



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

TEMA:

Interfaz de resolución de las ecuaciones diferenciales (tipo parabólicas)

AUTOR:

Encalada Jiménez Katherine del Rocío

**Trabajo de Integración Curricular previo a la obtención del título de
INGENIERA EN CIENCIAS DE LA COMPUTACIÓN**

TUTOR:

Ing. Céleri Mujica, Colón Mario Mgs.

Guayaquil, Ecuador

9 de marzo de 2022



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación, fue realizado en su totalidad por **Encalada Jiménez Katherine del Rocío**, como requerimiento para la obtención del título de **Ingeniera en Ciencias de la Computación**.

TUTOR

f.

Ing. Celleri Mujica, Colón Mario

Guayaquil, a los 9 días del mes de marzo de 2022.



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Encalada Jiménez, Katherine del Rocío**

DECLARO QUE:

El Trabajo de Titulación, **Interfaz de resolución de las ecuaciones diferenciales (tipo parabólicas)**, previo a la obtención del título de **Ingeniera en Ciencias de la Computación**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 9 días del mes de marzo de 2022.

EL AUTOR

f. _____
Encalada Jiménez, Katherine del Rocío



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

AUTORIZACIÓN

Yo, **Encalada Jiménez, Katherine del Rocío**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Interfaz de resolución de las ecuaciones diferenciales (tipo parabólicas)**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 9 días del mes de marzo de 2022.

EL AUTOR:

f. _____
Encalada Jiménez, Katherine del Rocío



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

REPORTE URKUND

URKUND

Documento: Documento titulación Katherine Encalada Jimenez final.docx (0128620830)

Presentado: 2022-02-22 19:02 (-05:00)

Presentado por: katherine.encalada@cu.ucsg.edu.ec

Recibido: colon.celleri.ucsg@analysis.orkund.com

Mensaje: Tesis [Mostrar el mensaje completo](#)

2% de estas 17 páginas, se componen de texto presente en 6 fuentes.

Lista de fuentes - Bloques

Categoría	Enlace/nombre de archivo
	NERV 02 A.docx
	https://www.ugr.es/~prodejaaffo/ETS/CCP/Ita
	http://www1.maths.tcd.ie/~berger/Text
	metodos-de-modelamiento.pdf

FACULTAD DE INGENIERÍA CARRERA DE INGENIERÍA EN CIENCIAS DE LA
COMPUTACIÓN

TEMA: Interfaz de resolución de las ecuaciones diferenciales (tipo parabólicas)

AUTOR: Encalada Jiménez Katherine del Rocío

Trabajo de Integración Curricular previo a la obtención del título de INGENIERO EN
CIENCIAS DE LA COMPUTACIÓN

TUTOR: Ing. Céleri Mujica, Mario Colón, Mgs.

TUTOR

f.

Ing. Céleri Mujica, Colón Mario

AGRADECIMIENTO

Es un honor poder decir que este trabajo es el resultado de la bondad de Dios, él ha sido mi guía y mi fortaleza. Gracias Señor por tu infinita misericordia.

Le agradezco a Alberto Encalada, mi papá, quien con esfuerzo logró sacarnos adelante. Gracias por tu amor, por todas las risas en medio de tantas lágrimas y por las largas platicas llenas de esperanza. A mi mamá, Rocio Jiménez, mi ejemplo de perseverancia y resiliencia, gracias por los consejos, el inmensurable amor y los abrazos reconfortantes. Los amo. A mi hermana, quien es mi ejemplo a seguir. Gracias por confiar en mí, gracias por hacerme entender que no solo se trata del final, sino de gozarse el proceso. Te amo, Florecita. A Jervis, gracias por los consejos que me han permitido dejar entrar un poco de realismo en mi eterno optimismo.

Le agradezco a todos los docentes que se han visto involucrados en mi carrera universitaria. Gracias a mi estimado tutor de tesis, Ing. Mario Celleri, muchas gracias por toda la paciencia, la enseñanza y la guía.

Quiero agradecerles a mis amigos. A los que comenzaron conmigo, los que se quedaron en el camino, los que se integraron casi al final y a los que están ahora. Carlos Luis, Emilito, Santi, Bryan y Jean Carlos gracias por su valiosa amistad. En especial, gracias a mis grandes amigas Rebeca Barahona y Paula Burbano, gracias por las lágrimas, las risas, los consejos y el apoyo moral, atesoraré cada momento de desesperación, de complicidad y de felicidad. Los llevo a todos en un lugar muy especial en mi corazón.

DEDICATORIA

A Dios, quien ha forjado mi camino y lo ha inundado de su amor, misericordia y bondad, quien ha sido mi fuente de fortaleza más grande.

A mis padres, quienes han sido mi guía y mi ejemplo para poder llegar a este punto de mi carrera, quienes con su esfuerzo y dedicación hicieron hasta lo imposible para que yo logre conseguir mi título universitario.

Los amo.

-Katherine Encalada Jiménez.

ÍNDICE

AGRADECIMIENTO	VI
DEDICATORIA	VII
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS	XIII
RESUMEN.....	XIV
ABSTRACT.....	XV
INTRODUCCIÓN	2
CAPÍTULO 1. EL PROBLEMA.....	4
Planteamiento del Problema	4
Delimitación del problema	5
Evaluación del Problema.....	5
Objetivos	6
General.....	6
Específicos	6
Alcance	6
Justificación.....	7
Pregunta de investigación	7
CAPÍTULO II. MARCO TEÓRICO Y CONCEPTUAL.....	8
Marco Teórico	8
Historia de las ecuaciones diferenciales.....	8
Ecuaciones Diferenciales	9
Clasificación de Ecuaciones Diferenciales	10

Clasificación según su tipo	10
Ecuaciones Diferenciales Ordinarias	10
Ecuaciones Diferenciales Parciales	11
Clasificación según su Orden	11
Clasificación según su grado	12
Solución de una Ecuación Diferencial	12
Tipos de soluciones	13
Solución General	13
Solución Particular	13
Solución Singular	13
Ecuaciones Diferenciales Parciales	13
Ecuaciones Diferenciales Parciales Parabólicas	15
Método de las Diferencias Finitas	17
Primera Derivada por Diferencia	18
Fórmulas Centradas	18
Fórmulas hacia atrás y hacia adelante	19
Segunda Derivada en Diferencias Finitas	20
Ecuación de calor en una dimensión	21
Método Explícito	22
Convergencia y estabilidad	23
Método Implícito	23
Marco Conceptual	24
Python	24
Librería Tkinter	25

SymPy	26
Librería Matplotlib	26
Librería NumPy.....	27
Visual Studio Code	27
CAPÍTULO III. METODOLOGÍA DE LA INVESTIGACIÓN	29
Metodología de la investigación	29
Metodología de desarrollo.....	33
CAPÍTULO IV. DESARROLLO DEL PROYECTO	35
Introducción.....	35
Objetivo	35
Responsable	35
Descripción de la interfaz	35
Arquitectura de la Interfaz	36
Herramientas Tecnológicas	37
Características de la interfaz.....	40
Página de inicio	40
Pantalla de resolución	41
Método Explícito	44
Método Implícito	46
Pantalla de gráfica.....	48
Pantalla Matriz de Resultados	49
Pantalla Tabla de Resultados.....	50
CONCLUSIONES	51
RECOMENDACIONES	52

REFERENCIAS 53

ANEXOS 56

 Manual de Usuario 56

 Código de la solución tecnológica propuesta. 57

ÍNDICE DE FIGURAS

FIGURA 1: CLASIFICACIÓN DE LAS ECUACIONES DIFERENCIALES. ELABORADO POR EL AUTOR.....	10
FIGURA 2: EJEMPLO COMPARATIVO DE ECUACIONES DIFERENCIALES PARABÓLICAS.....	16
FIGURA 3: ECUACIÓN DEL CALOR EN UNA DIMENSIÓN	21
FIGURA 4: ARQUITECTURA DE LA SOLUCIÓN. ELABORADO POR EL AUTOR.	36
FIGURA 5: PANTALLA INICIAL	40
FIGURA 6: PANTALLA DE RESOLUCIÓN	41
FIGURA 7: MENÚ ARCHIVO	42
FIGURA 8: MENÚ INFORMACIÓN.....	43
FIGURA 9: PANTALLA ACERCA DE LAS VARIABLES	43
FIGURA 10: PANTALLA DE ADVERTENCIA.	44
FIGURA 11: PANTALLA MÉTODO EXPLÍCITO	44
FIGURA 12: MÉTODO EXPLÍCITO DESARROLLADO EN PYTHON	45
FIGURA 13: PANTALLA DE INCUMPLIMIENTO DE LA CONDICIÓN DE ESTABILIDAD	46
FIGURA 14: VENTANA DE REINGRESO DE DATOS.	46
FIGURA 15: MÉTODO IMPLÍCITO.....	46
FIGURA 16: MÉTODO IMPLÍCITO DESARROLLADO EN PYTHON.....	47
FIGURA 17: PANTALLA DE GRÁFICA	48
FIGURA 18: PANTALLA TABLA DE RESULTADOS.....	49
FIGURA 19: PANTALLA DE PUNTOS DE LA ÚLTIMA ITERACIÓN	50

ÍNDICE DE TABLAS

TABLA 1: DEFINICIÓN DE OPERADORES DE DIFERENCIAS FINITAS	18
TABLA 2: COMPARACIÓN ENTRE MATLAB Y PYTHON.....	38

RESUMEN

En la actualidad existen una gran cantidad de alternativas de interfaces que permiten resolver ecuaciones diferenciales; sin embargo, no existe alguna que cumpla con los requisitos propuestos o que resuelva el tipo de ecuaciones presentados en este proyecto. El objetivo de este trabajo de titulación fue desarrollar una interfaz que resuelva y grafique ecuaciones diferenciales parciales de tipo parabólicas, los requisitos principales de esta interfaz son que sea gratis, intuitiva y que no necesite de una conexión a Internet para poder presentar los resultados. La interfaz fue desarrollada para el uso de los estudiantes de la carrera de Computación en la Universidad Católica de Santiago de Guayaquil. Para el desarrollo de la interfaz se implementó el lenguaje de programación Python junto a sus librerías Tkinter, SymPy y Matplotlib para el front end, la resolución de los cálculos numéricos y la representación de gráficos, respectivamente.

Key Words: Ecuaciones Diferenciales Parciales, EDP Parabólicas, Python, Interfaz, Calculadora de EDP Parabólicas.

ABSTRACT

Currently there are a large number of alternative interfaces that allow solving differential equations; however, there is no one that meets the proposed requirements or that solves the type of equations presented in this project. The objective of this degree work was to develop an interface that solves and graphs partial differential equations of parabolic type, the main requirements of this interface are that it is free, intuitive and that it does not need an Internet connection to be able to present the results. The interface was developed for the use of students of the Computer Science program at the Catholic University of Santiago de Guayaquil. For the development of the interface, the Python programming language was implemented along with its libraries Tkinter, SymPy and Matplotlib for the front end, the resolution of numerical calculations and the representation of graphics, respectively.

Key Words: Partial Differential Equations, Parabolic PDE, Python, Interface, Parabolic PDE Calculator.

INTRODUCCIÓN

Dentro de la carrera universitaria de un estudiante de ingeniería, un tema a destacar es el estudio de las ecuaciones diferenciales como base para la construcción de conocimientos permanentes.

Este trabajo tiene como objetivo principal la creación de una interfaz que sea capaz de resolver ecuaciones diferenciales tipo parabólicas y a su vez, debido a su desarrollo interactivo, lograr facilitar el proceso de aprendizaje de los estudiantes que harán uso de la misma.

Una interfaz se caracteriza por la agilidad y la facilidad con la que un usuario puede interactuar con la máquina. Su objetivo es lograr crear una comunicación directa y de esta manera, el usuario, por medio de conjuntos de imágenes y objetos gráficos, pueda acceder a la información y acciones presentadas a través de la interfaz.

La estructura del presente trabajo de integración curricular será planteada de la siguiente manera:

En el Capítulo I se podrá encontrar la descripción del problema que se busca solucionar con la creación de la interfaz. Así mismo, se detallará el objeto principal, los objetivos específicos, la justificación y el alcance.

En el Capítulo II se podrá encontrar el marco teórico donde se explicará explícitamente la teoría de las ecuaciones diferenciales, su clasificación y su alcance. Además, se encontrará todo en cuanto a la teoría de las ecuaciones diferenciales parciales parabólicas y su método de solución, el método de las

diferencias finitas. Del mismo modo, estará los conceptos básicos del lenguaje de programación a usar y sus librerías.

En el Capítulo III se abordará todo en cuanto a las metodologías utilizadas en el presente trabajo. Se profundizará en la metodología de investigación y metodología de desarrollo utilizadas.

En el Capítulo IV se detallará el proceso de desarrollo de la solución tecnológica propuesta, además, se podrán encontrar capturas donde se explica el funcionamiento detallado de los distintos bloques de la interfaz.

Finalmente, se encontrarán las conclusiones y recomendaciones a las que se han llegado una vez finalizado el desarrollo del presente trabajo de titulación.

CAPÍTULO 1. EL PROBLEMA

La materia “Ecuaciones Diferenciales” es impartida a los estudiantes de la carrera Computación de la Universidad Católica de Santiago de Guayaquil, en su tercer ciclo. Dentro de la malla curricular de la carrera Ingeniería en Ciencias de la Computación, la carrera está categorizada como “Fundamentos Teóricos”, convirtiéndose así en una disciplina base para los estudiantes de dicha rama de la ingeniería.

Siendo un requerimiento para la universidad, se procederá con la creación de una interfaz capaz de resolver y graficar ecuaciones diferenciales tipo parabólicas.

Planteamiento del Problema

Las ecuaciones diferenciales es un tema que está presente en los niveles básicos de la malla curricular de un estudiante de ingeniería. Con este proyecto se buscará que los estudiantes puedan comprobar si sus resoluciones individuales han llevado un proceso y resultado correcto.

En la actualidad, la carrera de Computación no cuenta con un software que permita que sus estudiantes refuercen el tema “ecuaciones diferenciales parciales parabólicas” impartido en el aula de clases. Por ende, esto puede llegar a generar confusión e incertidumbre en los estudiantes al momento de desarrollar ejercicios acerca del tema antes mencionado.

Por esta razón, se requiere desarrollar e implementar una interfaz, la cual sea tomada como un apoyo en las actividades de aprendizaje de los

estudiantes. En cuanto al funcionamiento, la lógica se basará en que el usuario ingrese los datos que la interfaz solicite y obtendrá como resultados una respuesta analítica y una gráfica.

Delimitación del problema

Los estudiantes de la carrera de Computación tienen como fortaleza la habilidad de entender tanto la lógica de programación como del entendimiento para desarrollar porciones de código que gradualmente se convierten en programas funcionales. Debido a esta razón, con el desarrollo e implementación de esta interfaz se buscará satisfacer la falta de material de estudio, es decir, programas que resuelvan problemas de las propias asignaturas reflejadas en la malla curricular, que podrían ser provistos por los propios estudiantes convirtiendo sus softwares resultados como material de alto impacto que beneficiaría a la comunidad académica.

Evaluación del Problema

Con el fin de proveer un marco de evaluación, se enfatizará en los siguientes tres aspectos orientados al presente trabajo de titulación.

Delimitado: Con el fin de formar estudiantes a los que se les ha brindado cuantas herramientas de aprendizaje sean posibles, nace la necesidad de satisfacer la falta de material académico que le permita a los alumnos reforzar lo aprendido en los salones de clases.

Factible: Orientado para una asignatura específicamente, se busca el desarrollo de una interfaz que sea amigable con el usuario y que le permita

fortificar sus conocimientos en cuanto a Ecuaciones Diferenciales Parciales Parabólicas.

Identifica los productos esperados: El uso de la interfaz por los estudiantes de la carrera de Computación.

Objetivos

General

- Diseñar e implementar una Interfaz de resolución de las ecuaciones diferenciales parciales (tipo parabólicas), para el uso de los estudiantes de la carrera de Computación en la Universidad Católica de Santiago de Guayaquil.

Específicos

- Fundamentar teóricamente sobre las ecuaciones diferenciales parciales parabólicas.
- Analizar cuáles son las herramientas de software más competentes para el desarrollo de la interfaz.
- Diseñar y desarrollar una interfaz que permita la resolución y graficación de ecuaciones diferenciales parciales de tipo parabólicas.
- Validar el desarrollo de la interfaz y de los algoritmos de resolución de ecuaciones usados.

Alcance

Amenizar el proceso de estudio de un estudiante, consiguiendo que este, por medio de la interfaz, logre complementar su entendimiento sobre las EDP.

Con el fin de conseguir lo antes mencionado, la interfaz constará con las siguientes características.

- Resolver las ecuaciones diferenciales de tipo parabólica y presentar su graficación de los resultados.
- Lograr desarrollar una interfaz intuitiva y de fácil uso para el desarrollo de las ecuaciones diferenciales tipo parabólicas.

Justificación

El desarrollo e implementación de la interfaz tendrá como finalidad potenciar el aprendizaje de los estudiantes de la carrera de Computación. Es importante aclarar que dicha interfaz solo servirá como reforzamiento positivo para los alumnos de la institución universitaria. A través de esta interface podrán corroborar tanto su desarrollo como graficas de los problemas propuestos en clases y de manera directa podrán analizar y comprobar sus resultados con los de la interfaz.

Pregunta de investigación

¿Qué beneficios obtendrán los estudiantes de la carrera de Computación de la Universidad Católica Santiago de Guayaquil con esta interfaz que le permitirá resolver y graficar ecuaciones diferenciales parciales de tipo parabólicas?

CAPÍTULO II. MARCO TEÓRICO Y CONCEPTUAL

Marco Teórico

Historia de las ecuaciones diferenciales

Un tema que se toca repetitivamente dentro de un salón de clases es ¿Cuándo haré uso de las ecuaciones diferenciales en la vida real? Ciertamente, no es algo que podamos diferenciar a simple vista, sin embargo, el uso de estas se emplea a diario. Desde modelos matemáticos, hasta modelos que alerten la propagación de alguna bacteria, básicamente las ecuaciones diferenciales se ven expuestas en más de un área en nuestra vida cotidiana.

Las ecuaciones diferenciales son empleadas como base para el estudio matemático de problemas que se presentan en diversas ramas de las ciencias exactas. A lo largo de su historia se pueden identificar 5 etapas fundamentales, marcando un proceso definido para las matemáticas actuales. La primera etapa de las ecuaciones diferenciales nace con Leibniz en 1660-1700, donde nacen una cantidad innumerable de problemas de análisis debido al arduo estudio de la mecánica, la astronomía dinámica y la física matemática, pero no es sino hasta la tercera década del siglo XIX que Cauchy proporcionara los primeros problemas de existencia, dándole inicio a la segunda fase. La tercera empieza en 1840-1890 con Marius Sophus Lie y la aplicación de su teoría de los grupos contiguos a las aplicaciones diferenciales. La etapa siguiente inicia en 1850 con el teorema de existencia de Charles Émile Picard, su objetivo era lograr que las ecuaciones diferenciales lineales tengan un equivalente a la teoría de Galois de

las ecuaciones algebraicas. Finalmente, en la última etapa se destaca el estudio de ecuaciones diferenciales de dimensión infinita.

Ecuaciones Diferenciales

Una ecuación diferencial (ED) es una ecuación para una función que relaciona los valores de las funciones con los valores de sus derivadas. (Chasnov, 2021) Dentro de la estructura de una ecuación diferencial se pueden encontrar: constantes, variables, funciones de una o más variables (una o más funciones de la misma) y las derivadas ordinarias o parciales de las funciones mencionadas. Al momento de resolver una ecuación diferencial, se busca encontrar una o más funciones que satisfagan la ecuación. Las ecuaciones diferenciales son indispensables para la descripción matemática de la naturaleza. Son el núcleo y la esencia de un gran número de teorías físicas. Entre estas se encuentran: las ecuaciones de Newton y Lagrange para la mecánica clásica, las ecuaciones de Maxwell para el electromagnetismo clásico, la ecuación de Schrödinger para la mecánica cuántica y la ecuación de Einstein para la teoría general de la gravitación. (Nagy, 2021)

Junto a su desarrollo, una ED busca encontrar una función $y(x)$ que corrobore la relación funcional de la forma

$$F(x, y, y', y'', \dots, y^n) = 0$$

Clasificación de Ecuaciones Diferenciales

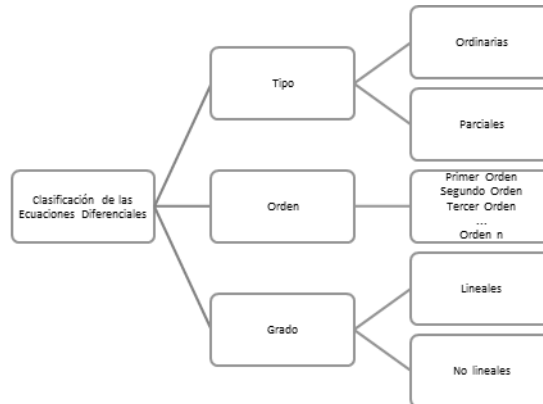


Figura 1: Clasificación de las Ecuaciones Diferenciales. Elaborado por el autor.

Clasificación según su tipo

Ecuaciones Diferenciales Ordinarias

Dentro de la clasificación de las ecuaciones diferenciales, se encuentran las ecuaciones diferenciales ordinarias. “Si una ecuación contiene sólo derivadas de una o más variables dependientes respecto a una sola variable independiente se dice que es una ecuación diferencial ordinaria (EDO).” (Zill, 2009)

Ejemplos:

$$1) y' = \cos x$$

$$2) y'^2 y - \sin x = 0$$

$$3) y'' + 9y = e^{-2x}$$

$$4) y'' = y' \sin x - \cos x$$

$$5) y' y''' - \frac{3}{2} y'^2 = 0$$

$$6) y''' + 2y'^2 + y' = e^x$$

$$7) \frac{dy}{dx} = x^2 + 4$$

$$8) \frac{dy}{dx} + 5y = e^x$$

$$9) \frac{d^2y}{dx^2} + 4\frac{dy}{dx} - y = 0$$

$$10) \frac{d^2y}{dx^2} - \frac{dy}{dx} + 6y = 0$$

Ecuaciones Diferenciales Parciales

El segundo tipo de ecuaciones diferenciales se define como “Una ecuación que involucra derivadas parciales de una o más variables dependientes de dos o más variables independientes se llama ecuación diferencial parcial (EDP).” (Zill, 2009)

Ejemplos:

$$1) \frac{\partial u}{\partial x} + \frac{\partial^2}{\partial y} + \frac{\partial u}{\partial z} = 0$$

$$2) \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

$$3) \frac{\partial^2 u}{\partial x^2} - a^2 \frac{\partial^2 u}{\partial y^2} = x$$

$$4) \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2}{\partial t^2} - 2 \frac{\partial u}{\partial t}$$

$$5) \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$$

Clasificación según su Orden

El orden de una ecuación diferencial es definido como el orden de derivación de mayor grado que aparece entre las derivadas contenida en ella.

Ejemplos:

- 1) $(x + y)dx = (y - x)dy$, es una ecuación diferencial de primer orden.
- 2) $\frac{dy}{dx} = 2e^{-x}$, es una ecuación diferencial de primer orden.
- 3) $y'' + \sin y = 0$, es una ecuación diferencial de segundo orden.
- 4) $x^2 \frac{d^2y}{dx^2} + x \frac{dy}{dx} + (x^2 - v^2)y = 0$, es una ecuación diferencial de segundo orden.
- 5) $y''' + y'' + y' + y = \text{sen}3x$, es una ecuación diferencial de tercer orden.
- 6) $y''' + (y')^{4'} + 4y = x$, es una ecuación diferencial de tercer orden.

Clasificación según su grado

Se conoce al grado de una ecuación diferencial como el mayor exponente que acompaña a la mayor derivada.

Ejemplos:

- 1) $(y')^3 + 5xy^2 = 4x$, es una ecuación diferencial de tercer grado.
- 2) $\frac{dy}{dx} + P(x)y = Q(x)$, es una ecuación diferencial de primer grado.
- 3) $\left(\frac{d^2y}{dx^2}\right)^5 + 5\frac{dy}{dx} - 3xy = x^5$, es una ecuación diferencial de orden 5.

Solución de una Ecuación Diferencial

Se define a la solución de una ecuación diferencial como la función que, al reemplazar la función incógnita, con sus respectivas derivaciones, logra verificar la ecuación. En otras palabras, una función $f(x)$ es una solución de la

ecuación diferencial si al sustituir la función $f(x)$ por y , se crea una identidad para x .

Tipos de soluciones

Solución General

La solución general de ecuaciones diferenciales es aquella ecuación de orden n que tiene n constantes diferentes.

Solución Particular

La solución particular es un apartado de la solución general. Usualmente utilizado para ecuaciones con una o más constantes que tengan un valor específico.

Solución Singular

La solución singular es aquella que cumple la identidad de la función, pero no se obtiene derivando la función general.

Ecuaciones Diferenciales Parciales

Dentro de la ingeniería, las ecuaciones diferenciales en derivadas parciales o ecuaciones diferenciales parciales, son utilizadas como herramientas de estudio dentro de campos como aerodinámica, transferencia de calor, meteorología, mecánica cuántica, electrostática, entre otros. Surgen de hacer modelos matemáticos a partir de fenómenos físicos.

Se denomina una ecuación diferencial parcial como aquellas ecuaciones cuya incógnita es una función con dos o más variables independientes, donde

el orden de la ecuación es representado por el orden de la derivada parcial más alta.

Se representa de la siguiente manera:

$$F\left(x, y, \dots, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \dots, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \dots\right) = 0,$$

donde $u = u(x, y, \dots)$ es la variable dependiente.

Una ecuación diferencial parcial lineal es aquella ecuación donde tanto su función como sus derivadas son lineales, y sus coeficientes dependen de las variables independientes de la función.

Ejemplos:

- 1) $\frac{\partial^2 u}{(\partial x^2)(\partial y)} + \frac{\partial^2 u}{\partial x^2} + 8 * u = 5 * y$, es una ecuación diferencial parcial lineal.
- 2) $\frac{\partial^2 u}{\partial x^2} + x * u * \frac{\partial u}{\partial y} = x$, es una ecuación diferencial parcial no lineal.

Una gran cantidad de problemas de ingeniería nacen a raíz del estudio de las ecuaciones diferenciales parciales resaltando a las de segundo orden. La forma general de las ecuaciones diferenciales de segundo orden en derivadas parciales se expresa de la siguiente manera:

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x * \partial y} + C \frac{\partial^2 u}{\partial y^2} + D = 0$$

Asumiendo que la ecuación es lineal, podemos establecer que A, B y C son funciones de x, e y D es una función de x, y, u, $\frac{\partial u}{\partial x}$ y $\frac{\partial u}{\partial y}$. A partir de

los coeficientes de los términos de la segunda derivada A, B y C, nace la ecuación diferencial parabólica, donde:

si, $B^2 - 4AC = 0$, es una ecuación parabólica.

Ecuaciones Diferenciales Parciales Parabólicas

Las ecuaciones diferenciales parciales parabólicas, son empleadas en el estudio de problemas que se ven durante los procesos como conductibilidad térmica, difusión o aplicaciones financieras. Mayormente son empleadas para la solución de problemas de propagación, indicando así que la solución de este tipo de ecuaciones es de un dominio espacio-tiempo abierto, y que consta con condiciones iniciales y de frontera. Para obtener la EDP, se considera la temperatura que actúa sobre la barra. La solución se construirá partiendo de las condiciones iniciales y se irá desarrollando rigiéndose a las condiciones de contorno. (Kreyszig, 2011)

La EDP parabólica es expresada de la siguiente forma:

$$k \frac{\partial^2 \phi}{\partial x^2} = \frac{\partial \phi}{\partial t}$$

Los problemas que predominan en esta rama de las ecuaciones diferenciales son aquellos relacionados con conducción de calor, problemas de difusión y aquellos que cambian con el tiempo.

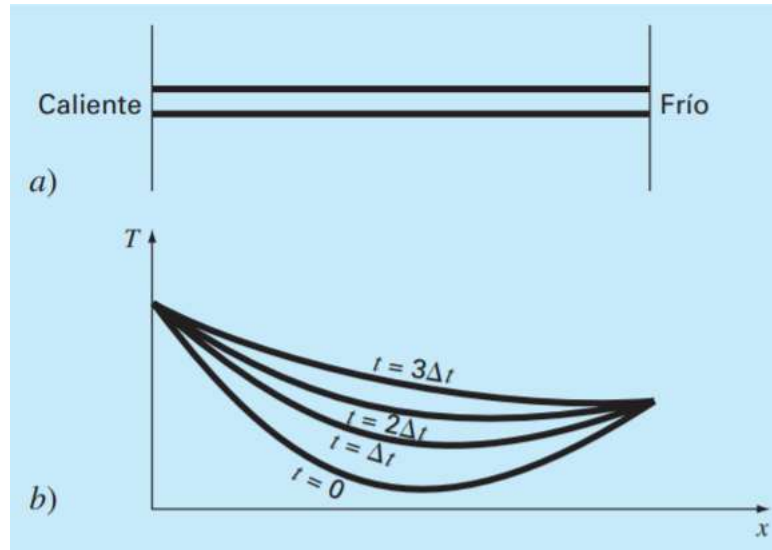


Figura 2: Ejemplo comparativo de ecuaciones diferenciales parabólicas.

En la Figura 2 se puede comparar dos casos que involucran las ecuaciones parabólicas: en el caso a) Una barra larga, totalmente aislada, excepto en sus extremos. “La dinámica de la repartición de una dimensión de la temperatura a través de toda la barra puede considerarse una ecuación diferencial parcial parabólica.” (Chapra & Canale, 2007, pág. 861) b) La solución, distribuciones del estado de la barra en distintos tiempos. (Chapra & Canale, 2007, pág. 862)

Ejemplo:

Ecuación de la conducción del calor, con coeficiente térmico c constante (Díaz, 2015):

$$U_t(x, t) = cU_{xx}(x, t) + Q(x, t),$$

para: $0 \leq x \leq a$ y $0 \leq t < \infty$, en donde $Q(x, t)$ representa las fuentes de calor existentes, sujeta a la condición inicial:

$$U(x, 0) = f(x), \text{ para } 0 < x < a,$$

y a dos condiciones de entorno:

- $U(0, t) = g_1(t)$, para $t > 0$
- $U(a, t) = g_2(t)$, para $t > 0$

Método de las Diferencias Finitas

El método de las Diferencias Finitas (DF) es usado no tan solo en modelos con coeficientes constantes, su uso abarca los modelos con coeficientes variables en el tiempo, modelos de tasa de interés simples o multifactoriales. (Sandoval-Ruiz, 2019)

El método de Diferencias Finitas permite discretizar ecuaciones diferenciales en derivadas parciales definidas en un dominio finito, el principio del método comprende la definición de las derivadas parciales de la función $f(x)$ (Sandoval-Ruiz, 2019). Se basa en el remplazo de las derivadas por aproximaciones de las mismas.

Operador	Símbolo	Representación
Diferencia hacia adelante	Δ	$\Delta_x u_{i,j} = u_{i+1,j} - u_{i,j}$
Diferencia hacia atrás	∇	$\nabla_x u_{i,j} = u_{i,j} - u_{i-1,j}$
Diferencia central	δ	$\delta_x u_{i,j} = u_{i+1/2,j} - u_{i-1/2,j}$
Desplazamiento	E	$E_x u_{i,j} = u_{i+1,j}$
Promedio	μ	$\mu_x u_{i,j} = \frac{u_{i+1/2,j} + u_{i-1/2,j}}{2}$
Diferenciación	D	$D_x u_{i,j} = \frac{du}{dx_{i,j}}$

Tabla 1: Definición de Operadores de Diferencias Finitas

Primera Derivada por Diferencia

Existen tres formas puntuales de obtener la derivada primera, llevando a obtener como resultado tres fórmulas distintas llamadas aproximaciones: centrada, hacia adelante, hacia atrás. (Universidad Nacional de Misiones, s.f.)

Fórmulas Centradas

El desarrollo de $T(x)$ en x_{i-1} alrededor de $T(x)$ en x_i y el de $T(x)$ en x_{i+1} alrededor de $T(x)$ en x_i son respectivamente. (Universidad Nacional de Misiones, s.f.)

$$a) T(x_{i-1}) = T(x_i) - T'(x_i)\Delta x + \frac{T''(x_i)}{2}\Delta x^2$$

$$b) T(x_{i+1}) = T(x_i) + T'(x_i)\Delta x + \frac{T''(x_i)}{2}\Delta x^2$$

Restando a de b, se obtiene:

$$T(x_{i+1}) = T(x_{i-1}) + 2T'(x_i)\Delta x + \frac{T''(x_i)}{2}\Delta x^2 + \frac{T'''(x_i)}{6}\Delta x^3$$

Donde, se obtiene:

$$T'(x_i) = \frac{T(x_{i+1}) - T(x_{i-1}))}{2\Delta x} - \frac{T'''(x_i)\Delta x^2}{6}$$

↓

$$O(\Delta x^2)$$

$$T'(x_i) = \frac{T(x_{i+1}) - T(x_{i-1}))}{2\Delta x} - O(\Delta x^2) + \dots$$

$$\overline{\Delta x^2} = (\Delta x)^2$$

que representa la derivada primera centrada de T en la dirección x. En diferencias finitas la primera derivada se aproxima por el primer sumando desechando los términos de orden superior. En este caso $O(\overline{\Delta x^2})$ es de segundo orden y la derivada primera se calcula con esta aproximación.

Fórmulas hacia atrás y hacia adelante

De la ecuación:

$$T(x_{i-1}) = T(x_i) - T'(x_i)\Delta x + \frac{T''(x_i)}{2}\Delta x^2, \text{ se obtiene:}$$

$$T'(x_i) \cong \frac{T(x_i) - T(x_{i-1}))}{\Delta x} + O(\Delta x)$$

$$\text{donde } O(\Delta x) = \frac{T''(x_i)}{2}\Delta x$$

Convirtiéndose esa ecuación en la expresión de la derivada primera en diferencia hacia atrás.

Por otro lado, de la ecuación:

$$T(x_{i+1}) = T(x_i) + T'(x_i)\Delta x + \frac{T''(x_i)}{2}\overline{\Delta x^2}, \text{ se obtiene:}$$

$$T'(x_i) \cong \frac{T(x_{i+1}) - T(x_i)}{\Delta x} - O(\Delta x)$$

$$\text{donde } O(\Delta x) = \frac{T''(x_i)}{2}\Delta x \dots$$

Convirtiéndose esta ecuación en la expresión de la derivada primera en diferencias hacia adelante.

Segunda Derivada en Diferencias Finitas

A raíz de la siguiente ecuación:

$$T(x_{i+2}) = T(x_i) + T'(x_i)(2\Delta x) + \frac{T''(x_i)}{2}\overline{\Delta x^2} + \dots$$

Se procede a restarla de la siguiente ecuación:

$$T(x_{i+1}) = T(x_i) + T'(x_i)\Delta x + \frac{T''(x_i)}{2}\overline{\Delta x^2}$$

Obteniendo:

$$T''(x_i) = \frac{T(x_{i+2}) - 2T(x_{i+1}) + T(x_i)}{(\Delta x^2)} + O(\Delta x)$$

que es la ecuación de expresión de derivada segunda en diferencias hacia adelante.

La versión de la expresión de derivada segunda en diferencias hacia atrás es:

$$T''(x_i) = \frac{T(x_i) - 2T(x_{i-1}) + T(x_{i-2}))}{\Delta x^2} + O(\Delta x)$$

Y la versión centrada:

$$T''(x_i) = \frac{T(x_{i+1}) - 2T(x_i) + T(x_{i-1}))}{(\Delta x^2)} + O(\Delta x)$$

Ecuación de calor en una dimensión

Realizando un balance de energía sobre el elemento de la Figura 3, en el que se supone existe flujo de calor unidimensional tenemos en un tiempo Δt :

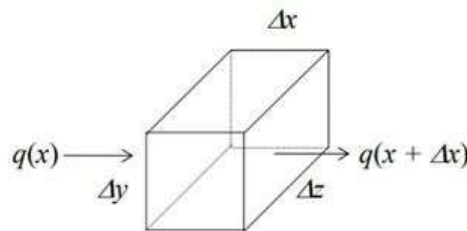


Figura 3: Ecuación del calor en una dimensión

donde, entrada-salida=acumulación

$$q(x)\Delta y\Delta z\Delta t - q(x + \Delta x)\Delta y\Delta z\Delta t = \Delta x\Delta y\Delta z\delta C\Delta T, \text{ ó}$$

$$\frac{q(x) - q(x + \Delta x)}{\Delta x} = \rho C \frac{\Delta T}{\Delta t}$$

$$-\frac{\partial q}{\partial x} = \rho C \frac{\partial T}{\partial t}$$

$$\frac{\partial^2 T}{\partial x^2} = k \frac{\partial T}{\partial t},$$

que es la ecuación de conducción de calor en una dimensión.

La resolución de esta ecuación de calor puede darse a través del método de diferencias finitas. Comúnmente, los métodos empleados para las diferencias finitas, se conocen como métodos explícitos e implícitos (dentro de este también podemos encontrar el método Crank-Nicolson). (Universidad Nacional de Misiones, s.f.)

Método Explícito

Se comienza por sustituir la derivada segunda espacial por un cociente de diferencias. En diferencias centradas tendremos que: (Universidad Nacional de Misiones, s.f.)

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{T_{i+1}^l - 2T_i^l + T_{i-1}^l}{\Delta x^2} + O(\overline{\Delta x}^{-2})$$

El superíndice l denota el tiempo ($l \Delta t$)

Para la derivada primera con respecto al tiempo se procede a usar una fórmula hacia adelante, se obtiene como resultado:

$$\frac{\partial T}{\partial x} \cong \frac{T_i^{l+1} - T_i^l}{\Delta t}$$

Se sustituyen las siguientes ecuaciones:

- $\frac{\partial^2 T}{\partial x^2} \cong \frac{T_{i+1}^l - 2T_i^l + T_{i-1}^l}{\Delta x^2} + O(\overline{\Delta x}^{-2})$
- $\frac{\partial T}{\partial x} \cong \frac{T_i^{l+1} - T_i^l}{\Delta t}$

en la ecuación de calor, y se obtiene:

$$T_i^{l+1} = T_i^l + \lambda(T_{i+1}^l - 2T_i^l + T_{i-1}^l) \text{ con } \lambda = k \Delta t / \overline{\Delta x^2}$$

Convergencia y estabilidad

Convergencia significa que para Δx y $\Delta t \rightarrow 0$ a $T_a(x, t) \rightarrow T_{real}(x, t)$.

Estabilidad significa que los errores en cualquier etapa de la computación no se amplifican sino se atenúan a medida que la computación progresa.

(Universidad Nacional de Misiones, s.f.)

$\lambda \leq 1/2$	Métodos explícitos estables y convergentes.
$\lambda = 1/6$	Método explícito estable y se tiende a minimizar los errores de truncamiento.
$\lambda \leq 1/4$	La solución del método no oscila.
$\lambda \leq 1/12$	La solución de método puede oscilar, pero se asegura que los errores no crecen

Método Implícito

En el método implícito, la derivada espacial se calcula en un tiempo futuro $l + 1$. Por ejemplo

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{T_{i+1}^{l+1} - 2T_i^{l+1} + T_{i-1}^{l+1}}{\Delta x^2}$$

Cuando esta ecuación se remplace en la ecuación diferencial original ($k \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t}$) la ecuación resultante arroja muchas incógnitas. Por esa razón no puede ser resuelta con un método explícito, obteniendo como resultado un sistema de ecuaciones lineales algebraicas que proceden a resolverse

simultáneamente a cada paso temporal l o Δt . (Universidad Nacional de Misiones, s.f.)

Marco Conceptual

Python

Python es un lenguaje de programación de alto nivel, fue creado por Guido Van Rossum, un programador holandés, a finales de los 80. Su primer uso fue destinado al manejo de excepciones y tener interfaces con Amoeba, sistema operativo con el que trabajaba en sus inicios. (Challenger-Pérez, Díaz-Ricardo, & Becerra-García, 2014)

Python puede ser utilizado en múltiples plataformas y sistemas operativos, entre esos se pueden destacar como los más populares a Windows, Mac Os y Linux. Al igual que algunos lenguajes de programación, Python también consta con un ámbito específico. A través de este lenguaje se puede desarrollar software para aplicaciones científicas, para comunicaciones de red, para aplicaciones de escritorio con interfaz gráfica de usuario (GUI), para crear juegos y para aplicaciones web. (Fernández Montero, 2012, pág. 2)

Características del lenguaje

- Es interpretado y multiplataforma, debido a estas características no es necesario compilar el código para su ejecución, debido a que ya existe un intérprete encargado de leer el fichero fuente y se encarga de ejecutarlo. Por esta razón, es posible la ejecución del mismo código en distintas plataformas y sistemas operativos. Para que el código sea leído

y ejecutado correctamente en diversas plataformas, cada plataforma deberá tener instalado el intérprete que será nativo para cada plataforma.

(Fernández Montero, 2012, pág. 4)

- Es open-source, es decir, toda persona interesada puede aportar en su desarrollo y divulgación. Además, no hace falta adquirir ninguna licencia para poder distribuir cualquier software desarrollado en este lenguaje.

(Fernández Montero, 2012, pág. 2)

Librería Tkinter

“Tk/Tcl ha sido durante mucho tiempo una parte integral de Python. Proporciona un conjunto de herramientas robusto e independiente de la plataforma para administrar ventanas.” (Python Software Foundation, 2006)

Tkinter es una librería de Python mayormente usada para el desarrollo de interfaces de usuario. Dentro de sus extensas ventajas, podemos encontrar que al ser una librería que viene por default en el lenguaje de programación Python, otorga a sus usuarios una experiencia más grata y ágil.

Tkinter no es la única librería para Python que se especializa en la creación de interfaces gráficas; sin embargo, cuenta con ventajas que la permiten destacar frente a otras. Entre esas ventajas podemos resaltar que esta librería viene de la mano con Python en casi todas las plataformas, su sintaxis es bastante clara, volviéndola fácil de aprender y manipular, además, que cuenta con una documentación completa que permite que los usuarios puedan acceder a la información de esta librería. (Unipython, 2020)

SymPy

“SymPy es una biblioteca de Python para matemáticas simbólicas. Su propósito es llegar a ser un sistema de álgebra por computadora (CAS) completo manteniendo el código tan simple como sea posible para poder ser legible y extensible de manera fácil. SymPy está escrito en Python enteramente.” (SymPy Development Team, 2021)

SymPy es una librería de Python completamente gratis, escrita en su totalidad en Python y que se ejecuta en el mismo lenguaje de programación. Symbolic Python (SymPy) es una librería que nos permite resolver cálculos simbólicos o también conocido como algebra computacional. Existen una gran cantidad de software comerciales que realizan tareas similares a las de SymPy, entre estos están: Maple, Matlab, Mathematica, y a pesar que son software muy reconocidos, estos requieren de una licencia de uso que frecuentemente se vuelve poco accesible. En cambio, SymPy, se distribuye bajo una licencia BSD, licencia que viene de una familia de licencias permisivas de software libre, que le permite el uso libre de la misma. (Santos, 2016)

Librería Matplotlib

“Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python. Matplotlib hace que las cosas fáciles sean fáciles y las difíciles sean posibles.” (Hunter, 2007)

Matplotlib es una biblioteca para crear gráficos a partir de datos contenidos en listas, vectores, en el lenguaje de programación Python y en su extensión matemática NumPy. Sus gráficos pueden variar desde gráficos

presentadas en dos dimensiones o tres dimensiones. Esta librería es mayormente utilizada en proyectos que involucren el análisis de datos, dado que, la visualización de datos es un componente del análisis de datos. (Unipython, 2017)

Librería NumPy

NumPy es una librería de Python que se especializa en el cálculo numérico y el análisis de datos. El único requisito para poder acceder a esta librería es tener instalado Python. Dentro de las ventajas de NumPy, podemos encontrar que es una librería fácil de usar debido a que su sintaxis de alto nivel la hace accesible y productivo para los programadores de cualquier nivel. Además, es de fuente abierta y al igual que SymPy, trabaja con una licencia BSD. (NumPy, 2021)

Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Tiene un rico ecosistema de extensiones para otros lenguajes como C ++, C #, Java, Python, PHP, Go. (VSC Development Team, 2021)

“VS Code es mi editor más utilizado y favorito. Me encanta poder personalizar el editor: cambiar el diseño, los iconos, las fuentes y la combinación de colores es muy fácil.” (Kodmad, 2022) Visual Studio Code es un editor de texto plano que fue desarrollado por Windows como una alternativa que sustituya a los editores de texto plano que vienen instalados por default en los sistemas operativos. Una de las ventajas más grandes que nos ofrece este

editor de texto, es la función IntelliSense que permite resaltar la sintaxis del código y nos ofrece la herramienta de autocompletar. (SoftZone, 2021)

CAPÍTULO III. METODOLOGÍA DE LA INVESTIGACIÓN

La finalidad de este capítulo es definir la metodología de investigación empleada dentro del desarrollo de este trabajo, además se busca analizar cuál será la técnica de investigación adecuada para la obtención de datos que permita alcanzar los objetivos propuestos para la solución tecnológica.

Metodología de la investigación

Para poder desplegarlos a lo largo de este capítulo, es primordial hacer énfasis en la definición de “metodología de investigación”. Carlos Muñoz Rocha define el término como:

Ciertamente que la metodología de la investigación comprende el estudio del método o métodos empleados en la investigación, el proceso de investigación, las técnicas de investigación documental, las técnicas de investigación de campo, la redacción de informes científicos, el análisis y el tratamiento estadístico de la información obtenida y, dependiendo del grado de complejidad del objeto u objetos investigados y el empleo de herramientas matemáticas más profundas para analizar los fenómenos. (2016, pág. 5)

Es decir, podemos deducir que la metodología de la investigación se refiere a cómo un investigador diseña sistemáticamente un estudio para garantizar resultados verídicos que cumplan con los propósitos de la investigación.

Dentro de las investigaciones científicas existen dos enfoques principales, el enfoque cuantitativo y el enfoque cualitativo. Se pueden resumir estos enfoques como:

- Enfoque cualitativo: Es un método de investigación sin medidas numéricas, mediante la realización de encuestas, entrevistas, describiendo, desde el punto de vista del investigador, y reconstruyendo los hechos, generalmente sin contrastar las hipótesis necesarias. (Cortés Cortés & Iglesias León, 2004)
- Enfoque Cuantitativo: Este método hace mediciones numéricas centrales para su investigación, utilizando el monitoreo de procesos como recopilación y análisis de datos para llegar a las respuestas de sus preguntas de investigación. La recogida y medición de parámetros y la recogida de frecuencias y estadísticas de la población en la que entran las encuestas se utiliza para contrastar hipótesis predeterminadas. En este enfoque, es necesario utilizar análisis estadístico, formular ideas y preguntas de investigación, objetivos e hipótesis, seleccionar variables de proceso y, durante el proceso de cálculo, las hipótesis se comparan entre sí. (Cortés Cortés & Iglesias León, 2004)

El enfoque representativo de este trabajo será el cualitativo, puesto que, con este se busca analizar y comprender la problemática y llegar a una solución

tecnológica óptima que satisfaga los requerimientos de la Universidad en cuestión.

En cuanto al tipo de investigación, esta será de tipo descriptiva. Según Cortés e Iglesias, “Los estudios descriptivos buscan especificar las propiedades, las características y los perfiles importantes de personas, grupos, comunidades o cualquier otro fenómeno que se someta a un análisis. Describen situaciones, eventos o hechos, recolectando datos sobre una serie de cuestiones y se efectúan mediciones sobre ellas, buscan especificar propiedades, características y rasgos importantes de cualquier fenómeno que se analice. Estos estudios presentan correlaciones muy incipientes o poco elaboradas.” (Cortés Cortés & Iglesias León, 2004, pág. 20)

El método de obtención de datos que será a través de la “Observación Participante”, pues según Cortés e Iglesias, “La observación es una de las técnicas cualitativas más aplicada en la etnografía y precisamente en el marco educativo, por la riqueza de su información y la influencia de la misma en la formación del estudiante durante el proceso de enseñanza-aprendizaje.”

Una de las ventajas del empleo del método de Observación es que se puede acceder a un punto de vista específico, sin manipularlo o modificarlo. Según Olabuénaga e Ispizua, la observación común puede convertirse fácilmente en una herramienta de investigación social y en técnica científica de recogida de información si se efectúa (Olabuénaga Ruiz & Ispizua, 1989):

- ✓ Planificándola sistemáticamente en fases, aspectos, lugares y personas.
- ✓ Controlándola y relacionándola con proposiciones y teorías sociales, planteamientos científicos y explicaciones profundas.
- ✓ Sometiéndola a controles de veracidad, de objetividad, de fiabilidad y de precisión.

Fueron observadas las necesidades y la participación de los estudiantes a los que les beneficiaría positivamente la solución tecnológica desarrollada con este trabajo y estos fueron las observaciones destacadas:

- ✓ La funcionalidad exitosa de la interfaz será de apoyo académico para los estudiantes de la carrera de Ingeniería en la Universidad Católica Santiago de Guayaquil.
- ✓ Al ser una interfaz intuitiva y teniendo en cuenta las capacidades tecnológicas de los estudiantes, la manipulación de la interfaz no presentará un grado de dificultad alto para las personas involucradas en su uso.

Para llegar a las observaciones previamente mencionadas, se realizó una indagación meticulosa en el perfil de los estudiantes que se vean involucrados en el uso de la interfaz.

La investigación de este trabajo de titulación ha sido dividida en dos secciones. La sección uno, es la indagación y revisión bibliográfica acerca del tema, debido a que el tópico “Ecuaciones Diferenciales Parciales Parabólicas”

es un tema extra de lo que consta en el syllabus. La segunda sección de la investigación es determinar cuáles son las herramientas más competentes para el desarrollo de la interfaz.

Metodología de desarrollo

El método de desarrollo de software que encaja y permite encaminar el proyecto hacia el cumplimiento de sus objetivos, es la metodología iterativa. Esta metodología permite obtener un producto final más completo y eficaz, sobre todo ya que es la fusión de dos metodologías de desarrollo ya existentes. Maida y Pacienza describen esta metodología como (Maida & Pacienza, 2015):

Permite construir el proyecto en etapas incrementales en donde cada etapa agrega funcionalidad. Estas etapas, consisten en requerimientos, diseño, codificación, pruebas y entrega. Permite entregar al cliente un producto más rápido. Esta metodología fue escogida debido a las características y ventajas que representa (Maida & Pacienza, 2015):

- ✓ Reduce los riesgos ya que provee visibilidad sobre el progreso de las nuevas versiones.
- ✓ Provee retroalimentación a través de la funcionalidad mostrada.
- ✓ Permite atacar los mayores riesgos desde el inicio.
- ✓ Las pruebas y la integración son constantes.
- ✓ Resulta más sencillo acomodar cambios al acortar el tamaño de los incrementos.

- ✓ Se puede planear en base a la funcionalidad que se quiere entregar primero.
- ✓ La solución se va mejorando en forma progresiva a través de las múltiples iteraciones, incrementa el entendimiento del problema y de la solución por medio de los refinamientos sucesivos.

En este trabajo, las iteraciones se repetirán hasta que se obtenga una solución tecnológica que satisfaga las necesidades del cliente. Los objetivos de las futuras iteraciones se establecen en función de la evaluación de iteraciones previas. Los clientes, en este caso, los estudiantes que cursan la materia ecuaciones diferenciales o los estudiantes que necesitan de la herramienta, serán quienes evalúen el producto y quienes sugerirán mejoras. Estas iteraciones se repetirán hasta obtener un producto que cumpla con los requerimientos inicialmente propuestos.

Esta investigación no tiene como objetivo determinar algún comportamiento particular de la población ni de un grupo de la población, este trabajo de titulación es un aporte, desde un punto de vista tecnológico, para la construcción de material didáctico de mayor impacto sobre temas de mayor nivel en las matemáticas. Debido a esta razón y a que este trabajo de titulación es un requerimiento solicitado por la universidad, no ha sido necesario acudir a técnicas de recolección de datos ya que este trabajo es la construcción de un requerimiento en base a una petición de la institución universitaria.

CAPÍTULO IV. DESARROLLO DEL PROYECTO

En este capítulo se describirá detalladamente todo en cuanto al desarrollo de la interfaz como solución. De igual forma, se presentará las herramientas utilizadas para el funcionamiento óptimo de esta interfaz.

Introducción

Siendo “Ecuaciones Diferenciales” una de las asignaturas perteneciente a la malla curricular de la Universidad Católica de Santiago de Guayaquil, carrera Computación, se procedió a diseñar y desarrollar una interfaz que permita resolver y graficar Ecuaciones Diferenciales Parciales de tipo Parabólico, con el fin de beneficiar a los usuarios que la implementen dentro de su vida universitaria.

Objetivo

Gestionar el desarrollo de un programa capaz de graficar Ecuaciones Diferenciales Parciales Parabólicas para la Universidad Católica Santiago de Guayaquil, con el fin de beneficiar a los usuarios que la implementen.

Responsable

Siendo una implementación para la carrera de Computación de la Universidad Católica Santiago de Guayaquil, y al ser una aplicación abierta, no existe un responsable específico.

Descripción de la interfaz

La solución tecnológica está implementada completamente bajo el lenguaje de programación interpretado Python, el mismo posee dentro de sus

múltiples ventajas, el hecho de ser gratuito, ya que es un lenguaje calificado como open source o también denominado de código abierto, y es multiplataforma por su adaptabilidad a todas las plataformas disponibles en el mercado. Así mismo, contiene diseños y gráficas, que son el resultado de la solución tecnológica, que fueron desarrollados bajo la implementación de librerías del mencionado lenguaje de programación.

Arquitectura de la Interfaz

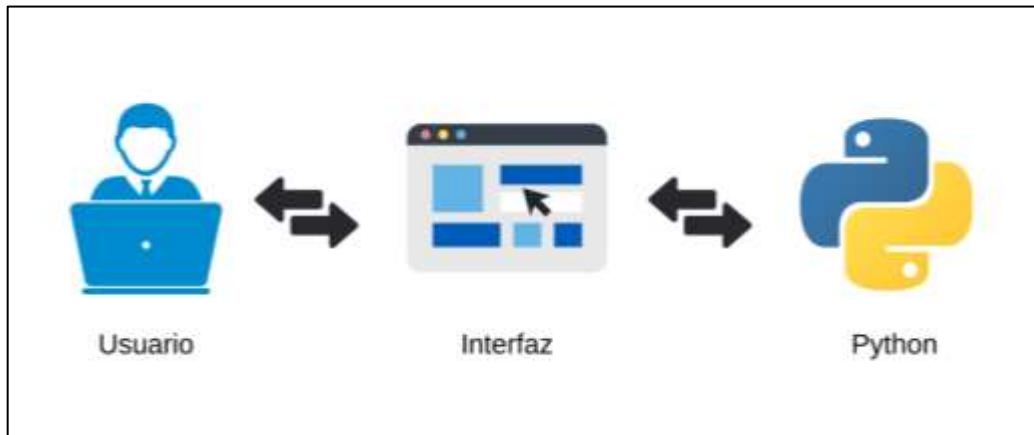


Figura 4: Arquitectura de la solución. Elaborado por el autor.

La arquitectura de la solución que ha sido propuesta abarca todos los componentes participantes dentro del desarrollo de la interfaz. En la figura #2, se aprecia la relación bidireccional que existe entre el usuario y la interfaz, se define como una interacción de doble dirección, debido a que el usuario hace el ingreso de los datos en la interfaz y esta le devuelve una solución analítica y una gráfica. Por otro lado, tenemos la relación bidireccional entre la interfaz y el lenguaje de programación, Python.

Herramientas Tecnológicas

Entre las opciones para el desarrollo de la interfaz se procedió a hacer una comparativa entre distintas alternativas que podían ser empleados para la solución del cálculo matemático. Estas alternativas son Python y Matlab.

- Matlab es un lenguaje de cálculo técnico desarrollado por MathWorks, este lenguaje es usado principalmente por ingenieros y científicos para una gran variedad de aplicaciones que buscan explotar el poder de las matemáticas computacionales. (MathWorks, s.f.).
- Python, el único lenguaje de programación utilizado para el desarrollo completo de la solución tecnológica propuesta. Dentro de esta herramienta se hicieron uso de distintas librerías especializadas que se encuentran disponibles dentro del mismo lenguaje. Se utilizó esta herramienta ya que permite una implementación rápida y fácil. Javier Pastor (Pastor, 2020) dice que hoy en día Python es el lenguaje de programación más popular, debido a su versatilidad y adaptabilidad.

Python	Matlab
Lenguaje de código abierto	Lenguaje de código cerrado y un producto comercial patentado.
Lenguaje equipado con una serie de bibliotecas ordinarias para realizar tareas de programación y computación	Uno de los lenguajes de programación mejores diseñados para la informática.

totalmente diferentes.	
Diseñado para todo tipo de operaciones genéricas.	Diseñado para tareas particulares

Tabla 2: Comparación entre Matlab y Python

Opiniones de usuarios (Quora, s.f.):

- “Tanto MATLAB como Python son igualmente buenos. Si quieres escribir los scripts realmente rápido, entonces MATLAB es una mejor opción. En Python, tienes que saber qué paquete tiene qué funciones y, por tanto, debes importarlo. Ten en cuenta que MATLAB es un software propietario y consume alrededor de 450 MB de RAM incluso cuando está en reposo. Por otro lado, Python es de código abierto y no consume mucha RAM. Yo sugeriría que es mejor ir con la alternativa de código abierto, Python. Para obtener la típica interfaz de MATLAB en Python, puedes instalar Spyder IDE.” - Isac Rajan (Mechanical Engineering Undergraduate at NITK Surathkal)
- “Mi experiencia es que es mucho más rápido desarrollar con Python, simplemente porque es más fácil de leer. Además, es un verdadero lenguaje orientado a objetos, y tiene muchas librerías externas, donde es probable que encuentres trozos que puedas utilizar directamente.” - Stéphane Poss
- “Python es una excelente opción. MATLAB está en la lista de los mejores, pero es comercial, por lo que está perdiendo popularidad, mientras que los laboratorios de python (numpy, scipy) están

aumentando cada vez más.” -Corrado Mencar, Università degli Studi di Bari Aldo Moro

Por consecuencia, se procedió a escoger a Python como el lenguaje de programación con el que se desarrollará este trabajo de titulación. Una de las principales razones por la que se escogió Python es debido a que es un lenguaje de código abierto, convirtiéndolo en una herramienta más accesible que permitirá eficazmente el desarrollo de la interfaz.

A continuación, se detallará las herramientas tecnológicas empleadas para el desarrollo óptimo de la interfaz:

- Python
- Visual Studio Code, como editor de código. Para Jorge Ferreiro (Herranz, 2021), Product Manager en StreamYardApp, “En Visual Studio Code se puede trabajar rápido, de forma eficiente y tiene extensiones e integraciones creadas por usuarios que realmente hacen que la experiencia completa de usar el editor sea muy potente.” Ciertamente, existen una gran cantidad de editores de código en el mercado, y escoger uno depende de muchos factores, esto va desde el entorno en el que se va a desarrollar, la compatibilidad con ciertos lenguajes de programación o incluso las capacidades de los mismos.

Características de la interfaz

Página de inicio



Figura 5: Pantalla Inicial

La primera pantalla con la que se encontrará un usuario al correr este programa, será la Pantalla Inicio. En esta pantalla se encontrará información acerca de los factores más importantes de este proyecto. Es decir, en la pantalla se logrará apreciar: el título de este trabajo de titulación "Interfaz de resolución de Ecuaciones Diferenciales Parciales Parabólicas", junto a una gráfica representativa del tema a trabajar, un logo perteneciente a la entidad universitaria para la cual se está desarrollando este producto y finalmente un botón "Inicio". Este botón es el conector entre la pantalla Inicial y la pantalla en la cual se desarrollará la resolución de las ecuaciones diferenciales parciales parabólicas.

Pantalla de resolución

The screenshot shows a software interface for solving a parabolic partial differential equation using the explicit finite difference method. The window title is 'Ecuaciones Diferenciales Parciales Parabólicas'. Below the title bar, there are tabs for 'Método Explicito de Diferencias Finitas' (selected) and 'Método Implícito de Diferencias Finitas'. The main area is titled 'Método Explicito' and displays the equation $\frac{\partial^2 T}{\partial x^2} = K \frac{\partial T}{\partial t}$. The interface includes several input fields: 'L:' for length, 'Condición Inicial:' for initial condition, 'Condiciones de frontera:' with sub-fields for 'U(0, t):' and 'U(L, t):', 'k:' for the constant, 'U(x,0):' for initial temperature, 'Paso en x:' for spatial step, 'Paso en t:' for time step, and 'n:' for the number of time steps. There are also radio buttons for 'Matriz de Resultados' and 'Tabla de Resultados'. At the bottom, there are two buttons: 'Graficar' and 'Calcular Resultados'.

Figura 6: Pantalla de resolución

En la figura #6 se puede apreciar la interfaz que resolverá la solución tecnológica propuesta. Esta pantalla está dividida en dos pestañas: método implícito y método explícito. Para ambos métodos de resolución, el usuario tendrá que hacer el ingreso de las mismas variables de entrada:

- L: Longitud de la barra.
- Condiciones de frontera
 - U(0, t) Condición de frontera inicial
 - U(L, t) Condición de frontera final
- k: constante que depende del material.
- Condición inicial: U(x, 0) cuando la temperatura es 0.

- Paso en x
- Paso en t
- n: Número de niveles en t.

Adicionalmente, dentro de la pantalla de resolución, se encuentran los botones principales: Graficar y Tabla de Resultados. Luego de ingresar los datos para la ecuación diferencial parcial parabólica que se busca resolver, el botón graficar abre una tercera pestaña mostrando la gráfica obtenida. Esta gráfica se ve parametrizada por los valores añadidos en la pantalla de resolución. Por otro lado, se encuentra el botón Tabla de Resultados que abre una nueva ventana, mostrando los puntos obtenidos, de los datos ingresados, en forma de matrices.

Dentro de esta pantalla también se puede apreciar una barra de menú, que se encuentra situado en la parte superior de la pantalla. La barra de menú cuenta con dos menús: el menú “Archivo” y el menú “Información”.

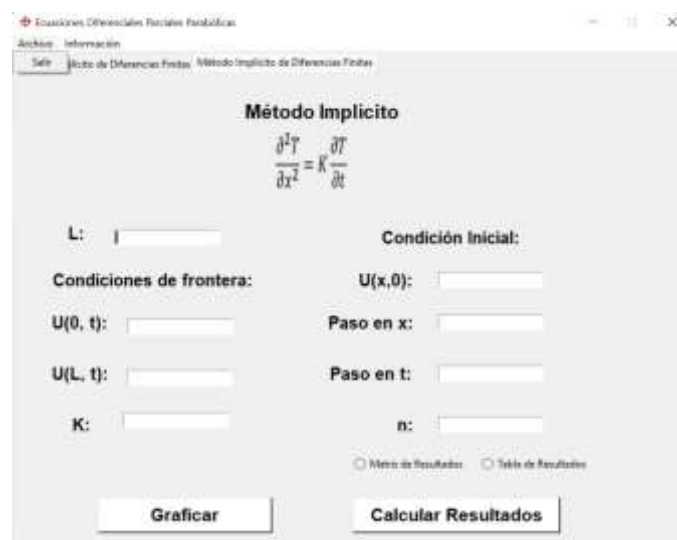


Figura 7: Menú Archivo

Dentro del menú Archivo, se despliega en forma de cascada una nueva opción, la opción Salir, al clicar sobre esta opción la interfaz cierra todas las ventanas abiertas.

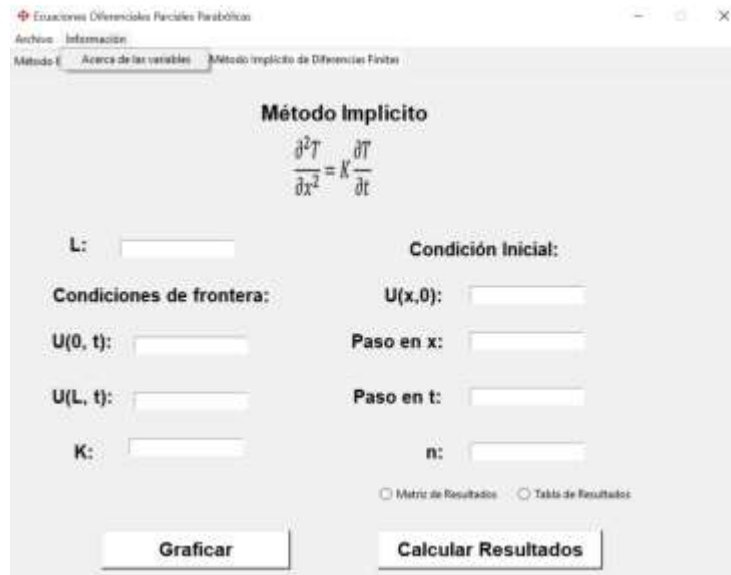


Figura 8: Menú Información

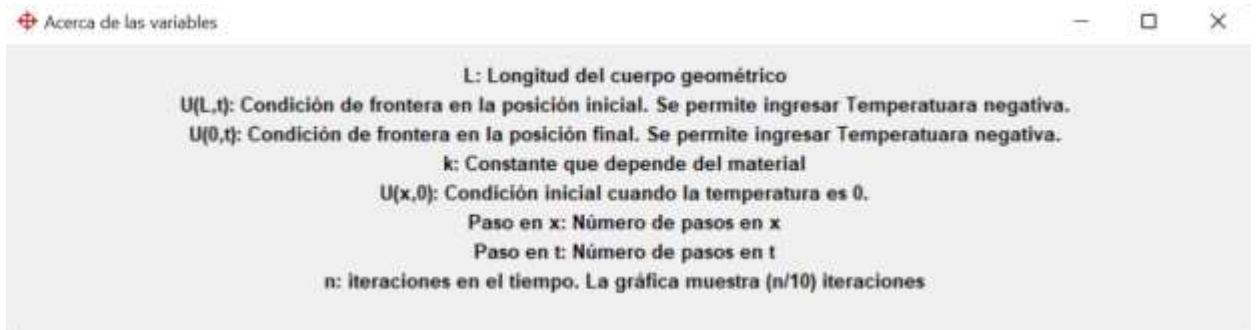


Figura 9: Pantalla Acerca de las variables

Por otro lado, dentro del menú de Información, se despliega, en forma de cascada, una nueva opción desde la cual se puede acceder a una nueva ventana donde se indica la información de las variables. De esta manera, el usuario reconocerá la información que ingresará dentro de cada variable.

Finalmente, en caso que el usuario ingrese datos donde los incrementos en x no coincidan con la longitud de la barra, la interfaz arroja la siguiente advertencia y procede en no resolver la ecuación propuesta.



Figura 10: Pantalla de advertencia.

Método Explícito

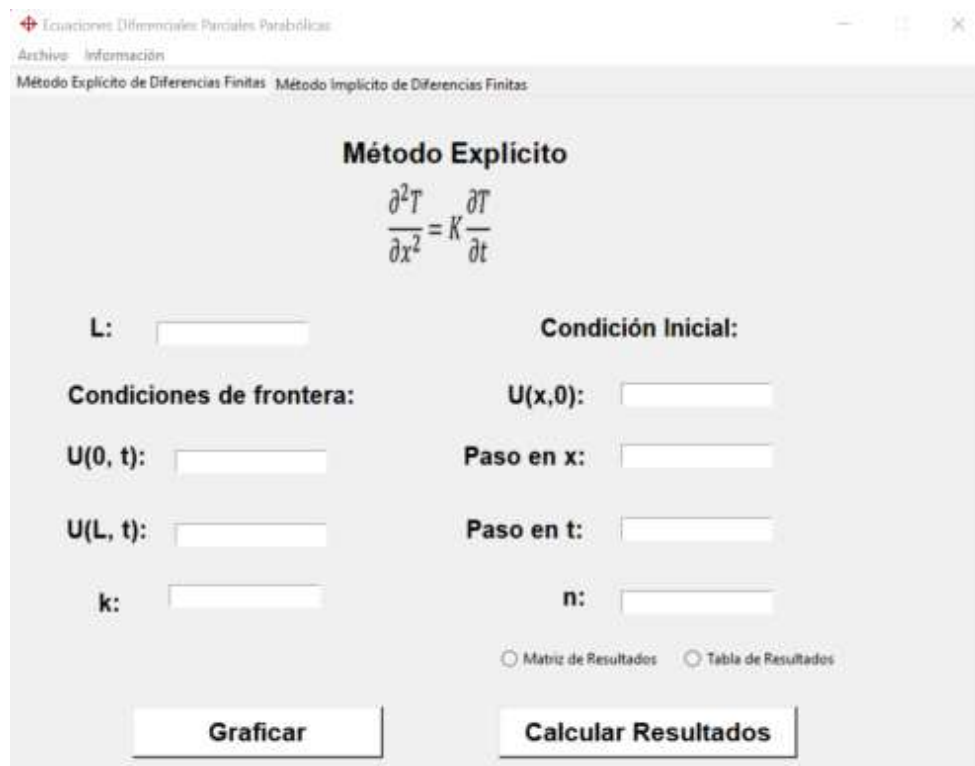
A software interface for the explicit method. It features a title bar "Ecuaciones Diferenciales Parciales Parabólicas" and a menu "Archivo Información". The main content area is titled "Método Explícito" and displays the equation $\frac{\partial^2 T}{\partial x^2} = K \frac{\partial T}{\partial t}$. Below the equation are input fields for "L:", "Condición Inicial:", "Condiciones de frontera:", "U(0, t):", "U(L, t):", "k:", "U(x,0):", "Paso en x:", "Paso en t:", and "n:". At the bottom, there are two radio buttons for "Matriz de Resultados" and "Tabla de Resultados", and two buttons: "Graficar" and "Calcular Resultados".

Figura 11: Pantalla método explícito

El método de resolución explícito, es el resultado del despeje de la ecuación general, obteniendo como resultado el valor del nodo $u_{i,j+1}$. A partir de

la obtención del valor del nodo antes mencionado, se procede a realizar una matriz, donde el resultado entrega los valores faltantes de los nodos restantes. Además, en este método se considera el valor de lambda, donde su resultado debe ser menor a $\frac{1}{2}$, ya que esa es una condición de estabilidad que permitirá tener un método estable y convergente.

```
#Lambda
#(<1/2) condicion de estabilidad
lamb= dt/(k*dx**2)
P = lamb
Q = 1-2*lamb
R = lamb

#Calcula U para cada tiempo +dt
j=0
while not(j>=ultimot):
    for i in range(1, ultimox, 1):
        u[i,j+1] = P*u[i-1,j] + Q*u[i,j] + R*u[i+1,j]
    j=j+1
```

Figura 12: Método Explícito desarrollado en Python

En caso que el usuario introduzca valores que no coincidan con la condición de estabilidad, la interfaz muestra una ventana de advertencia indicándole al usuario que su gráfica resultante no será una parábola y se le permitirá decidir si desea continuar con el proceso o corregir los valores. Si el usuario desea no continuar, se abrirá una ventana de advertencia pidiéndole que ingrese nuevamente sus datos.

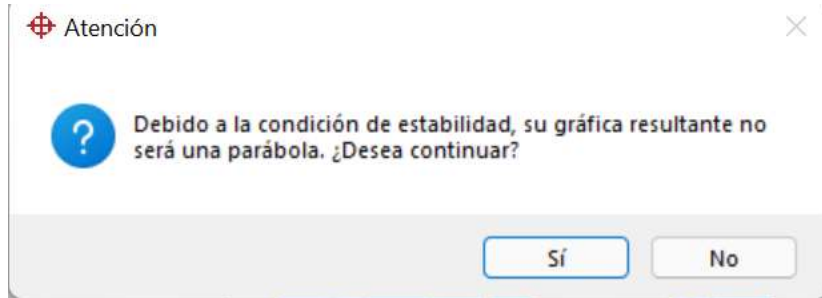


Figura 13: Pantalla de incumplimiento de la condición de estabilidad

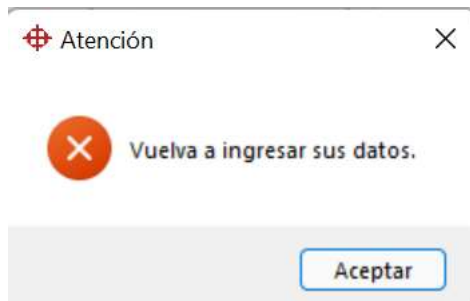


Figura 14: Ventana de reingreso de datos.

Método Implícito

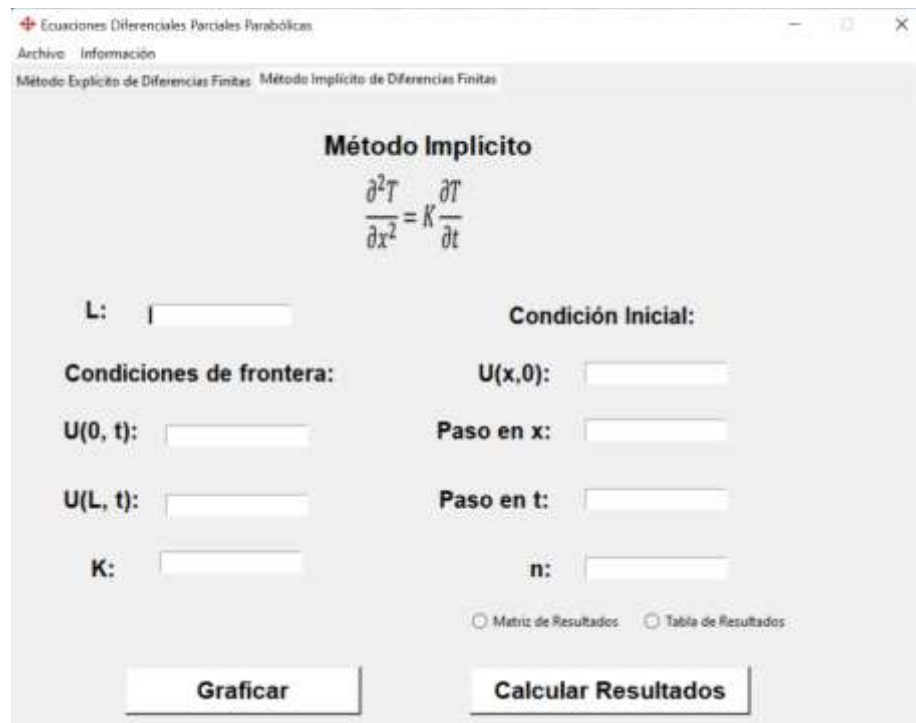


Figura 15: Método Implícito

A diferencia del método explícito, en este método los resultados se obtienen a través de la creación de un sistema lineal tridiagonal, es decir, sus únicos elementos distintos de cero son los que se encuentran tanto en la diagonal principal como en las diagonales contiguas por encima y debajo de esta.

```
#Lambda
lamb= dt/(k*dx**2)
P = lamb
Q = -1-2*lamb
R = lamb

#Bucle para sistema de ecuaciones
j = 1
while not(j>=n):
    u[0,j] = Ta
    u[m-1, j] = Tb
    #Matriz de ecuaciones
    tamano = m-2
    #Inicializar la matriz A
    A = np.zeros(shape=(tamano, tamano), dtype=float)
    #Inicializar la matriz B
    B = np.zeros(tamano, dtype=float)
    for f in range(0, tamano, 1):
        if(f>0):
            A[f, f-1]=P
            A[f,f]=Q
            if(f<(tamano-1)):
                A[f,f+1]=R
            B[f]= -u[f+1, j-1]
    B[0]=B[0]-P*u[0,j]
    B[tamano-1]=B[tamano-1]-R*u[m-1, j]
    #Resuelve sistema de ecuaciones
    C=np.linalg.solve(A, B)

    #copia resultados a u[i,j]
    for f in range(0, tamano, 1):
        u[f+1, j]=C[f]

    #Proximas iteraciones
    j=j+1
```

Figura 16: Método Implícito desarrollado en Python

Pantalla de gráfica

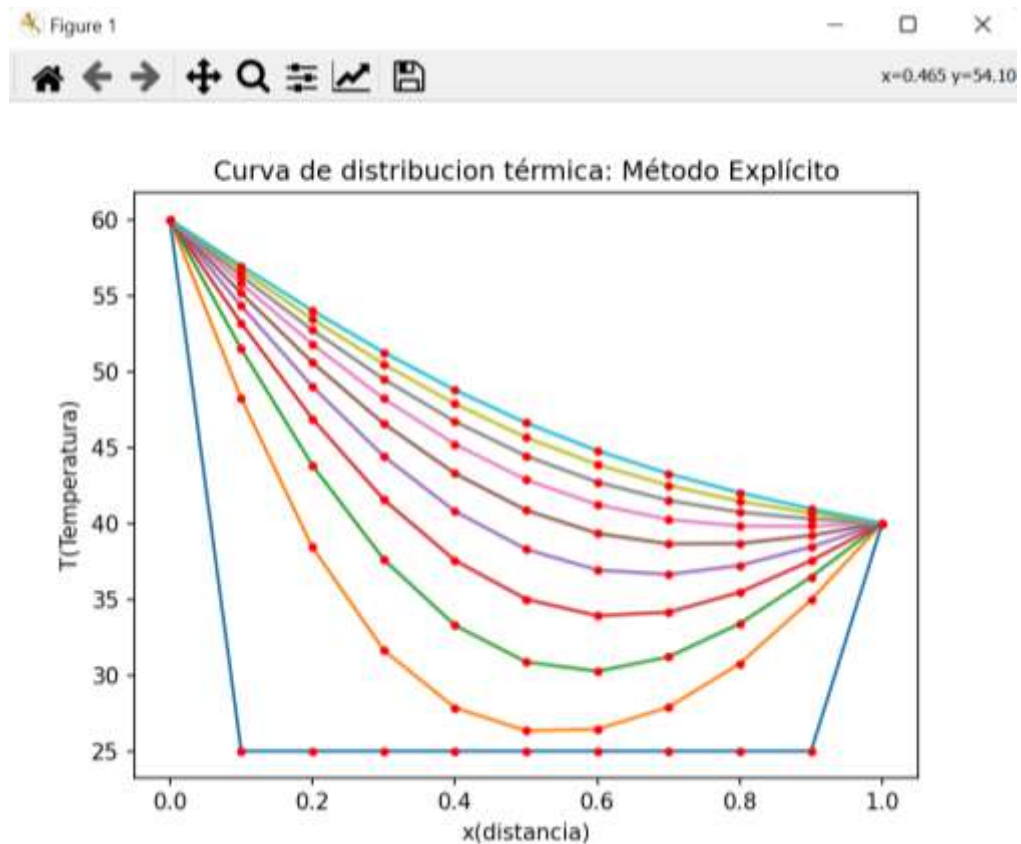


Figura 17: Pantalla de gráfica

Como resultado de un ejercicio de prueba, se pudo obtener la gráfica demostrada en la figura 13. La pantalla de la gráfica, muestra la distribución de la temperatura en el cuerpo geométrico. Se muestra también los nombres de los ejes: $x(\text{distancia})$ para el eje x , y $T(\text{Temperatura})$ para el eje y , también está reflejado, en la parte derecha superior, los valores en el eje x como en el eje y , de acuerdo a como se vaya desplazando el cursor sobre la gráfica. Así mismo, en el título de la gráfica resultante se obtiene el método por el cual se trabajó. Adicionalmente, esta pantalla cuenta con una barra de menú que permite desplazarse en x e y , seleccionar una porción del gráfico para agrandarlo,

resetear las modificaciones para volver a la gráfica inicial y una opción para guardar la gráfica dentro del ordenador del usuario.

Pantalla Matriz de Resultados



Figura 18: Pantalla Tabla de Resultados

Dentro de esta pantalla se obtendrán los resultados obtenidos a partir de la gráfica. Los valores son presentados en forma de matriz, revelando los valores de la evolución de la temperatura en la última curva de la gráfica.

Pantalla Tabla de Resultados



T-°C	x
80.0	0.0
73.02	0.3
66.41	0.6
60.41	0.9
55.02	1.2
50.0	1.5

Para navegar entre los resultados, use las flechas de su teclado.

Figura 19: Pantalla de puntos de la última iteración

En esta ventana, se apreciarán, de acuerdo a los incrementos en x , una tabla con valores específicos de la última iteración (la última curva de la gráfica) de la solución.

CONCLUSIONES

La constante evolución tecnológica y la fuerte demanda de implementación de las mismas, es un sinónimo de la necesidad de buscar soluciones tecnológicas que permitan simplificar procesos de la vida cotidiana. Por esta razón, se buscó desarrollar una solución tecnológica que permita obtener una gráfica de una Ecuación Diferencial Parcial Parabólica.

Con el fin de atender tal requerimiento, se procedió a recopilar toda la información perteneciente al tema e identificar las características que permitirían obtener el resultado deseado.

Tras finalizar este proceso, se procede a analizar las opciones de herramientas tecnológicas que permitirán la creación de la interfaz, concluyendo que se logró implementar correctamente una interfaz que permitirá reforzar el tema “Ecuaciones Diferenciales Parciales de tipo Parabólico” a los usuarios que hagan uso de la misma.

RECOMENDACIONES

Se recomienda a la entidad universitaria a la cual le pertenece este proyecto de titulación, que la manipulación de la interfaz sea para usuarios que ya tienen base sobre el tema de Ecuaciones Diferenciales Parciales Parabólicas.

Por otro lado, se recomienda la idea de parametrizar y recrear esta interfaz con temas estudiados en la misma rama. De esta manera, los usuarios podrán acudir a más de una herramienta que les permita reforzar sus conocimientos.

Con el fin de que esta interfaz se encuentre más accesible para sus usuarios, se le recomienda a la carrera de computación la creación de una página web donde sean colocados los programas desarrollados por sus estudiantes. El propósito de esta página web sería que tanto usuarios de la Universidad Católica de Santiago de Guayaquil como usuarios de otras universidades tengan la oportunidad de acceder a este software y aportar positivamente a la comunidad académica.

Finalmente, como idea para futuros trabajos, se propone la implementación de resolución de Ecuaciones Diferenciales Parciales Parabólicas, a través de distintos métodos de resolución, y de esta manera se logrará obtener un resultado comparativo entre los distintos métodos.

REFERENCIAS

- Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*. Obtenido de <https://www.redalyc.org/articulo.oa?id=181531232001>
- Chapra, S., & Canale, R. (2007). *Métodos numéricos para ingenieros*. McGraw-Hill Interamericana.
- Chasnov, J. R. (2021). *Differential Equations*. Hong Kong.
- Cortés Cortés, M. E., & Iglesias León, M. (2004). *Generalidades sobre Metodología*. Ciudad del Carmen, Campeche, México : Universidad Autónoma del Carmen.
- Díaz, M. P. (2015). *Desarrollo de las Ecuaciones Diferenciales Parciales Parabólicas mediante Diferencias Finitas, Elementos Finitos y Meshless*.
- Fernández Montero, A. (2012). *Python 3 al descubierto*. Madrid: RC Libros.
- Herranz, A. (2021). *Xataka*. Obtenido de <https://www.xataka.com/otros/editor-codigo-favorito-programadores-profesionales-nos-dan-su-respuesta>
- Hunter, J. D. (2007). *Matplotlib: visualización con Python*. Obtenido de <https://matplotlib.org/>
- Kodmad, P. (2022). *Visual Studio Code*. Obtenido de <https://code.visualstudio.com/>
- Kreyszig, E. (2011). *Advanced Engineering Mathematics* (10th ed ed.). Hoboken, NJ: John Wiley.
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Pontificia Universidad Católica Argentina Sandta María de los Buenos

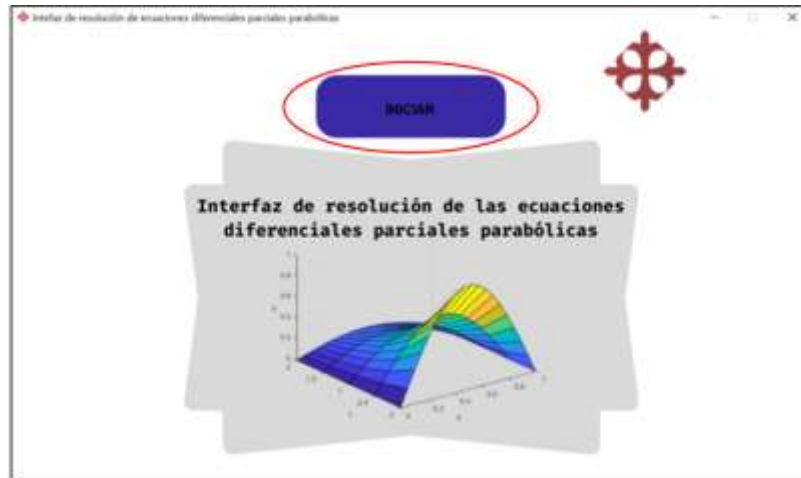
- Aires, Buenos Aires. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- MathWorks. (s.f.). *MathWorks*. Obtenido de What Is MATLAB?: <https://la.mathworks.com/discovery/what-is-matlab.html>
- Moreno, F. M. (2015). Desarrollo analítico y numérico de las Ecuaciones Diferenciales Parciales Clásicas.
- Muñoz Rocha, C. I. (2016). *Metodología de la Investigación*. Distrito Federal: Oxford University Press México.
- Nagy, G. (2021). *Ordinary Differential Equations*.
- NumPy. (2021). *Numpy*. Obtenido de numpy.org
- Olabuénaga Ruiz, J. I., & Ispizua, M. A. (1989). *La descodificación de la vida cotidiana: métodos de investigación cualitativa*. Bilbao: Universidad de Deusto.
- Pastor, J. (2020). *Xataka*. Obtenido de <https://www.xataka.com/aplicaciones/python-basic-nuestra-era-lenguaje-programacion-moda-ieee-spectrum-todos-quieren-conocerlo>
- Python Software Foundation. (8 de 02 de 2006). *Interfaces gráficas de usuario con Tk*. Obtenido de <https://docs.python.org/es/3/library/tk.html>
- Quora. (s.f.). *Which is better for scientific calculations, MatLab or Python?* Obtenido de <https://www.quora.com/Which-is-better-for-scientific-calculations-MatLab-or-Python>
- Sandoval-Ruiz, C. E. (2019). Métodos numéricos en diferencias finitas para la estimación de recursos de Hardware FPGA en arquitecturas LFSR(n,k)

- fractales. *Ingeniería, investigación y tecnología*. Obtenido de <https://doi.org/10.22201/fi.25940732e.2019.20n3.032>
- Santos, P. d. (2016). *Numython: Una introducción a SymPy*. Obtenido de sympy.org/en/index.html
- SoftZone. (2021). Obtenido de Visual Studio Code: el editor de código de Microsoft que querrás instalar: <https://www.softzone.es/programas/utilidades/visual-studio-code/>
- Sympy Development Team. (2021). Obtenido de <https://www.sympy.org/en/index.html>
- Unipython. (2017). *Unipython*. Obtenido de <https://unipython.com/matplotlib-funciones-principales/>
- Unipython. (2020). *Unipython*. Obtenido de <https://unipython.com/tkinter-introduccion/>
- Universidad Nacional de Misiones. (s.f.). *Facultad de Ciencias Exactas, Químicas y Naturales. Aula Virtual*. Recuperado el 24 de 11 de 2021, de <http://www.aulavirtual-exactas.dyndns.org/claroline/backends/download.php?url=LzdfRURQLzctN19UZXh0b19EaWZlcmVuY2lhc19GaW5pdGFzLVNjaHZlem92LnBkZg%3D%3D&cidReset=true&cidReq=ANUM>
- VSC Development Team. (8 de 12 de 2021). *Visual Studio Code*. Recuperado el 2021, de <https://code.visualstudio.com/docs/?dv=win64user>
- Zill, D. (2009). *Ecuaciones Diferenciales con aplicaciones de modelado*. Cengage Learning Editores S.A. de C.V.

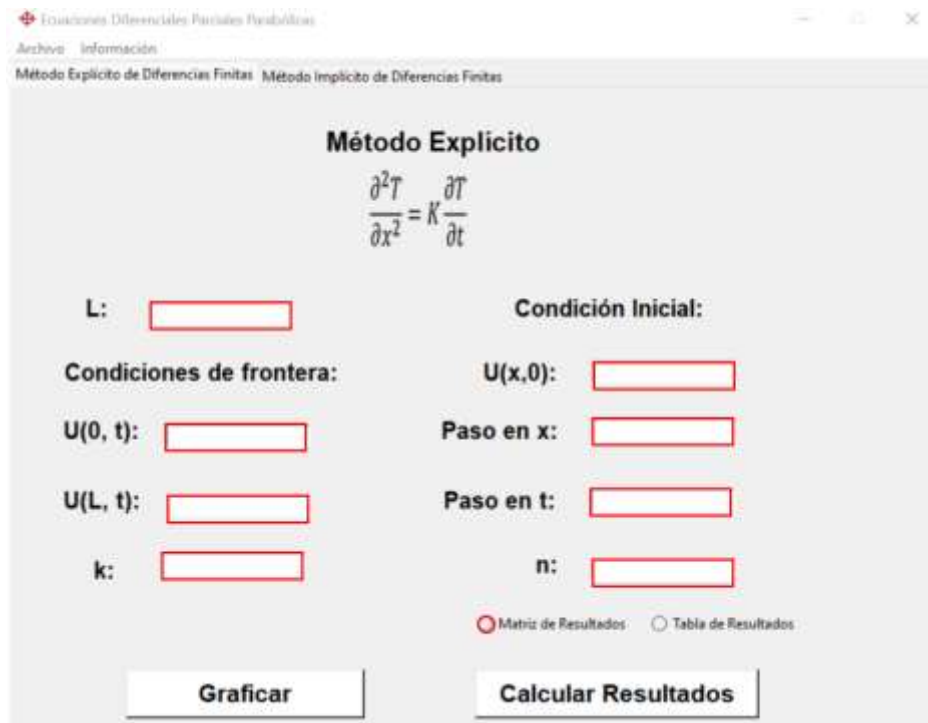
ANEXOS

Manual de Usuario

1. Dar clic en el botón “Iniciar” para poder acceder a la calculadora de Ecuaciones Diferenciales Parciales Parabólicas.



2. Ingresar valores en los campos seleccionados:



3. En el método explícito, si se ingresan valores que no cumplan con la condición de estabilidad, reingresar los valores correctamente.
4. Clicar sobre graficar para obtener la gráfica resultante
5. Cerrar la gráfica para poder obtener la tabla de resultados. Una vez abierta la tabla de resultados, se podrá abrir la gráfica sin ningún problema.
6. Archivo>Salir, para cerrar todas las ventanas de la interfaz.

Código de la solución tecnológica propuesta.

```
from tkinter import *
import tkinter
from tkinter import ttk
from turtle import bgcolor, width
from matplotlib import image
from matplotlib.pyplot import show, text
from numpy import imag
import pylab as pl
import tkinter.messagebox
from PIL import Image, ImageTk
import numpy as np
import matplotlib.pyplot as plt
import io
import contextlib

#PRIMERA VENTANA
def funcion():
    ve1.state(newstate = "normal")

root = Tk()
root.state(newstate="normal")
root.iconbitmap("logo.ico")
root.geometry("960x540+200+50")
root.resizable(width=False, height=False)
root.title("Interfaz de resolución de ecuaciones diferenciales parciales
parabólicas")
fondo = PhotoImage(file= "pag_inicial.png")
fondo1 = Label(root, image=fondo).place(x=0, y=0, relwidth=1, relheight=1)
```

```

abrirventana = Button(root, text="INICIAR", cursor="hand2", bg="#3a2aa6",
width=12, relief="flat", font=("Anonymous Pro", 12, "bold"),
command=funcion)
abrirventana.pack(pady=79)

#VENTANA RESOLVER.
ve1 = Toplevel()
ve1.state(newstate = "withdraw")
ve1.title("Ecuaciones Diferenciales Parciales Parabólicas")
ve1.iconbitmap("logo.ico")
#Tamaño de la pantalla
ve1.geometry("800x600+200+50")
ve1.resizable(width=False, height=False)

#INCLUIAMOS PANEL PARA LAS PESTAÑAS.
nb = ttk.Notebook(ve1)
nb.pack(fill='both', expand='yes')

#CREAMOS PESTAÑAS
p1 = ttk.Frame(nb)
p2 = ttk.Frame(nb)

def implicito():
    b_info = entry12.get()
    ta_info = entry13.get()
    tb_info = entry14.get()
    k_info = entry19.get()
    to_info = entry15.get()
    dx_info = entry16.get()
    dt_info = entry17.get()
    n_info = entry18.get()

    n=int(n_info) # Número de niveles en t
    Ta=int(ta_info) ; Tb=int(tb_info) # Condiciones en los
bordes
    To=int(to_info) # Condición en el inicio
    dx=float(dx_info); dt=float(dt_info) # incrementos
    b=float(b_info) #Posicion final
    k=float(k_info)

    #Particion de la posicion
    xi = np.arange(0, b+dx, dx)

```



```

m=len(xi)
ultimox = m-1

#b/dx==int
var = (str(b/dx))
var = var.split(".")
num_int = var[1]
if num_int != '0':
    tkinter.messagebox.showwarning('Atención','Los incrementos en x no
coinciden con la longitud de la barra.')
    return(p1)

#matriz de inicializacion de ceros
u = np.zeros(shape=(m,n), dtype=float)

j= 0
ultimot = n-1
#matriz de frontera Ta
u[0,j]=Ta
#matriz condicion inicial
u[1:ultimox, j]=To
#matriz condicion Tb
u[ultimox, j]=Tb

#Lambda
lamb= dt/(k*dx**2)
P = lamb
Q = -1-2*lamb
R = lamb

#Bucle para sistema de ecuaciones
j = 1
while not(j>=n):
    u[0,j] = Ta
    u[m-1, j] = Tb
    #Matriz de ecuaciones
    tamaño = m-2
    #Inicializar la matriz A
    A = np.zeros(shape=(tamaño, tamaño), dtype=float)
    #Inicializar la matriz B
    B = np.zeros(tamaño, dtype=float)
    for f in range(0, tamaño, 1):
        #Sistema Lineal
        if(f>0):
            A[f, f-1]=P

```

```

        A[f,f]=Q
        if(f<(tamano-1)):
            A[f,f+1]=R
            B[f]= -u[f+1, j-1]
        B[0]=B[0]-P*u[0,j]
        B[tamano-1]=B[tamano-1]-R*u[m-1, j]

#Resuelve sistema de ecuaciones
C=np.linalg.solve(A, B)

#copia resultados a u[i,j]
for f in range(0, tamano, 1):
    u[f+1, j]=C[f]

#Proximas iteraciones
j=j+1

print('Tabla de resultados')
np.set_printoptions(precision=2)
global result
result = u
global point_y
global point_x
point_y = []
point_x = []

salto = int(n/10)
if (salto == 0):
    salto = 1
for j in range(0, n, salto):
    vector = u[:, j]
    point_y.append(vector)
    point_x.append(xi)
    plt.plot(xi,vector)
    plt.plot(xi,vector, '.m')

plt.xlabel('x(Distancia)')
plt.ylabel('T-°C(Temperatura)')
plt.title('Curva de distribución térmica: Método Implícito')
plt.show()

def embed():
    if seleccion2.get()==1:

```

```

toplevel = tkinter.Toplevel()
toplevel.title('Matriz de Resultados: Método Implícito')
text = tkinter.Text(toplevel, font=("Helvetica", 15))
text.pack(fill=tkinter.BOTH, expand=True)
toplevel.iconbitmap("logo.ico")

file = io.StringIO()
with contextlib.redirect_stdout(file):
    print(result)
    text.insert(tkinter.END, file.getvalue())

if seleccion2.get()==2:
    res = tkinter.Toplevel()
    res.title('Tabla de resultados')
    res.geometry("800x300+400+300")
    res.resizable(width=False, height=False)
    text = tkinter.Text(res, font=("Helvetica", 20))
    text.pack(fill=tkinter.BOTH, expand=True)
    res.iconbitmap("logo.ico")
    labelembed = Label(res, font=('Helvetica', 12), text="Para navegar
entre los resultados, use las flechas de su teclado.", justify=RIGHT)
    labelembed.place(x=0, y=240)

    file = io.StringIO()
    with contextlib.redirect_stdout(file):
        arbol = ttk.Treeview(res, columns=("y", "x"))
        #arbol.insert("", END, text="Principio", values=(point_y[-
1], "15"))

        # arbol.insert("", END, text="Papas", values=("12", "7"))
        #arbol.heading("#0", text="Nombre")
        arbol.heading("y", text="T-°C")
        arbol.heading("x", text="x")
        arbol.place(x=-190, y=10)
        for tupla in zip(point_y[-1], point_x[-1]):

arbol.insert("", END, text="linea", values=(np.around(tupla[0], decimals=2,
out=None), np.around(tupla[1], decimals=2, out=None)))

        # print(np.around(tupla[0], decimals=2,
out=None), np.around(tupla[1], decimals=2, out=None))
        # for tupla2 in zip(tupla[0], tupla[1]):
        #     print(np.around(tupla2[0], decimals=2,
out=None), np.around(tupla2[1], decimals=2, out=None))
        text.insert(tkinter.END, file.getvalue())

```

```

        # lista = ttk.Treeview(res, columns=('x', 'T'))
        # lista.heading('x', text='x')
        # lista.heading('T', text='T')

def explicito():
    b_info = entry2.get()
    ta_info = entry3.get()
    tb_info = entry4.get()
    k_info = entry9.get()
    to_info = entry5.get()
    dx_info = entry6.get()
    dt_info = entry7.get()
    n_info = entry8.get()

    n=int(n_info)                #Número de niveles en t
    Ta=int(ta_info) ; Tb=int(tb_info) #Condiciones en los
bordes
    To=int(to_info)             #Condición en el inicio
    dx=float(dx_info); dt=float(dt_info) #incrementos
    b=float(b_info)             #Posicion final
    k=float(k_info)

    #Particion de la posicion
    xi = np.arange(0, b+dx, dx)
    m=len(xi)
    ultimox = m-1

    #print('division : ',b/dx)
    #print('tipo : ',type(b/dx))
    #b/dx==int
    var = (str(b/dx))
    var = var.split(".")
    num_int = var[1]
    if num_int != '0':
        tkinter.messagebox.showwarning('Atención','Los incrementos en x no
coinciden con la longitud de la barra.')
        return(p1)

    #Matriz de inicializacion de ceros
    u = np.zeros(shape=(m,n), dtype=float)

```

```

#valores iniciales de u, matriz de fronteras e inicial
j=0
ultimot = n-1
u[0, :] = Ta
u[1:ultimox, j] = To
u[ultimox, :] = Tb

#Lambda
#(<1/2) condicion de estabilidad
lamb = dt / (k * dx ** 2)
P = lamb
Q = 1 - 2 * lamb
R = lamb

#Calcula U para cada tiempo +dt
j = 0
while not(j >= ultimot):
    for i in range(1, ultimox, 1):
        u[i, j+1] = P * u[i-1, j] + Q * u[i, j] + R * u[i+1, j]
    j = j+1

if lamb >= 1/2:
    response = tkinter.messagebox.askyesno(message="Debido a la
condición de estabilidad, su gráfica resultante no será una parábola.
¿Desea continuar?", title='Atención')
    if response == False:
        tkinter.messagebox.showerror("Atención", "Vuelva a ingresar
sus datos.")
        return(p1)

print('Tabla de Resultados')
np.set_printoptions(precision=2)
global result1
result1 = u
global point_y
global point_x
point_y = []
point_x = []

salto = int(n/10)
if (salto == 0):
    salto = 1
for j in range(0, n, salto):
    vector = u[:, j]

```

```

point_y.append(vector)
point_x.append(xi)
    # print('vector',vector)
    # print('xi',xi)
plt.plot(xi, vector) #Curvas a cada 10 niveles de tiempo
plt.plot(xi, vector, '.r')
    # print(xi, vector, '.r')
    #print(point_x)
    #print(point_y)
plt.xlabel('x(distancia)')
plt.ylabel('T-°C(Temperatura)')
plt.title('Curva de distribucion térmica: Método Explícito')
plt.show()

def embed1():
    if seleccion1.get()==1:
        toplevel = tkinter.Toplevel()
        toplevel.title('Tabla de Resultados: Método Explícito')
        text = tkinter.Text(toplevel,font=("Helvetica", 15))
        text.pack(fill=tkinter.BOTH, expand=True)
        toplevel.iconbitmap("logo.ico")

        file = io.StringIO()
        with contextlib.redirect_stdout(file):
            print(result1)
            text.insert(tkinter.END, file.getvalue())
    if seleccion1.get()==2:
        res = tkinter.Toplevel()
        res.title('Tabla de resultados')
        res.geometry("800x300+400+300")
        res.resizable(width=False, height=False)
        text = tkinter.Text(res, font=("Helvetica", 20))
        text.pack(fill=tkinter.BOTH, expand=True)
        res.iconbitmap("logo.ico")
        labelembed1 = Label(res, font=('Helvetica', 12), text="Para
navegar entre los resultados, use las flechas de su teclado.",
justify=RIGHT)
        labelembed1.place(x=0, y=240)

        file = io.StringIO()
        with contextlib.redirect_stdout(file):
            arbol = ttk.Treeview(res,columns=("y","x"))
            text = tkinter.Text(res, )

```

```

# arbol.insert("",END,text="Principe",values=(point_y[-
1],"15"))
# arbol.insert("",END,text="Papas",values=("12","7"))
#arbol.heading("#0",text="Ultima curva")
arbol.heading("y",text="T-°C")
arbol.heading("x",text="x")
arbol.place(x=-190,y=10)
for tupla in zip(point_y[-1], point_x[-1]):
    arbol.insert("", END, values=(np.around(tupla[0],
decimals=2, out=None),np.around(tupla[1], decimals=2, out=None))

    # print(np.around(tupla[0], decimals=2,
out=None),np.around(tupla[1], decimals=2, out=None))
    # for tupla2 in zip(tupla[0], tupla[1]):
    #     print(np.around(tupla2[0], decimals=2,
out=None),np.around(tupla2[1], decimals=2, out=None))
    text.insert(tkinter.END, file.getvalue())

# lista = ttk.Treeview(res, columns=('x', 'T'))
# lista.heading('x', text='x')
# lista.heading('T', text='T')

entry2= StringVar()
entry3= StringVar()
entry4= StringVar()
entry5= StringVar()
entry6= StringVar()
entry7= StringVar()
entry8= StringVar()
entry9= StringVar()
entry10= StringVar()
seleccion1=IntVar()

entry12= StringVar()
entry13= StringVar()
entry14= StringVar()
entry15= StringVar()
entry16= StringVar()
entry17= StringVar()
entry18= StringVar()
entry19= StringVar()
entry20= StringVar()
seleccion2=IntVar()

```

```

Entry(p1, textvariable=entry2).place(x=120,y=185)
Entry(p1, textvariable=entry3).place(x=135,y=290)
Entry(p1, textvariable=entry4).place(x=135,y=350)
Entry(p1, textvariable=entry5).place(x=500,y=235)
Entry(p1, textvariable=entry6).place(x=500,y=285)
Entry(p1, textvariable=entry7).place(x=500,y=345)
Entry(p1, textvariable=entry8).place(x=500,y=405)
Entry(p1, textvariable=entry9).place(x=130,y=400)

Entry(p2, textvariable=entry12).place(x=120,y=185)
Entry(p2, textvariable=entry13).place(x=135,y=290)
Entry(p2, textvariable=entry14).place(x=135,y=350)
Entry(p2, textvariable=entry15).place(x=500,y=235)
Entry(p2, textvariable=entry16).place(x=500,y=285)
Entry(p2, textvariable=entry17).place(x=500,y=345)
Entry(p2, textvariable=entry18).place(x=500,y=405)
Entry(p2, textvariable=entry19).place(x=130,y=400)

#METODO EXPLICITO
#Boton Graficar
btnGraficar = Button(p1, text="Graficar", width=15,
                    height=1, bg='white',
                    font=('Helvetica', 16, 'bold'),
                    command=explicito).place(x=100,y=500)
#Check Buttons
c1=ttk.Radiobutton(p1, text='Matriz de Resultados', variable=seleccion1,
width=20, value=1)
c1.pack()
c1.place(x=400, y=450)
c2=ttk.Radiobutton(p1, text='Tabla de Resultados', variable=seleccion1,
value=2, width=20)
c2.pack()
c2.place(x=550, y=450)
#Boton Calcular Resultados
btnResultados = Button(p1, text="Calcular Resultados", width=18,
                    height=1, bg='white',
                    font=('Helvetica', 16, 'bold'),
                    command=embed1).place(x=400,y=500)
#Titulo
labeln = Label(p1, font=('Helvetica', 17, 'bold'), text="Método
Explícito",width=13, justify=RIGHT)
labeln.place(x=270, y=30)

#L

```



```

label2 = Label(p1, font=('Helvetica', 15, 'bold'), text="L:", width=2,
justify=RIGHT)
label2.place(x=60, y=175)
#Condiciones de frontera
label3 = Label(p1, font=('Helvetica', 15, 'bold'), text="Condiciones de
frontera:", width=27, justify=RIGHT)
label3.place(x=0, y=230)
label4 = Label(p1, font=('Helvetica', 15, 'bold'), text="U(0, t):",
width=6, justify=RIGHT)
label4.place(x=40, y=280)
label5 = Label(p1, font=('Helvetica', 15, 'bold'), text="U(L, t):",
width=6, justify=RIGHT)
label5.place(x=40, y=340)
#Condicion Inicial
label6 = Label(p1, font=('Helvetica', 15, 'bold'), text="Condición
Inicial:", width=27, justify=RIGHT)
label6.place(x=350, y=175)
label7 = Label(p1, font=('Helvetica', 15, 'bold'), text="U(x,0):",
width=6, justify=RIGHT)
label7.place(x=400, y=230)
#Paso en x
label8 = Label(p1, font=('Helvetica', 15, 'bold'), text="Paso en x:",
width=8, justify=RIGHT)
label8.place(x=370, y=280)
#Paso en y
label9 = Label(p1, font=('Helvetica', 15, 'bold'), text="Paso en t:",
width=8, justify=RIGHT)
label9.place(x=370, y=340)
#n
label10 = Label(p1, font=('Helvetica', 15, 'bold'), text="n:", width=3,
justify=RIGHT)
label10.place(x=441, y=395)
#k
label11 = Label(p1, font=('Helvetica', 15, 'bold'), text="k:", width=3,
justify=RIGHT)
label11.place(x=60, y=400)

#Ecuacion
image = Image.open("edpp.png")
image = image.resize((700, 90), Image.ANTIALIAS)
img1 = ImageTk.PhotoImage(image)
lbl_img1 = Label(ve1, image= img1)
lbl_img1.place(x=0, y=90)

```

```

#METODO IMPLICITO
#Boton Graficar
btnGraficar = Button(p2, text="Graficar", width=15,
                    height=1, bg='white',
                    font=('Helvetica', 16, 'bold'),
                    command=implicito).place(x=100,y=500)
#Check Buttons
c3=ttk.Radiobutton(p2, text='Matriz de Resultados', variable=seleccion2,
width=20, value=1)
c3.pack()
c3.place(x=400, y=450)
c3=ttk.Radiobutton(p2, text='Tabla de Resultados', variable=seleccion2,
value=2, width= 20)
c3.pack()
c3.place(x=550, y=450)
#Boton Carcular Resultados
btnResultados = Button(p2, text="Calcular Resultados", width=18,
                    height=1, bg='white',
                    font=('Helvetica', 16, 'bold'),
                    command=embed).place(x=400,y=500)
#Titulo
label13 = Label(p2, font=('Helvetica', 17, 'bold'), text="Método
Implícito",width=13, justify=RIGHT)
label13.place(x=270, y=30)
#L
label15 = Label(p2, font=('Helvetica', 15, 'bold'), text="L:", width=2,
justify=RIGHT)
label15.place(x=60, y=175)
#Condiciones de frontera
label16 = Label(p2, font=('Helvetica', 15, 'bold'), text="Condiciones de
frontera:", width=27, justify=RIGHT)
label16.place(x=0, y=230)
label17 = Label(p2, font=('Helvetica', 15, 'bold'), text="U(0, t):",
width=6, justify=RIGHT)
label17.place(x=40, y=280)
label18 = Label(p2, font=('Helvetica', 15, 'bold'), text="U(L, t):",
width=6, justify=RIGHT)
label18.place(x=40, y=340)
#Condicion Inicial
label19 = Label(p2, font=('Helvetica', 15, 'bold'), text="Condición
Inicial:", width=27, justify=RIGHT)
label19.place(x=350, y=180)

```

```

label20 = Label(p2, font=('Helvetica', 15, 'bold'), text="U(x,0):",
width=6, justify=RIGHT)
label20.place(x=400, y=230)
#Paso en x
label21 = Label(p2, font=('Helvetica', 15, 'bold'), text="Paso en x:",
width=8, justify=RIGHT)
label21.place(x=370, y=280)
#Paso en y
label22 = Label(p2, font=('Helvetica', 15, 'bold'), text="Paso en t:",
width=8, justify=RIGHT)
label22.place(x=370, y=340)
#n
label23 = Label(p2, font=('Helvetica', 15, 'bold'), text="n:", width=3,
justify=RIGHT)
label23.place(x=441, y=400)
#K
label24 = Label(p2, font=('Helvetica', 15, 'bold'), text="K:", width=3,
justify=RIGHT)
label24.place(x=60, y=400)

menubar = Menu(ve1)

#Menu Bar SALIR
def iExit():
    iExit = tkinter.messagebox.askyesno("Exit",
                                        "¿Deseas salir?")

    if iExit > 0:
        ve1.destroy()
        root.destroy()

    return

#Menu Bar INFORMACION
def info():
    info_ven = Toplevel()
    info_ven.title("Acerca de las variables")
    info_ven.geometry('850x200+300+200')
    info_ven.iconbitmap("logo.ico")
    label_xf = Label(info_ven, font=('Helvetica', 10, 'bold'), text="L:
Longitud del cuerpo geométrico", width=100, justify=RIGHT)
    label_xf.place(x=20, y=10)
    label_f1 = Label(info_ven, font=('Helvetica', 10, 'bold'),
text="U(L,t): Condición de frontera en la posición inicial. Se permite
ingresar Temperatuara negativa.", width=100, justify=RIGHT)

```

```

    label_f1.place(x=20, y=30)
    label_f2 = Label(info_ven, font=('Helvetica', 10, 'bold'),
text="U(0,t): Condición de frontera en la posición final. Se permite
ingresar Temperatuara negativa.", width=100, justify=RIGHT)
    label_f2.place(x=20, y=50)
    label_k = Label(info_ven, font=('Helvetica', 10, 'bold'), text="k:
Constante que depende del material", width=100, justify=RIGHT)
    label_k.place(x=20, y=70)
    label_i = Label(info_ven, font=('Helvetica', 10, 'bold'),
text="U(x,0): Condición inicial cuando la temperatura es 0.", width=100,
justify=RIGHT)
    label_i.place(x=20, y=90)
    label_px = Label(info_ven, font=('Helvetica', 10, 'bold'), text="Paso
en x: Número de pasos en x", width=100, justify=RIGHT)
    label_px.place(x=20, y=110)
    label_py = Label(info_ven, font=('Helvetica', 10, 'bold'), text="Paso
en t: Número de pasos en t", width=100, justify=RIGHT)
    label_py.place(x=20, y=130)
    label_n = Label(info_ven, font=('Helvetica', 10, 'bold'), text="n:
iteraciones en el tiempo. La gráfica muestra (n/10) iteraciones",
width=100, justify=RIGHT)
    label_n.place(x=20, y=150)

#AGREGAMOS PESTAÑAS CREADAS
nb.add(p1,text='Método Explícito de Diferencias Finitas')
nb.add(p2,text='Método Implícito de Diferencias Finitas')

# MenuBar
filemenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label='Archivo', menu=filemenu)
filemenu.add_command(label="Salir", command=iExit)

infomenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label='Información', menu=infomenu)
infomenu.add_command(label="Acerca de las variables", command=info)

ve1.config(menu=menubar)

root.mainloop()

```



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT

Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Katherine del Rocio Encalada Jimenez**, con C.C: # **0930486394** autor/a del trabajo de integración curricular: **Interfaz de resolución de las ecuaciones diferenciales (tipo parabólicas)** previo a la obtención del título de **Ingeniera en Ciencias de la Computación** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **9 de marzo de 2022**

f. _____

Nombre: **Encalada Jiménez Katherine del Rocio**

C.C: **0930486394**

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN			
TEMA Y SUBTEMA:	Interfaz de resolución de las ecuaciones diferenciales (tipo parabólicas)		
AUTOR(ES)	Encalada Jiménez, Katherine del Rocío		
REVISOR(ES)/TUTOR(ES)	Céleri Mujica, Colón Mario		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	Ingeniería en Ciencias de la Computación		
TITULO OBTENIDO:	Ingeniera en Ciencias de la Computación		
FECHA DE PUBLICACIÓN:	9 de marzo de 2022	No. DE PÁGINAS:	70
ÁREAS TEMÁTICAS:	Ecuaciones Diferenciales Parciales, Ecuaciones Diferenciales Parciales Parabólicas, Interfaz gráfica.		
PALABRAS CLAVES/KEYWORDS:	Ecuaciones Diferenciales Parciales, EDP Parabólicas, Python, Interfaz, Calculadora de EDP Parabólicas.		
RESUMEN/ABSTRACT:			
<p>En la actualidad existen una gran cantidad de alternativas de interfaces que permiten resolver ecuaciones diferenciales; sin embargo, no existe alguna que cumpla con los requisitos propuestos o que resuelva el tipo de ecuaciones presentados en este proyecto. El objetivo de este trabajo de titulación fue desarrollar una interfaz que resuelva y grafique ecuaciones diferenciales parciales de tipo parabólicas, los requisitos principales de esta interfaz son que sea gratis, intuitiva y que no necesite de una conexión a Internet para poder presentar los resultados. La interfaz fue desarrollada para el uso de los estudiantes de la carrera de Computación en la Universidad Católica de Santiago de Guayaquil. Para el desarrollo de la interfaz se implementó el lenguaje de programación Python junto a sus librerías Tkinter, SymPy y Matplotlib para el front end, la resolución de los cálculos numéricos y la representación de gráficos, respectivamente.</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: 0982710991	E-mail: katherine.encalada@cu.ucsg.edu.ec	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):	Nombre: Toala Quimí, Edison Jose		
	Teléfono: +593-990976776		
	E-mail: edison.toala@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			