\pi n}\right]$$

cos(π) = -1, cos(2π) = 1, cos(3π) = -1, ..., por lo tanto, podemos asumir que: cos(nπ) = $(-1)^n$, haciendo la sustitución:

$$b_n = [0] + \left[-\frac{(-1)^n}{\pi n}\right]$$

Se muestra el resultado de b_n:

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

Planteando la Serie de Fourier:

$$F(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos\left(\frac{n\pi}{p}x\right) + b_n sin\left(\frac{n\pi}{p}x\right)$$

Se obtiene la Serie de Fourier:

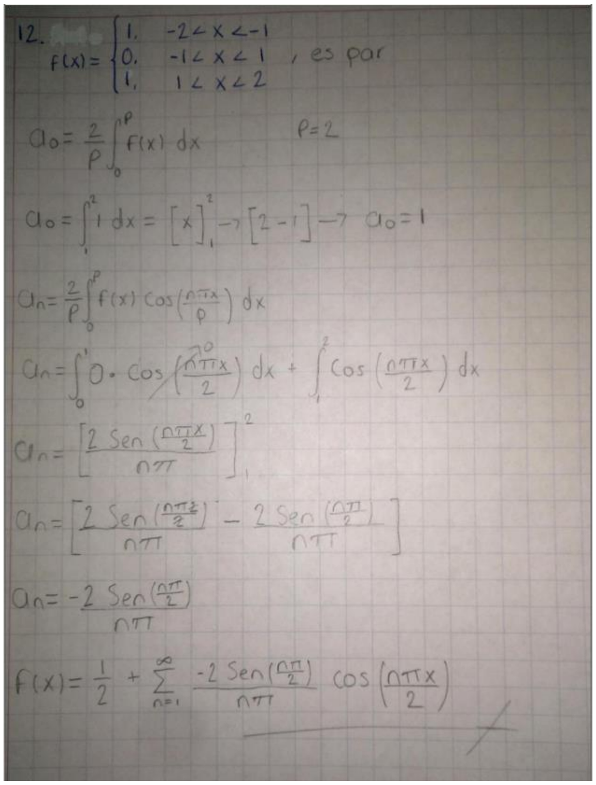
$$F(x) = \frac{1}{4} + \sum_{n=1}^{\infty} \frac{((-1)^n - 1)\cos(\pi n x)}{\pi^2 n^2} + -\frac{(-1)^n \sin(\pi n x)}{\pi n}$$

Se grafica la Serie de Fourier con 50 armónicos:

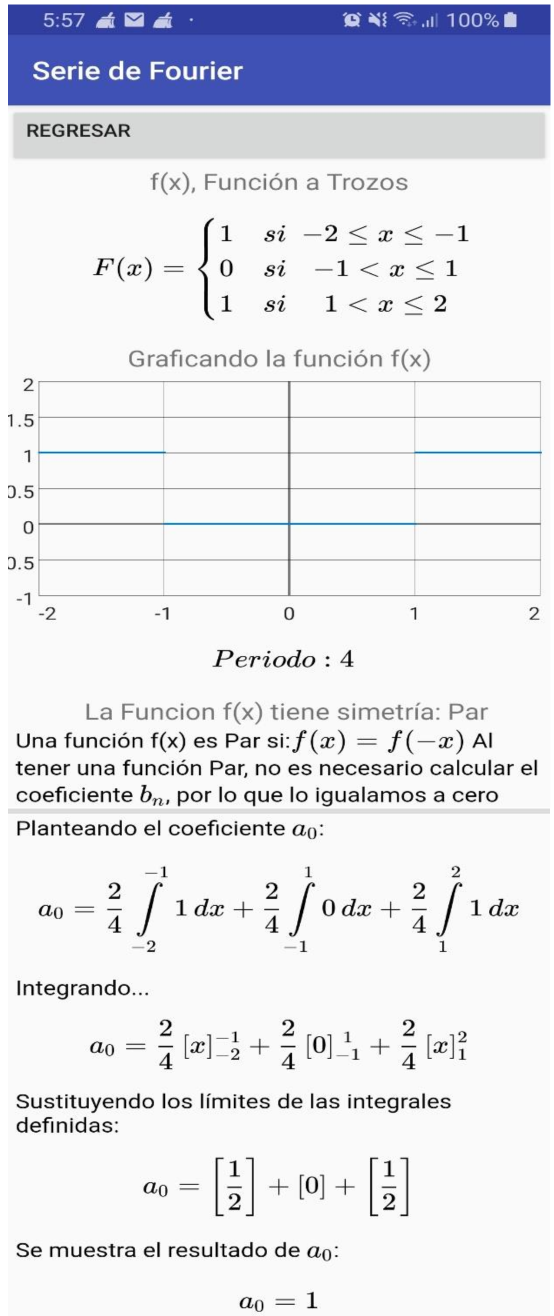
El siguiente caso es para una función de tres trozos, figura 5.9:

Figura 5.9. Comparación de cálculos (Fidelidad)

a) Resolución en hoja del alumno



b) Resolución en pantalla de la aplicación



Planteando el coeficiente a_n:

$$a_n = \frac{2}{4}\int_{2}^{-1}[1]\cos\left(\frac{2n\pi x}{p}\right)\,dx + \frac{2}{4}\int_{1}^{1}[0]\cos\left(\frac{2n\pi x}{p}\right)\,dx + \frac{2}{4}\int_{1}^{2}[1]\cos\left(\frac{2n\pi x}{p}\right)\,dx$$

Integrando...

$$a_n = \frac{2}{4}\left[\frac{2\sin\left(\frac{\pi n x}{2}\right)}{\pi n}\right]_{-2}^{-1} + \frac{2}{4}\,[0]_{-1}^{1} + \frac{2}{4}\left[\frac{2\sin\left(\frac{\pi n x}{2}\right)}{\pi n}\right]_{1}^{2}$$

Sustituyendo los límites de las integrales definidas:

$$a_n = \left[\frac{-\sin\left(\frac{\pi n}{2}\right) + \sin(\pi n)}{\pi n}\right] + [0] + \left[\frac{-\sin\left(\frac{\pi n}{2}\right) + \sin(\pi n)}{\pi n}\right]$$

sin(π) = 0, sin(2π) = 0, sin(3π) = 0, ..., por lo tanto, podemos asumir que: sin(nπ) = 0, haciendo la sustitución:

$$a_n = \left[-\frac{\sin\left(\frac{\pi n}{2}\right)}{\pi n}\right] + [0] + \left[-\frac{\sin\left(\frac{\pi n}{2}\right)}{\pi n}\right]$$

cos(π) = -1, cos(2π) = 1, cos(3π) = -1, ..., por lo tanto, podemos asumir que: cos(nπ) = $(-1)^n$, haciendo la sustitución:

$$a_n = \left[-\frac{\sin\left(\frac{\pi n}{2}\right)}{\pi n}\right] + [0] + \left[-\frac{\sin\left(\frac{\pi n}{2}\right)}{\pi n}\right]$$

Se muestra el resultado de a_n:

$$a_n = -\frac{2\sin\left(\frac{\pi n}{2}\right)}{\pi n}$$

$$b_n = 0$$

La Serie de Fourier:

$$F(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos\left(\frac{n\pi}{p}x\right) + b_n sin\left(\frac{n\pi}{p}x\right)$$

Se obtiene la Serie de Fourier:

$$F(x) = \frac{1}{2} + \sum_{n=1}^{\infty} -\frac{2\sin\left(\frac{\pi n}{2}\right)\cos\left(\frac{\pi n x}{2}\right)}{\pi n}$$

Se grafica la Serie de Fourier con 50 armónicos:

La página principal de la aplicación en la Consola de Google Play, presenta datos generales de la misma, como lo son las instalaciones activas a la fecha y el promedio de valoración de los usuarios, tabla 5.1. Se ve que, a la fecha mostrada, la aplicación sigue activa en 1650 dispositivos.

Tabla 5.1. Datos generales en la Consola de Google Play

Nombre de la aplicación	Instalaciones activas	Valoración en Google Play	Última actualización	Estado
Serie de Fourier	1650	4.25	12/1/2020	Publicada

La aplicación ha sido descargada por 5587 usuarios a la fecha, la tabla 5.2 muestra el número de instalaciones y de desinstalaciones por mes.

Tabla 5.2. Número total de descargas de la aplicación por mes

Mes/Año	Descargas Totales	Usuarios Activos	Usuarios Perdidos
Mayo 2019	149	149	58
Junio 2019	230	230	209
Julio 2019	260	256	178
Agosto 2019	287	254	219
Septiembre 2019	451	399	286
Octubre 2019	570	513	398
Noviembre 2019	874	788	556
Diciembre 2019	782	639	831
Enero 2020	398	339	443
Febrero 2020	585	491	476
Marzo 2020	998	841	744
Abril 2020	1.268	1.029	1.134
Mayo 2020	1.887	1.531	1.516
Total Descargas	5587.155	4901.56	4400.65

Al día 26 de mayo de 2020, la aplicación Serie de Fourier está instalada y activa en 300 dispositivos diferentes, para dar un total de 1625 instalaciones activas, tabla 5.3.

Tabla 5.3. Dispositivos que usan los usuarios que han calificado la aplicación:

dispositivo	Valoración
Asus ZenFone 5 (ZE620KL) (ASUS_X00QD)	5.00
Nokia Nokia 7 plus (B2N_sprout)	5.00
Cat Cat S41 (CatS41)	5.00
Huawei Honor 7X (HWBND-H)	5.00
Huawei Y6II (HWCAM-H)	5.00
Huawei HUAWEI Y6 2019 (HWMRD-M1)	5.00
Huawei Mate 20 lite (HWSNE)	5.00
Lanix Ilium Alpha 3 (Ilium_Alpha_3)	5.00
OnePlus OnePlus 7 (OnePlus7)	5.00
ZTE BLADE V9 (P450L10)	5.00
Condor PGN605 (PGN605)	5.00
Samsung Galaxy A20 (a20)	5.00
Samsung Galaxy A30 (a30)	5.00
Samsung Galaxy A5(2017) (a5y17lte)	5.00
Motorola moto g(6) (ali)	5.00
Motorola moto z3 play (beckham)	5.00
Samsung Galaxy S7 edge (hero2lte)	5.00
Samsung Galaxy J2 (j2corepltemtr)	5.00
Samsung Galaxy J4+ (j4primelte)	5.00
Samsung Galaxy J6+ (j6primelte)	5.00
Samsung Galaxy J7 Neo (j7velte)	5.00
Samsung Galaxy J7 (j7y17lte)	5.00
Redmi Redmi Note 7 (lavender)	5.00
LGE LG K10 LTE (m253)	5.00
LGE LG Q6 (mh)	5.00
Motorola Moto G (5S) (montana)	5.00
Samsung Galaxy J5 Prime (on5xelte)	5.00
Samsung Galaxy S9+ (star2lte)	5.00
Motorola moto g(7) play (channel)	4.50
SILVER_MAX	4.00
Samsung Galaxy E7 (e73g)	4.00
Samsung Galaxy J2 (j2y18lte)	4.00

Huawei P20 lite (HWANE)	3.50
Samsung Galaxy J7 (j7elte)	3.00
Huawei HUAWEI Y9 2019 (HWJKM-H)	1.00
Huawei HUAWEI Y7 Prime 2018 (HWLDN-Q)	1.00
Lenovo Lenovo TAB3 7 Essential (TB3-710F)	1.00
Samsung Galaxy Tab A (2016) (gtaxlwifi)	1.00
Samsung Galaxy J7(2016) (j7xelte)	1.00
Motorola Moto E (5) (nora_8917)	1.00
Valoración Media	4.25

La tabla 5.4, muestra la distribución de usuarios activos por país.

Tabla 5.4. Usuarios activos por país:

País	Todas las adquisiciones	País	Todas las adquisiciones	País	Todas las adquisiciones
México	310	Kirguistán	6	Líbano	2
Colombia	128	Polonia	6	Noruega	2
El Salvador	85	Estados Unidos	6	Singapur	2
Indonesia	74	Uzbekistán	6	Uganda	2
Rusia	72	Moldavia	5	República Democrática del Congo	1
Argentina	68	Uruguay	5	Albania	1
Ecuador	58	Kazajistán	5	Armenia	1
España	58	Malasia	5	Angola	1
Ucrania	57	Filipinas	5	Austria	1
Jordania	57	Pakistán	5	Bangladés	1
Irak	42	Rumania	5	Bulgaria	1
India	33	Taiwán	5	Barín	1
Perú	31	Omán	4	Suiza	1
Egipto	31	Arabia Saudí	4	Costa Rica	1
Chile	27	Venezuela	4	Chipre	1
Brasil	22	Nicaragua	3	Chequia	1
Francia	22	Reino Unido	3	Hungría	1
Italia	20	Irlanda	3	Israel	1
Emiratos Árabes Unidos	17	Japón	3	Camboya	1
Corea del Sur	17	Portugal	3	Kuwait	1
Bolivia	15	Tailandia	3	Libia	1
Guatemala	14	Túnez	3	Madagascar	1

Palestina	14	Turquía	3	Myanmar (Birmania)	1
Panamá	10	Región desconocida	3	Puerto Rico	1
Irán	9	Marruecos	2	Catar	1
Argelia	7	Bosnia y Herzegovina	2	Somalia	1
Bielorrusia	7	Bélgica	2	Trinidad y Tobago	1
Sudáfrica	7	Costa de Marfil	2	Yemen	1
Mozambique	6	Alemania	2	Zambia	1
República Dominicana	6	Gabón	2	Mauricio	0
Honduras	6	Grecia	2		
Todos los países 1,478					

Capítulo VI. Conclusiones

Los estudiantes que acceden hoy en día a la educación superior se consideran nativos digitales. Prefieren aprender de una forma más interactiva y en su mayoría siempre cuentan con un dispositivo móvil. Por lo que las Universidades tienen que voltear hacia esta tecnología y desarrollar aplicaciones móviles de los contenidos que se aborden en sus instituciones para, de esta manera, contribuir al logro de las competencias, tanto profesionales, disciplinares, genéricas y en este caso, matemáticas.

En la actualidad existen diversas aplicaciones desarrolladas para los contenidos matemáticos que ofrecen las distintas universidades a nivel internacional, pero cada institución debe ser capaz de evaluar dichas aplicaciones y determinar cuáles reúne los requisitos esenciales para sus cursos, es decir, dependiendo de su contexto y del objetivo a cumplir.

Con respecto a la aplicación Serie de Fourier, se observa que, en base a los resultados de la tienda de Google Play, en un año de estancia en la tienda, la aplicación tiene más de 8,000 descargas en más de 90 países, con comentarios muy alentadores hacia la misma, los cuales superan los 4.25 puntos en promedio.

Se logró que la aplicación se ejecute en el sistema operativo Android en sus diferentes versiones desde la 4.4 y en todos los teléfonos inteligentes (smartphones).

De acuerdo con la encuesta realizada a los alumnos en el salón de clases de la materia de Ecuaciones Diferenciales del Instituto Tecnológico de Chihuahua II, el uso de la aplicación es sencilla e intuitiva.

Se logró que la aplicación diera los resultados sin la necesidad de estar conectada a Internet. Esto se logró mediante la integración de la librería Chaquopy, la cual nos ayuda a ejecutar código Python en un sistema operativo Android, haciendo posible que todos los cálculos se hagan de manera local en el dispositivo. Esto hace una gran diferencia con respecto a otras aplicaciones matemáticas, las cuales mandan los datos capturados en el celular a un servidor en Internet, que se encarga de resolver el problema y de mandar los resultados al dispositivo móvil.

El hecho de que además de resolver el problema la aplicación nos permita observar dos gráficas, una de la función de entrada y la otra del desarrollo de la serie de Fourier, dan un valor agregado a la misma, puesto que el usuario tiene estas dos resultantes en un mismo lugar, ya que eso es único en esta aplicación.

La aplicación fue útil para desarrollar series de Fourier en el salón de clases, como refuerzo a la materia de Ecuaciones Diferenciales así como, también, para otros profesionistas que requieren resolver un problema de cualquier tipo, en donde se involucrara una serie de Fourier.

Cabe destacar que la realización de este proyecto, se tomó en cuenta la fidelidad del procedimiento, es decir, los pasos intermedios para obtener el desarrollo de la serie de Fourier, ya que esto es importante para los alumnos de la materia de Ecuaciones Diferenciales o alguna otra materia relacionada con este tema, para la apropiación conceptual de las series de Fourier. Sin embargo, si a un usuario solo le interesa el resultado de la serie de Fourier y su gráfica, solo basta con que se desplace hasta el final de la pantalla para ver esos resultados.

Para lograr la fidelidad, se trabajó con cálculo simbólico mediante la librería SymPy que se desarrolla en Python, la cual dio un muy buen resultado.

Esta aplicación logra el objetivo propuesto al inicio del programa de maestría, ya que se logró concluir la aplicación móvil que desarrolla y grafica la Serie de Fourier de una Función Matemática. La función puede ser continua o hasta de tres trozos. Puede detectar casos especiales, en donde alguno de los primeros tres términos de la serie se indetermine. Tanto la gráfica de la función de entrada, como la gráfica de la Serie de Fourier son exactas. Y lo más interesante, es que no necesita de una conexión a Internet para desarrollar la serie de Fourier.

Como conclusión general se logró la implementación de un desarrollo tecnológico en forma de aplicación móvil, el cual desarrolla series de Fourier basándose en la simetría de la función de entrada, la cual puede ser Par, Impar o ninguna y en un intervalo acotado en el dominio de la función. Pudiendo este desarrollo ser utilizado tanto en el ámbito escolar, como en el industrial.

Como puntos de mejora se pueden mencionar los siguientes:

En cuanto a funcionalidad, puede ser adaptar la función a media onda o a un cuarto de onda, determinar, de una función sin simetría, una serie de Fourier Par o Impar, también mostrar los resultados en forma polar. En cuanto a estética y usabilidad, mejorar las pantallas mostradas con botones más interactivos y desplazamientos entre pantallas más fluidas.

La implementación de este proyecto fue muy interesante, ya que el gran reto fue el poder hacer que Python se ejecutara en el sistema operativo Android, al final con la librería Chaquopy se logró el objetivo.

Como un proyecto siguiente, continuación de éste, sería la ejemplificación de la solución de una ecuación diferencial de un sistema de masa resorte por medio de serie de Fourier.

Capítulo VII. Referencias

Abud, M. A. (2005). MECSE: Conjunto de Métricas para Evaluar Software Educativo. *UPIICSA en línea. Tecnologías, Ciencia y Cultura*(39), 7-10. Recuperado el 30 de enero de 2019, de http://repositoriodigital.ipn.mx/handle/123456789/5329

Aparisi, L., & Pochulu, M. (2013). Dificultades que enfrentan los profesores en escenarios de modelización. *Acta Latinoamericana de Matemática Educativa, Vol. 26*, pp. 1387-1397.

Assum, D., Guil, D., & Malet, O. (2014). El uso de GeoGebra® en las aulas del Curso de Ingreso a la Universidad: los porqués de una elección. *Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación*(647).

Bourne, M. (2018). Obtenido de Matemáticas Interactivas: https://www.intmath.com/fourier-series/3-fourier-even-odd-functions.php#cosblip

Cabral, M. (2012). *FouSE-Fourier Series Expansion*. Obtenido de https://play.google.com/store/apps/details?id=examples.mcabral.pds&hl=es

Camacho, A., Caldera, M., & Valenzuela, V. (2019). Fidelidad en el uso de app para la resolución de ecuaciones diferenciales. *Apertura, 11*(1), pp. 74-89. doi:http://dx.doi.org/10.32870/Ap.v11n1.1463

Chaquopy Ltd. (2020). Obtenido de Chaquopy: https://chaquo.com/chaquopy/

Covacevich, C. (2014). Cómo seleccionar un instrumento para evaluar aprendizajes estudiantiles. *Banco Interamericano de Desarrollo*, pp. 1-40. Recuperado el 1 de septiembre de 2018, de https://publications.iadb.org/bitstream/handle/11319/ 6758/C%C3%B3mo-seleccionar-un-instrumento-para-evaluar-aprendizajes-estudiantiles.pdf

Desmos. (2020). *Desmos Calculadora Graficadora*. Obtenido de https://play.google.com/store/apps/details?id=com.desmos.calculator&hl=es

Fallas, J., & Chavarría, J. (2010). Validación de Software Educativo. *VII Festival Internacional de Matemática, Sede San Carlos, Instituto Tecnológico de Costa Rica*, pp. 1-8. Recuperado el 2 de Septiembre de 2018, de http://www.cientec.or.cr/matematica/2010/ponenciasVI-VII/Validacion-Fallas-Jeffrey.pdf

Ferrari, S. I., & Mariño, S. (2014). Guía de evaluación de la usabilidad para herramientas de minería de datos. *No Solo Usabilidad*(13). Recuperado el 31 de Agosto de 2018, de http://www.nosolousabilidad.com/articulos/usabilidad_mineria_datos.htm

Flores, I. P., Valadez, S., & Atencio, A. M. (2016). La Didáctica de la matemática en contexto, promotora de la motivación matemática en ecuaciones diferenciales. *Revista electrónica Humanidades, Tecnología y Ciencia. IPN*(14). Obtenido de https://docplayer.es/34845077-La-didactica-de-la-matematica-en-contexto-promotora-de-la-motivacion-matematica-en-ecuaciones-diferenciales.html

Floría, A. (2001). Manual de Técnicas para el Diseño Participativo de Interfaces de Usuario de Sistemas basados en Software y Hardware. 1-173. Zaragoza, España. Recuperado el 3 de Septiembre de 2018, de https://www.disenomovil.mobi/multimedia_un/01_intro_ux/Manual_de_Tecnicas_para_el_Diseno_Participativo-usabilidad_corregido.pdf

Google. (2020). *Google Play Store*. Obtenido de https://play.google.com/store?hl=es

International Geogebra Institute. (2018). *Geogebra Clásico*. Obtenido de https://play.google.com/store/apps/details?id=org.geogebra&hl=es

Kreyszing, E. (2011). *Advanced engineering mathematics.* Unites States of America: Jhon Wiley & Sons, INC.

Marqués, P. (2002). Evaluación y selección de software educativo. *Comunicación y pedagogía: Nuevas tecnologías y recursos didácticos*(185), 31-37. Obtenido de https://diversidad.murciaeduca.es/tecnoneet/docs/2002/62002.pdf

Mathstools. (2018). *Calculadora de integrales*. Obtenido de https://play.google.com/store/apps/details?id=com.mathstools.integrals&hl=es

Microsoft Corporation. (2020). *Excel: Ver, editar y crear hojas de cálculo*. Obtenido de https://play.google.com/store/apps/details?id=com.microsoft.office.excel&hl=es

Mosquera, M. A., & Vivas, S. J. (2017). Análisis comparativo de software matemático para la formación de competencias de aprendizaje en cálculo diferencial. *Plumilla Educativa, 19*(1), 98-113. doi:https://doi.org/10.30554/plumillaedu.19.2476.2017

Numpy. (2020). Obtenido de Numpy: https://numpy.org/

Petuhov, I. (2016). *Serie de Fourier*. Obtenido de https://play.google.com/store/apps/details?id=org.krapp2.fourier&hl=es

Plaza, L. F. (2016). Obstáculos presentes en modelación matemática. Caso ecuaciones diferenciales en la formación de ingenieros. *Revista Científica*(25), 176-187. doi:10.14483/udistrital.jour.RC.2016.25.a1

Python. (2020). Obtenido de Python: https://www.python.org/

Pythones. (2020). Obtenido de Pythones: https://pythones.net/

Rackauckas, C. (2018). A Comparison Between Differential Equation Solver Suites In MATLAB, R, Julia, Python, C, Mathematica, Maple, and Fortran. *The Winnower*. doi:10.15200/winn.153459.98975

Symbolab. (2020). *Symbolab*. Obtenido de https://play.google.com/store/apps/details?id=com.devsense.symbolab&hl=es

SymPy Development Team. (2020). Obtenido de Sympy: https://www.sympy.org/en/index.html

Zill, D. (2008). *Matemáticas avanzadas para ingeniería.* México: McGraw_Hill.

Zill, D., & Warren, W. (2015). *Ecuaciones Diferenciales con problemas con valores en la frontera* (octava ed.). México: Cengage Learning. Recuperado el 1 de Mayo de 2019