



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE MATEMÁTICAS



Ejecutable de Interfaz Gráfica de Usuario en App Designer
de Matlab para la Integración Numérica

TESIS

Para optar el título profesional de
Licenciada en Matemáticas

Autoras:

Bach. Mat. Sandoval Tarrillo Xintia Haiduk

Bach. Mat. Lazo Enriquez Lina Elizabeth

Asesor:

Dr. Cuti Gutiérrez Alcides Raúl

LAMBAYEQUE – PERÚ

2025

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE MATEMÁTICAS

Los firmantes, por la presente certifican que han leído y recomiendan a la Facultad de Ciencias Físicas y Matemáticas la aceptación de la tesis titulada **Ejecutable de Interfaz Gráfica de Usuario en App Designer de Matlab para la Integración Numérica**, presentada por las bachilleres en matemáticas, Sandoval Tarrillo Xintia Haiduk y Lazo Enriquez Lina Elizabeth, en el cumplimiento parcial de los requisitos necesarios para la obtención del título profesional de licenciadas en matemáticas.



Dr. Coronado Juárez William Wilmer
Presidente Jurado de Tesis



Dr. Arriaga Delgado Walter
Secretario Jurado de Tesis



Dr. Gonzáles Herrera Mardo Victor
Vocal Jurado de Tesis

Fecha de Defensa: 11 de julio de 2025

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE MATEMÁTICAS

Ejecutable de Interfaz Gráfica de Usuario en App Designer
de Matlab para la Integración Numérica



Bach. Mat. Sandoval Tarrillo Xintia Haiduk

Autor



Bach. Mat. Lazo Enriquez Lina Elizabeth

Autor



Dr. Cuti Gutiérrez Alcides Raúl

Asesor

Lambayeque – Perú

Junio-2025



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DECANATO

Ciudad Universitaria – Lambayeque

LICENCIADA - RESOLUCIÓN DEL CONSEJO DIRECTIVO N° 015 -2023-SUNEDU / CD



0156



ACTA DE SUSTENTACIÓN N° 40.-2025.-D/FACFyM

Siendo las 10:00 am del día 11 de julio del 2025, se reunieron los miembros del jurado evaluador de la Tesis titulada:

Ejecutable de Interfaz Gráfica de Usuario en App
Designer de Matlab para Integración Numérica

Designados por Resolución N° 136 - 2022 D/FACFyM de fecha 04 de febrero de 2022

Con la finalidad de evaluar y calificar la sustentación de la tesis antes mencionada, conformada por los siguientes docentes:

<u>Dr. William Wilmer Coronado Juárez</u>	Presidente
<u>Dr. Walter Arriaga Delgado</u>	Secretario
<u>Dr. Sr. Mardo Víctor González Herrera</u>	Vocal

La tesis fue asesorada por (el) (la) Alcides Paul Cuti Gutiérrez, nombrado por Resolución N° 136 - 2022 D/FACFyM de fecha 04 de febrero de 2022

El Acto de Sustentación fue autorizado por Resolución N° 600 - 2025 D/FACFyM de fecha 01 de julio 2025 y 10 de julio 2025 respectivamente

La Tesis fue presentada y sustentada por (el) (los) Bachiller (es): Sandra Val Tarrillo Xiména Maiduq y Lago Enriquez Lina Elizabeth y tuvo una duración de 25 minutos.

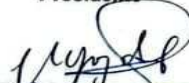
Después de la sustentación, y absueltas las preguntas y observaciones de los miembros del jurado se procedió a la calificación respectiva, otorgándole el Calificativo de dieciséis (16) en la escala vigesimal, mención (Buena).

Por lo que queda(n) apto(s) para obtener el Título Profesional de Licenciado en Matemática, de acuerdo con la Ley Universitaria 30220 y la normatividad vigente de la Facultad de Ciencias Físicas y Matemáticas y la Universidad Nacional Pedro Ruiz Gallo.

Siendo las 10:45 am se dio por concluido el presente acto académico, dándose conformidad al presente acto con la firma de los miembros del jurado.


 Dr. Lic. William Wilmer Coronado Juárez
 Presidente


 Dr. Lic. Walter Arriaga Delgado
 Secretario


 Dr. Lic. Mardo Víctor González Herrera
 Vocal


 Dr. Lic. Alcides Paul Cuti Gutiérrez
 Asesor

CONSTANCIA DE VERIFICACIÓN DE ORIGINALIDAD

Yo, **Dr. Cuti Gutiérrez Alcides Raúl**, usuario revisor de la tesis titulada: **Ejecutable de Interfaz Gráfica de Usuario en App Designer de Matlab para la Integración Numérica**, cuyas autoras son: **Bach. Sandoval Tarrillo Xintia Haiduk, identificada con D.N.I. N°. 72741895** y **Lazo Enriquez Lina Elizabeth, identificada con D.N.I. N°. 71997465**, ha arrojado un porcentaje de similitud de 15%, verificable en el Resumen de Reporte automatizado de similitudes que se acompaña.

El suscrito analizó dicho reporte y concluyó que cada una de las coincidencias detectadas dentro del porcentaje de similitud permitido no constituyen plagio y que el documento cumple con la integridad científica y con las normas para el uso de citas y referencias establecidas en los protocolos respectivos.

Se cumple con adjuntar el Recibo Digital a efectos de la trazabilidad respectiva del proceso.

Lambayeque, 28 de junio de 2025



Dr. Cuti Gutiérrez Alcides Raúl
DNI: 32802095
Asesor

Ejecutable de Interfaz Gráfica de Usuario en App Designer de Matlab para la Integración Numérica

INFORME DE ORIGINALIDAD

15%

INDICE DE SIMILITUD

13%

FUENTES DE INTERNET


6%

PUBLICACIONES

10%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad Nacional Pedro Ruiz Gallo Trabajo del estudiante	4%
2	gredos.usal.es Fuente de Internet	2%
3	repositorio.unprg.edu.pe Fuente de Internet	1%
4	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Trabajo del estudiante	1%
5	stackoverflow.com Fuente de Internet	1%
6	dokumen.pub Fuente de Internet	1%
	 Dr. Cuti Gutiérrez Alcides Raúl DNI: 32802095 Departamento de Matemática Asesor	1%
7	dspace.unitru.edu.pe Fuente de Internet	1%
8	www.mathworks.com Fuente de Internet	<1%

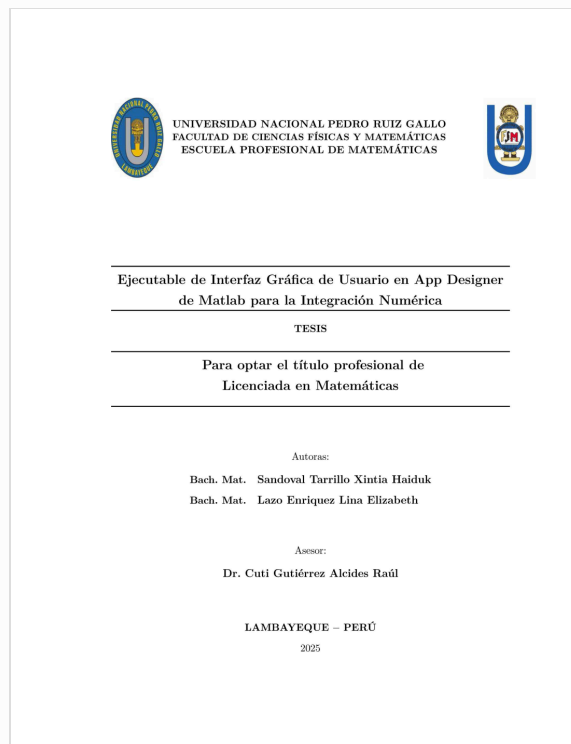



Recibo digital

Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: Sandoval Tarrillo Xintia Haiduk, Lazo Enriquez Lina Elizabeth
Título del ejercicio: Quick Submit
Título de la entrega: Ejecutable de Interfaz Gráfica de Usuario en App Designer de ...
Nombre del archivo: TESIS_XINTIA-LINA_2025.pdf
Tamaño del archivo: 5.78M
Total páginas: 153
Total de palabras: 20,966
Total de caracteres: 110,883
Fecha de entrega: 09-dic-2025 01:34p. m. (UTC-0500)
Identificador de la entrega: 2841452786




Dr. Cuti Gutiérrez Alcides Raúl
DNI: 32802095
Departamento de Matemática
Asesor

Agradecimientos

A Dios, por darme sabiduría e inteligencia para culminar con éxito esta etapa de mi vida.

A mi madre y hermanos por su amor, fortaleza y apoyo incondicional.

A nuestro asesor por impartir con nosotras sus conocimientos y guiarnos hasta el día de hoy.

Xintia

El principal agradecimiento a Dios quien a guiado y a fortalecido mi camino para seguir adelante , a mis padres por el apoyo incondicional, a mi familia por ser mi soporte y mi empuje a seguir adelante.

Lina

Dedicatoria

A mi abuela, quien hoy me acompaña espiritualmente, demostrándome una vez más su fortaleza ante la vida.

A mi madre, mi pilar de vida, motivo y orgullo de ser quien soy.

A mis hermanos, gracias por confiar en mí.

Xintia

A mis padres por su amor, trabajo y sacrificio en todos estos años.

A mi esposo e hijas quienes me han ayudado a nunca rendirme en este largo camino.

A nuestros docentes de Escuela por haber compartido sus conocimientos a lo largo de la preparación de nuestra profesión de manera especial al asesor de nuestro proyecto de investigación quien a guiado con su paciencia y rectitud como docente.

Lina

Índice

Agradecimientos	I
Dedicatoria	II
Índice	III
Índice de tablas	VI
Índice de figuras	VII
Resumen	IX
Abstract	X
Introducción	XI
I. DISEÑO TEÓRICO	1
1.1 Antecedentes de la Investigación	1
1.1.1 A nivel internacional	1
1.1.2 A nivel nacional	6
1.1.3 A nivel local	6
1.2 Base Teórica	7
1.2.1 Serie de Taylor	7
1.2.2 Método del Trapecio	11
1.2.3 Error de truncamiento del método del trapecio	15
1.2.4 Método de Simpson 1/3	17
1.2.5 Método de Simpson 1/3 compuesta	19
1.2.6 Método de Simpson 3/8	22
1.2.7 Método de Simpson 3/8 compuesta	26
1.2.8 Objetos de Diseño en App Designer de Matlab R2021a	30
	III

1.3 Bases Conceptuales	33
1.3.1 Integración Numérica	33
1.3.2 Método del Trapecio	33
1.3.3 Regla de Simpson 1/3	33
1.3.4 Regla de Simpson 3/8	34
1.3.5 Interfaces gráficas de usuario (GUI) en App Designer de Matlab R2021a	34
1.3.6 Matlab R2021a	35
II. DISEÑO METODOLÓGICO	36
2.1 Tipo de Investigación	36
2.2 Método de Investigación	36
2.3 Diseño de Contrastación	36
2.4 Población, Muestra y Muestreo	37
2.5 Técnicas, Instrumentos, Equipos y Materiales de Recolección de Datos	37
2.6 Procesamiento y Análisis de Datos	37
III. RESULTADOS	38
3.1 Diseño de la Interfaz Gráfica de Usuario (GUI) en App Designer.	38
3.1.1 Diseño de la GUI del Método del Trapecio.	38
3.1.2 Diseño de la GUI del Método de Simpson 1/3	50
3.1.3 Diseño de la GUI del Método de Simpson 3/8	52
3.1.3 Diseño de la Portada	54
3.1.4 Convertir a Ejecutable	55
3.2 Aplicación 1	58
3.3 Aplicación 2	61
3.4 Aplicación 3	64
IV. DISCUSIÓN DE RESULTADOS	67
V. CONCLUSIONES	70

RECOMENDACIONES	72
Referencias Bibliográficas	73
Anexos	75

Índice de tablas

1.	Valores de $f(x_i) = x_i^3 e^{x_i^2}$	21
2.	Tabulando la función $f(x) = x^3 e^{x^2}$ para $h = 0.1667$ en el intervalo $[0, 2]$. .	29
3.	Comparación de resultado de $\int_0^2 x^3 e^{x^2} dx$	30
4.	Comparando los resultados de la aplicación 1, para $n=12$	60
5.	Comparando los resultados de la aplicación 2, para $n=12$	63
6.	Comparando los resultados de la aplicación 3, para $n=48$	66

Índice de figuras

1.	<i>Representación geométrica de la regla del trapecio.</i>	12
2.	<i>Regla de Simpson $\frac{1}{3}$.</i>	17
3.	<i>Regla de Simpson $\frac{3}{8}$.</i>	22
4.	<i>Ingreso en App Designer.</i>	38
5.	<i>Se selecciona una ventana en blanco.</i>	38
6.	<i>Interfaz gráfica de usuario.</i>	39
7.	<i>Se coloca un Label para el título.</i>	39
8.	<i>Ingresar un panel 1.</i>	40
9.	<i>Se coloca el nombre de “Ingrese la funcion $f(x)=:$”.</i>	40
10.	<i>Edit Text Fiel, para ingresar $f(x)$.</i>	41
11.	<i>Se coloca el nombre de “Ingrese el Intervalo $[a , b] =$”.</i>	41
12.	<i>Edit Text Fiel, para ingresar $[a,b]$.</i>	42
13.	<i>Se coloca un Label para “Nº de subintervalos $n=$”.</i>	42
14.	<i>Edit Text Fiel, para ingresar el número de subintervalos.</i>	43
15.	<i>Ingresar un botón para ingresar datos predefinidos.</i>	43
16.	<i>Ingresar un panel 2.</i>	44
17.	<i>Ingresar 3 botones.</i>	44
18.	<i>Se ingresa una tabla para mostrar resultados.</i>	45
19.	<i>Muestra el resultado.</i>	45
20.	<i>Se muestra la gráfica de la función.</i>	46
21.	<i>Guardar la GUI en App Designer.</i>	46
22.	<i>Colocar el nombre de la GUI en Designer.</i>	47
23.	<i>Diseño de la GUI en App Designer.</i>	47
24.	<i>Diseño final de la GUI en App Designer.</i>	48
25.	<i>Diseño final del método de Simpson compuesto $1/3$.</i>	50
26.	<i>Diseño final del método de Simpson compuesto $3/8$.</i>	52
27.	<i>Portada.</i>	54

28.	<i>Deploytool.</i>	56
29.	<i>Application compiler.</i>	56
30.	<i>Cargar datos.</i>	57
31.	<i>Activación de los 4 chek.</i>	57
32.	<i>Resultado de la aplicación 1 método del Trapecio, para n=12.</i>	58
33.	<i>Resultado de la aplicación 1 método de Simpson 1/3, para n=12.</i>	59
34.	<i>Resultado de la aplicación 1 método de Simpson 3/8, para n=12.</i>	59
35.	<i>Resultado de la aplicación 2 método del Trapecio, para n=12.</i>	61
36.	<i>Resultado de la aplicación 2 método de Simpson 1/3, para n=12.</i>	62
37.	<i>Resultado de la aplicación 2 método de Simpson 3/8, para n=12.</i>	62
38.	<i>Resultado de la aplicación 3 método del Trapecio, para n=48.</i>	64
39.	<i>Resultado de la aplicación 2 método de Simpson 1/3, para n=48.</i>	65
40.	<i>Resultado de la aplicación 2 método de Simpson 3/8, para n=48.</i>	65

Resumen

El objetivo principal de esta investigación fue desarrollar un ejecutable de interfaz gráfica de usuario en App Designer de Matlab R2021a para la integración numérica mediante los métodos del trapecio, Simpson compuesto $1/3$ y Simpson compuesto $3/8$. Para lograrlo, se diseñó un entorno interactivo que permitió al usuario ingresar la función, los límites del intervalo y el número de subintervalos, generando automáticamente los resultados numéricos y su representación gráfica. Además, se programaron validaciones internas que aseguraron el uso correcto de cada método: se mostró un mensaje de advertencia si el valor de n no era par en Simpson compuesto $1/3$, o no múltiplo de 3 en Simpson compuesto $3/8$. La metodología consistió en implementar cada método en código Matlab R2021a, construir la interfaz gráfica en App Designer y convertir el proyecto en un ejecutable independiente, eliminando la necesidad de contar con el software instalado. Se aplicaron los tres métodos a distintas funciones, evaluando la precisión de los resultados comparándolos con soluciones analíticas. Los resultados demostraron que el método de Simpson compuesto $1/3$ fue el más preciso en dos de las tres aplicaciones evaluadas, especialmente cuando se usaron más subintervalos. Sin embargo, el método del trapecio fue el más exacto en una de las funciones racionales con solo 12 particiones. En conclusión, el aplicativo permitió resolver integrales definidas de forma visual, precisa y práctica, demostrando que el método de Simpson compuesto $1/3$ es la opción más recomendable cuando se requiere mayor número de subintervalos. Su independencia del entorno Matlab R2021a lo convirtió en una herramienta didáctica accesible, útil para estudiantes y profesionales que requieren aplicar métodos de integración numérica sin complicaciones técnicas.

Palabras clave: Integración numérica, Interfaz gráfica, App Designer, Ejecutable, Matlab.

Abstract

The main objective of this research was to develop an executable graphical user interface in Matlab R2021a App Designer for numerical integration using the trapezoidal rule, composite Simpson's $1/3$ rule, and composite Simpson's $3/8$ rule. To achieve this, an interactive environment was designed, allowing the user to input the function, the integration limits, and the number of subintervals, automatically generating both numerical results and their graphical representation. Additionally, internal validations were programmed to ensure the correct use of each method: a warning message appeared if the value of n was not even in Simpson's $1/3$ rule or not a multiple of 3 in Simpson's $3/8$ rule. The methodology involved implementing each method in Matlab R2021a code, building the graphical interface in App Designer, and converting the project into an independent executable, removing the need for MATLAB to be installed. The three methods were applied to different functions, and the accuracy of the results was evaluated by comparison with analytical solutions. The results showed that the composite Simpson's $1/3$ method was the most accurate in two of the three applications tested, especially when more subintervals were used. However, the trapezoidal rule was the most precise for one rational function using only 12 partitions. In conclusion, the application enabled defined integrals to be solved visually, accurately, and efficiently, proving that Simpson's $1/3$ method is the most recommended option when a higher number of subintervals is required. Its independence from the Matlab R2021a environment made it an accessible educational tool, useful for students and professionals who need to apply numerical integration methods without technical limitations.

Keywords: Numerical Integration, Graphical User Interface, App Designer, Executable, Matlab

Introducción

En los últimos años, el avance de la tecnología ha llevado a que en diversas áreas del conocimiento se presenten problemas que requieren el uso de la integral definida, especialmente para calcular áreas, volúmenes o resolver situaciones relacionadas con fenómenos físicos y matemáticos. Aunque existen métodos clásicos como la integración por partes, el cambio de variable o la sustitución trigonométrica, hay casos en los que dichas técnicas no permiten encontrar una solución exacta. Ante esta dificultad, se recurre a los métodos numéricos, que ofrecen resultados aproximados con alta precisión.

Dentro de los métodos más conocidos se encuentran el método del trapecio, el método de Simpson compuesto $1/3$ y el método de Simpson compuesto $3/8$. Estos métodos, por su estructura algorítmica, pueden ser implementados fácilmente en herramientas computacionales. En este contexto, la presente investigación tuvo como objetivo desarrollar un ejecutable de interfaz gráfica de usuario en App Designer de Matlab R2021a para aplicar estos tres métodos de integración numérica. El propósito fue facilitar la solución de integrales definidas de manera rápida, visual y sin complicaciones, permitiendo su uso sin necesidad de tener instalado el software Matlab R2021a. De esta forma, el aplicativo está dirigido a estudiantes, docentes e investigadores, promoviendo el aprendizaje autónomo y práctico (Tineo y Torres, 2022).

El problema que motivó esta investigación se formuló en la siguiente pregunta: ¿De qué manera se puede desarrollar un ejecutable de interfaz gráfica de usuario en App Designer de MATLAB para la integración numérica? Para dar respuesta, se planteó como hipótesis que el desarrollo de este ejecutable permitiría resolver métodos numéricos de integración de manera eficiente y accesible.

El trabajo comprendió el estudio teórico de los métodos mencionados, el diseño metodológico del aplicativo, su validación mediante diversas funciones, el análisis de los resultados obtenidos y la discusión de su precisión. Además, se presentaron las conclusiones, recomendaciones y anexos con evidencia del funcionamiento. Con esta estructura, se busca contribuir a la enseñanza y aplicación práctica de los métodos de integración numérica en entornos

educativos y profesionales.

I. DISEÑO TEÓRICO

1.1 Antecedentes de la Investigación

1.1.1 A nivel internacional

Eidous y Ananbeh (2024) su objetivo fue proponer nuevos estimadores no paramétricos para calcular el coeficiente de solapamiento de Weitzman (Δ) entre dos distribuciones continuas, mediante métodos de integración numérica. La metodología consistió en utilizar funciones kernel para estimar las densidades de probabilidad y aplicar las reglas del trapecio y de Simpson para aproximar la integral que define el coeficiente Δ . A diferencia de trabajos anteriores que usaban métodos de momentos, esta investigación optó por una integración directa de la función mínima entre ambas densidades estimadas. Se realizó una simulación utilizando distribuciones Weibull, Lomax, Gumbel y Normal, en diferentes combinaciones y tamaños muestrales, y se compararon los nuevos estimadores con uno previo propuesto por Eidous y Al-Talafhah (2020). Los resultados mostraron que los nuevos estimadores presentaron menores sesgos relativos y errores cuadráticos medios, incluso con muestras pequeñas. En la mayoría de los casos, también ofrecieron una mayor eficiencia. Finalmente, se aplicaron los métodos a un conjunto de datos reales sobre fallas de dos aviones, confirmando el mejor rendimiento de los estimadores propuestos. En conclusión, los métodos desarrollados resultaron ser más precisos y consistentes, y representan una mejora significativa respecto a enfoques anteriores para estimar el coeficiente de solapamiento.

Debasish y Sanjit (2024) su objetivo fue desarrollar una fórmula de cuadratura alternativa a la de Simpson, denominada anti-Simpson, y extenderla para mejorar la evaluación numérica de integrales, incluyendo las de tipo elíptico, en entornos adaptativos. La metodología consistió en construir una fórmula de cuadratura de cuatro puntos basada en la de Simpson 1/3, siguiendo ideas de Laurie, y luego combinarla linealmente con la misma fórmula de Simpson para obtener una versión extendida con mayor precisión. Se realizó un análisis teórico del error asociado y se evaluó el desempeño numérico comparando con reglas clásicas como la de Boole. Los resultados mostraron que la fórmula extendida DS1

superó en precisión a las fórmulas originales en varios ejemplos, especialmente en entornos adaptativos donde se evidenció una reducción significativa del número de pasos y del error. En las conclusiones, se afirmó que la nueva regla DS1 puede considerarse una alternativa viable a otras reglas existentes debido a su facilidad de implementación y mejor comportamiento en integrales complejas. Además, se destacó que esta técnica podría extenderse a otras fórmulas como la anti-Lobatto o anti-Gauss para futuras aplicaciones.

Guo (2023) tuvo como objetivo mejorar el aprendizaje de la integración numérica en estudiantes de ingeniería, aclarando errores comunes y demostrando cómo usar diferentes métodos según cada situación. La metodología consistió en presentar ocho ejemplos prácticos observados en clases y evaluaciones reales en la Universidad Central de Queensland, donde se compararon los métodos del trapecio y de Simpson. El autor explicó cuándo es correcto aplicar cada uno, destacando que muchos estudiantes aplicaban mal el método de Simpson en subdivisiones impares, lo que generaba errores significativos. Los resultados mostraron que el método de Simpson es más preciso si se usa con intervalos pares y funciones suaves, pero el método del trapecio puede superar a Simpson en casos con datos lineales o dispares. También se ilustró cómo validar resultados usando interpolaciones, series de Maclaurin y herramientas computacionales como MATLAB. Se concluyó que los estudiantes deben aprender a elegir el método adecuado según el tipo de problema, y validar sus resultados con otra técnica si no hay solución exacta. El enfoque permitió desarrollar pensamiento estratégico en los estudiantes, una habilidad esencial en problemas de ingeniería del mundo real donde no siempre hay soluciones analíticas disponibles.

Madrid (2023) tuvo como objetivo analizar el comportamiento del orden de convergencia de las reglas de integración numérica del trapecio y de Simpson cuando se aplican a funciones que no cumplen las condiciones clásicas de diferenciabilidad. Para ello, se empleó una metodología teórica y experimental, en la que primero se desarrollaron estimaciones de error para funciones no diferenciables, y luego se validaron mediante simulaciones numéricas. En particular, se consideraron funciones de la forma $f(x) = x^\alpha$ con $0 < \alpha < 2$ y otras más generales expresadas como series de potencias multiplicadas por términos fraccionarios. Los

resultados mostraron que, aunque las funciones no poseen derivadas continuas del orden exigido, las reglas mencionadas aún presentan convergencia, aunque de orden fraccionario, siendo menor o igual al orden clásico. Además, en algunos casos, se mantuvo el mismo orden de convergencia esperado sin requerir la diferenciabilidad tradicional. Se concluyó que es posible aplicar métodos de integración numérica a una clase más amplia de funciones, lo cual amplía el campo de aplicación de estas reglas. Este hallazgo representa un aporte importante al análisis numérico, al demostrar que la suavidad de la función no siempre es una condición indispensable para obtener buenos resultados.

Solano (2023) tuvo como objetivo mejorar el aprendizaje en temas de cálculo, específicamente en integración numérica, utilizando la plataforma Computer Supported Learning (CSL) como herramienta educativa. La metodología consistió en implementar un entorno virtual donde los estudiantes interactuaron con actividades prácticas relacionadas con el método del trapecio, el método de Simpson y otros enfoques para resolver integrales definidas. Participaron estudiantes de primer año de ingeniería, quienes resolvieron ejercicios diseñados para fortalecer su comprensión conceptual y procedimental. Durante el estudio, se analizaron los resultados de aprendizaje de los estudiantes antes y después del uso del entorno CSL. Los datos se recopilaban mediante pruebas escritas, tareas digitales y retroalimentación de los participantes. Los resultados mostraron una mejora significativa en la precisión de las respuestas, el razonamiento matemático y la confianza de los estudiantes al aplicar los métodos de integración numérica. Como conclusión, se evidenció que el uso de herramientas digitales como CSL no solo facilitó la enseñanza de contenidos abstractos, sino que también motivó a los estudiantes a explorar diferentes estrategias para resolver problemas. La experiencia demostró que el apoyo tecnológico puede ser clave para mejorar la comprensión en cursos iniciales de matemáticas en carreras de ingeniería.

Aflook y Al-Rammahi (2021) su objetivo principal fue mejorar la precisión del método de integración numérica de Simpson $3/8$ mediante nuevas fórmulas de aceleración exponencial. Para lograrlo, se desarrollaron dos leyes de aceleración derivadas de series de Maclaurin, aplicadas a los errores del método original. La metodología consistió en dedu-

cir estas leyes matemáticamente y comprobar su eficacia resolviendo integrales definidas con funciones continuas. Luego, se compararon los resultados obtenidos con los métodos tradicionales como Simpson 1/3, Trapezoidal y el mismo Simpson 3/8 sin aceleración. Los resultados mostraron que las nuevas aceleraciones alcanzaron una mayor precisión en menos particiones. Por ejemplo, mientras el método de Simpson 3/8 alcanzó ocho cifras decimales correctas con 30 subintervalos, la nueva aceleración logró nueve cifras con solo 15. En conclusión, las leyes de aceleración exponencial ofrecieron mejoras significativas en eficiencia y exactitud, duplicando en muchos casos la precisión del método original con menos esfuerzo computacional. Estas mejoras son útiles en contextos donde se requiere alta precisión en integración numérica con funciones continuas.

Ahmed et al. (2021) El objetivo general del estudio fue determinar cuál de los métodos de integración numérica basados en la fórmula de Newton Cotes regla del trapecio, Simpson 1/3, Simpson 3/8 y Weddle resulta más preciso en la resolución de distintos problemas de ingeniería, dependiendo del número de subintervalos usados. La metodología se centró en resolver 17 problemas reales mediante integración numérica, utilizando Python 3.8 para las aproximaciones y Maple para obtener los valores exactos. Se evaluaron errores porcentuales según diferentes cantidades de subintervalos, analizando patrones de precisión para cada método. Entre los resultados más relevantes, se encontró que no existe un único método que siempre sea el más preciso; en cambio, la precisión varía periódicamente con los subintervalos. En la mayoría de los casos, Weddle fue el más exacto cuando el número de subintervalos era múltiplo de 6. Las conclusiones indicaron que existe una secuencia periódica en la precisión de los métodos, lo cual es útil para ingenieros al elegir el método más adecuado en función de la exactitud deseada. Además, se comprobó que el método del trapecio rara vez es el más exacto, mientras que los métodos de Simpson y Weddle presentan mejor desempeño en la mayoría de aplicaciones prácticas.

Uddin et al. (2019) su objetivo fue analizar cómo mejorar la enseñanza y comprensión del concepto de integral definida mediante el uso del software GeoGebra, integrando además métodos numéricos como el método del trapecio y el método de Simpson. La intención fue

que los estudiantes no solo entendieran el área bajo la curva de forma visual, sino también aprendieran a calcularla numéricamente en casos donde la solución exacta no era posible. La metodología fue cualitativa, con un enfoque de estudio de caso. Participaron estudiantes de secundaria, quienes realizaron actividades prácticas con GeoGebra. Se les presentaron funciones donde debían explorar gráficamente el área bajo la curva y luego aplicar los métodos numéricos para aproximar el valor de la integral. Se observaron sus procesos, se analizaron sus tareas y se les aplicaron entrevistas cortas. Los resultados demostraron que el uso combinado de representaciones visuales con cálculos numéricos fortaleció la comprensión del concepto. Los estudiantes lograron relacionar el área representada en GeoGebra con los valores aproximados obtenidos mediante los métodos numéricos. En conclusión, se confirmó que GeoGebra, junto con la enseñanza de métodos numéricos, es una estrategia eficaz. Este enfoque permitió a los estudiantes comprender mejor la integral definida y aplicar técnicas de aproximación de forma significativa.

Md. et al. (2019) tuvo como objetivo comparar tres métodos de integración numérica la regla del trapecio, la regla de Simpson $1/3$ y la regla de Simpson $3/8$ aplicados a espacios con datos desiguales, para identificar cuál ofrecía mayor precisión. Para ello, se emplearon funciones específicas con intervalos no uniformes, resolviendo cada una mediante los tres métodos y comparando los resultados obtenidos con las soluciones exactas. La metodología incluyó el uso de fórmulas de cuadratura general adaptadas a intervalos desiguales, aplicando luego los métodos numéricos a tres ejemplos concretos. Los resultados mostraron que la regla de Simpson $1/3$ arrojó errores porcentuales más bajos en los tres casos evaluados, en comparación con los otros métodos. Se concluyó que, dentro del contexto de datos desiguales, la regla de Simpson $1/3$ ofreció la mayor precisión y, por tanto, se consideró la más recomendable entre las técnicas evaluadas. Esta comparación aportó una visión clara sobre cuál método conviene usar en problemas donde los puntos de datos no están espaciados uniformemente.

1.1.2 A nivel nacional

Mallma et al. (2022) su estudio tuvo como propósito calcular el área total de la superficie exterior de la Universidad Nacional Mayor de San Marcos. Para ello, se dividió el terreno en cuatro secciones, cuyos puntos fueron determinados utilizando coordenadas obtenidas de Google Maps. Se aplicó la interpolación de Lagrange para encontrar las ecuaciones de las curvas que representan los bordes de cada sección, y luego se calculó el área mediante métodos de integración numérica: la regla del trapecio, Simpson 1/3 y Simpson 3/8. Los valores obtenidos fueron comparados con el área oficial registrada en documentos topográficos. Los resultados indicaron que la suma de las áreas calculadas fue de aproximadamente $739,084.86 \text{ m}^2$, con un margen de error de solo 0.033 % respecto al área oficial. El método de Simpson (1/3 y 3/8) presentó mayor precisión, coincidiendo casi exactamente con el valor real. Además, se desarrolló una aplicación en C++ para validar los cálculos y comprobar su eficacia. En conclusión, el uso de interpolación polinómica junto con métodos numéricos permitió estimar con alta exactitud superficies complejas. Esta metodología resulta útil para proyectos de ingeniería donde las formas del terreno no son regulares ni fácilmente medibles.

1.1.3 A nivel local

Tineo y Torres (2022) su objetivo fue aplicar métodos de integración numérica apoyados por una interfaz gráfica desarrollada en Matlab, con la finalidad de resolver integrales definidas de forma rápida y accesible. Para ello, se implementaron tres métodos numéricos: el del trapecio, Simpson 1/3 y Simpson 3/8. Se diseñaron interfaces gráficas que permitieron al usuario ingresar los límites de integración, la función deseada y el número de subintervalos, generando automáticamente resultados numéricos, la solución exacta, el error y la gráfica correspondiente. La metodología incluyó el desarrollo de tres aplicaciones prácticas: en dos se compararon los métodos analítico y numérico, y en la tercera se contrastaron los resultados con interpolaciones tipo spline. Los resultados mostraron que el método de Simpson 1/3 fue el más preciso, con un error muy bajo. Finalmente, se concluyó que la interfaz gráfica

elaborada en Matlab facilitó considerablemente el uso de estos métodos, permitiendo a los usuarios resolver integrales definidas de manera visual, interactiva y eficiente, lo cual valida su utilidad como herramienta educativa y computacional.

1.2 Base Teórica

1.2.1 Serie de Taylor

Definición 1. Sea $f(x)$ una función que posee derivadas de todos los órdenes en un punto $x_0 \in \mathbb{R}$. La expresión

$$f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + \cdots$$

se denomina serie de Taylor de la función f centrada en x_0 .

Definición 2. Sea f una función que posee derivadas continuas hasta el orden n en un punto x_0 . El polinomio

$$P_n(x) = \sum_{i=0}^n \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i$$

se denomina polinomio de Taylor de orden n de f alrededor del punto x_0 .

Tineo y Torres (2022) mencionaron el siguiente Teorema

Teorema 1. Sea x_0 un número real, n un entero no negativo y f una función tal que $f^{n+1}(x)$ existe para todo x en algún intervalo abierto I que contenga x_0 . Sea P_n el polinomio de Taylor de orden n para f sobre x_0 . Entonces para todo $x \in I - \{x_0\}$ existe un c entre x_0 y x tal que

$$f(x) = P_n(x) + \frac{f^{(n+1)}(c)}{(n+1)!} (x - x_0)^{n+1} \quad (1)$$

Demostración. Si $x \in I - \{x_0\}$, y para cada $t \in I$ definimos

$$\begin{aligned} F(t) &= f(x) - \sum_{i=0}^n \frac{f^{(i)}(t)}{i!} (x-t)^i \\ &= f(x) - f(t) - \sum_{i=1}^n \frac{f^{(i)}(t)}{i!} (x-t)^i. \end{aligned}$$

Dado que la derivada de orden $n+1$ de la función f existe para todo valor t dentro del intervalo I , se concluye que todas las derivadas desde la primera hasta la n -ésima también son continuas en ese intervalo. Por lo tanto, la función F es continua en todo el tramo cerrado entre x_0 y x , y además se puede derivar dentro del intervalo abierto que los une. Es importante considerar este detalle.

$$F(x) = f(x) - f(x) = 0$$

y

$$F(x_0) = f(x) - \sum_{i=0}^n \frac{f^{(i)}(a)}{i!} (x-x_0)^i = f(x) - P_n(x)$$

Así tenemos que demostrar que

$$F(x_0) = \frac{f^{(n+1)}(c)}{(n+1)!} (x-x_0)^{n+1}$$

para algún c entre x_0 y x .

Usando la regla del producto y la regla de la cadena, obtenemos

$$\begin{aligned} F'(t) &= -f'(t) - \sum_{i=1}^n \left(\frac{f^{(i+1)}(t)}{i!} (x-t)^i + \frac{f^{(i)}(t)}{i!} i(x-t)^{i-1} (-1) \right) \\ &= -f'(t) - \sum_{i=1}^n \left(\frac{f^{(i+1)}(t)}{i!} (x-t)^j - \frac{f^{(i)}(t)}{(i-1)!} (x-t)^{i-1} \right) \end{aligned}$$

Por la propiedad telescópica, se tiene

$$\begin{aligned} &= -f'(t) - \left(\frac{f^{(n+1)}(t)}{n!} (x-t)^n - f'(t) \right) \\ &= -\frac{(x-t)^n}{n!} f^{(n+1)}(t) \end{aligned}$$

Haciendo

$$G(t) = (x-t)^{n+1}$$

para todo $t \in I$. Entonces

$$G'(t) = -(n+1)(x-t)^n$$

Se tiene que $G(x) = 0$ y que $G'(t) \neq 0$ para todo $t \neq x$. Por la fórmula del valor medio de Cauchy, existe un c entre x_0 y x tal que

$$\frac{F'(c)}{G'(c)} = \frac{F(x_0) - F(x)}{G(x_0) - G(x)} = \frac{F(x_0)}{G(x_0)}$$

entonces, se tiene

$$\begin{aligned} F(x_0) &= \frac{F'(c)}{G'(c)} G(x_0) \\ &= \frac{-\frac{(x-c)^n}{n!} f^{(n+1)}(c) (x-x_0)^{n+1}}{-(n+1)(x-c)^n} \\ &= \frac{f^{(n+1)}(c)}{(n+1)!} (x-x_0)^{n+1} \end{aligned}$$

□

Definición 3. Para cada $x \in I - \{x_0\}$ definimos la función R_n llamada resto de Taylor de grado n , como

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} (x - x_0)^{n+1}$$

con c entre x_0 y x .

Una manera práctica de expresar la serie de Taylor es cambiar en la ecuación (1) el valor de x por $x + h$ y el de x_0 por x . Así se obtiene la siguiente forma:

$$f(x + h) = \sum_{i=0}^n \frac{f^{(i)}(x)}{i!} h^i + \frac{f^{(n+1)}(c)}{(n+1)!} h^{n+1} \quad (2)$$

Definición 4. Sea $n \in \mathbb{N}$, $n > 0$. Se dice que una función $F(h)$ es de orden h^n , y se escribe

$$F(h) = O(h^n),$$

si existe una constante $C \in \mathbb{R}$, con $C > 0$, tal que

$$|F(h)| \leq C |h^n|$$

para todo h suficientemente cercano a cero. En otras palabras, $F(h)$ decrece, como máximo, con la misma rapidez que h^n cuando $h \rightarrow 0$.

De la ecuación (2) y la definición 4, se deduce el desarrollo de Taylor en la forma

$$f(x + h) = \sum_{k=0}^n \frac{f^{(k)}(x)}{k!} h^k + O(h^{n+1}), \quad (3)$$

donde el término $O(h^{n+1})$ representa el resto acotado del orden correspondiente.

“

1.2.2 Método del Trapecio

Tineo y Torres (2022) establecen que la integral definida se expresa como

$$I = \int_a^b f(x), dx \quad (4)$$

donde $[a, b]$ es un intervalo acotado.

Una de las estrategias más empleadas para aproximar el valor de una integral definida es la regla del trapecio, la cual consiste en sustituir la función $f(x)$ por segmentos de recta que conectan puntos específicos denominados nodos, x_0, x_1, \dots, x_n . De esta manera, el área bajo la curva se aproxima como la suma de áreas de trapecios formados sobre cada subintervalo.

Para su implementación, el intervalo $[a, b]$ se divide en n subintervalos de igual longitud, determinada por

$$h = \frac{b - a}{n},$$

y los nodos se definen como

$$x_i = a + i h, \quad i = 0, 1, \dots, n.$$

En cada subintervalo $[x_{i-1}, x_i]$, el área aproximada se calcula como

$$A_i = \frac{h}{2} [f(x_{i-1}) + f(x_i)]. \quad (5)$$

La suma de las áreas de los n trapecios conduce a

$$\begin{aligned} T_n &= A_1 + A_2 + \dots + A_n \\ &= \frac{h}{2} [f(x_0) + f(x_1)] + \frac{h}{2} [f(x_1) + f(x_2)] + \dots + \frac{h}{2} [f(x_{n-1}) + f(x_n)]. \end{aligned}$$

Agrupando términos semejantes se obtiene:

$$T_n = \frac{h}{2}f(x_0) + hf(x_1) + \cdots + hf(x_{n-1}) + \frac{h}{2}f(x_n),$$

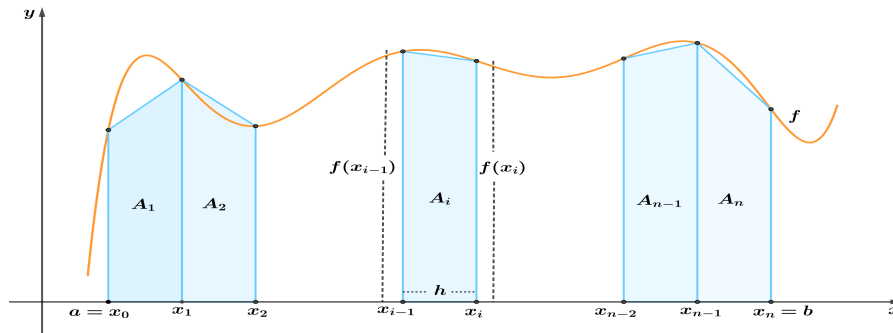
lo que permite expresar la fórmula compuesta de la regla del trapecio como:

$$T_n = \frac{h}{2} [f(x_0) + f(x_n)] + h \sum_{i=1}^{n-1} f(x_i). \quad (6)$$

La ecuación (6) proporciona una estimación de la integral definida I mediante la suma de las áreas de n trapecios, constituyendo así un método de integración numérica ampliamente utilizado por su simplicidad y aplicabilidad (Kharab & Guenther, 2012).

Figura 1

Representación geométrica de la regla del trapecio.



Ejemplo 1. Resolver la siguiente integral

$$\int_0^2 x^3 e^{x^2} dx$$

Solución.

Sea:

$$\int_0^2 x^3 e^{x^2} dx = I$$

Aplicamos el cambio de variable:

$$u = x^2 \Rightarrow du = 2x dx \Rightarrow x dx = \frac{1}{2} du$$

$$x^3 = x \cdot x^2 = x \cdot u$$

Sustituyendo en la integral:

$$I = \int_0^2 x^3 e^{x^2} dx = \int_0^2 x \cdot u \cdot e^u dx$$

Transformamos completamente a la variable u :

$$I = \int_{u=0}^{u=4} u e^u \cdot \frac{1}{2} du = \frac{1}{2} \int_0^4 u e^u du$$

Aplicamos integración por partes:

$$u = u \rightarrow du = du$$

$$dv = e^u du \rightarrow v = e^u$$

$$\int u e^u du = u e^u - \int e^u du = u e^u - e^u + C$$

Entonces:

$$I = \frac{1}{2} [u e^u - e^u]_0^4 = \frac{1}{2} [(4e^4 - e^4) - (0 - e^0)] = \frac{1}{2}(3e^4 + 1)$$

$$I = \frac{3}{2}e^4 + \frac{1}{2} \approx \frac{3}{2}(54.59815003) + \frac{1}{2} = 81.89722505 + 0.5 = 82.39722505$$

Usando la ecuación (6) regla del trapecio para $n = 12$ se tiene:

$$h = \frac{2 - 0}{12} = 0.1667$$

$$\begin{aligned}
T_{12} &= \frac{0.1667}{2} [f(x_0) + f(x_{12})] + 0.1667 \sum_{i=1}^{11} f(x_i) \\
&= \frac{0.1667}{2} [f(0) + f(2)] + 0.1667 [f(0.1667) + f(0.3333) + f(0.5) + f(0.6667) + f(0.8333) + \\
&\quad f(1) + f(1.1667) + f(1.3333) + f(1.5) + f(1.6667) + f(1.8333)]
\end{aligned}$$

$$\begin{aligned}
f(0) &= 0 \\
f(2) &= 8e^4 = 8 \cdot 54.59815003 = 436.7852 \\
f(0.1667) &= (0.1667)^3 e^{(0.1667)^2} = 0.0046 \\
f(0.3333) &= (0.3333)^3 e^{(0.3333)^2} = 0.0640 \\
f(0.5) &= (0.5)^3 e^{(0.5)^2} = 0.5530 \\
f(0.6667) &= (0.6667)^3 e^{(0.6667)^2} = 2.3797 \\
f(0.8333) &= (0.8333)^3 e^{(0.8333)^2} = 7.3427 \\
f(1) &= 1^3 e^{1^2} = e = 2.7183 \\
f(1.1667) &= (1.1667)^3 e^{(1.1667)^2} = 14.7762 \\
f(1.3333) &= (1.3333)^3 e^{(1.3333)^2} = 34.4387 \\
f(1.5) &= (1.5)^3 e^{(1.5)^2} = 72.3314 \\
f(1.6667) &= (1.6667)^3 e^{(1.6667)^2} = 134.5217 \\
f(1.8333) &= (1.8333)^3 e^{(1.8333)^2} = 232.9824
\end{aligned}$$

$$\begin{aligned}
T_{12} &= 0.08335 \cdot (0 + 436.7852) + 0.1667 \cdot (0.0046 + 0.0640 + 0.5530 + 2.3797 + 7.3427 + \\
&\quad 2.7183 + 14.7762 + 34.4387 + 72.3314 + 134.5217 + 232.9824) \\
&= 36.3971 + 51.4754 = 87.8725
\end{aligned}$$

Resultado exacto: 82.3972

Resultado aproximado con regla del trapecio: 87.8725

Como se observa, el resultado obtenido por la regla del trapecio aproxima razonable-

mente bien el valor real. Sin embargo, para funciones con crecimiento exponencial como e^{x^2} , un mayor número de subintervalos mejora la estimación.

1.2.3 Error de truncamiento del método del trapecio

La exactitud que ofrece la regla del trapecio al calcular una integral se relaciona con lo que se conoce como error de truncamiento. Para poder obtener una fórmula que describa este error, es necesario utilizar el Teorema 1. Después, se toma en cuenta el desarrollo en serie de Taylor de la función $F(x)$, centrado en el punto x_i , suponiendo que la función $f(x)$ tiene todas las derivadas continuas que se requieran para el análisis.

$$F(x_{i-1}) = F(x_i - h)$$

$$F(x_{i-1}) = F(x_i) - hF'(x_i) + \frac{h^2}{2!}F''(x_i) - \frac{h^3}{3!}F'''(x_i) + \dots$$

multiplicando por -1 a la ecuación anterior se tiene

$$-F(x_{i-1}) = -F(x_i) + hF'(x_i) - \frac{h^2}{2!}F''(x_i) + \frac{h^3}{3!}F'''(x_i) - \dots$$

de ahí se tiene

$$I_i = F(x_i) - F(x_{i-1}) = hF'(x_i) - \frac{h^2}{2!}F''(x_i) + \frac{h^3}{3!}F'''(x_i) - \dots$$

Siendo $F'(x) = f(x)$, tenemos

$$I_i = hf(x_i) - \frac{h^2}{2!}f'(x_i) + \frac{h^3}{3!}f''(x_i) - \dots \quad (7)$$

Por otra parte, haciendo la expansión de Taylor para $f(x)$, se tiene

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \dots$$

luego, despejando $f'(x)$

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + \frac{h}{2} f''(x_i) + \dots \quad (8)$$

Sustituyendo (8) en (7)

$$\begin{aligned} I_i &= \frac{h}{2} [f(x_i) + f(x_{i-1})] - \frac{h^3}{12} f''(x_i) + \dots \\ &= A_i - \frac{h^3}{12} f''(x_i) + \dots, \text{ donde } A_i = 1, \dots, n. \end{aligned}$$

Por lo tanto, el término de error E_i para el i -ésimo trapecio es

$$\begin{aligned} E_i &= I_i - A_i \\ &= -\frac{h^3}{12} f''(x_i) + \text{términos de orden superior.} \end{aligned}$$

Para h pequeño, despreciamos los términos de orden superior para obtener

$$E_i = -\frac{h^3}{12} f''(x_i) \quad (9)$$

Si $|f''(x)| \leq M$ para todo x en el intervalo $[a, b]$, entonces

$$|E_i| \leq \frac{h^3}{12} M \quad (10)$$

Aplicando (10) sobre todo el intervalo, obtenemos el error total E_T

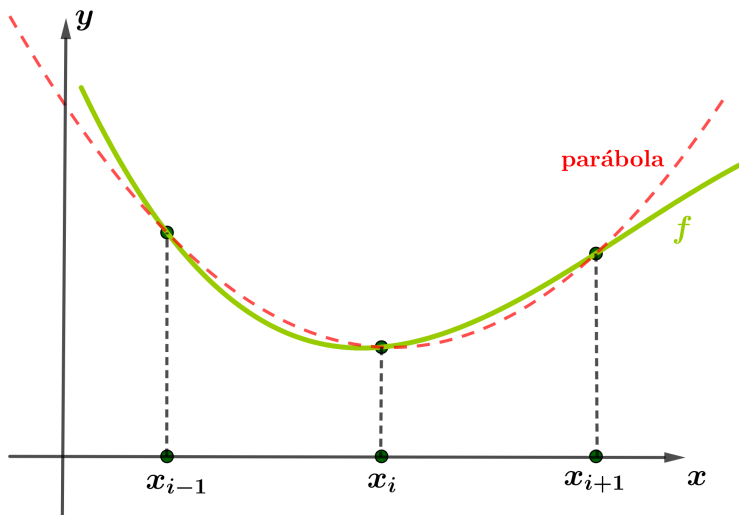
$$|E_T| \leq n \frac{h^3}{12} M = (b-a) \frac{h^2}{12} M \quad (11)$$

1.2.4 Método de Simpson 1/3

De acuerdo con Kharab y Guenther (2012), la regla de Simpson $\frac{1}{3}$ se basa en reemplazar la función $f(x)$ por una parábola que pasa por tres puntos consecutivos: x_{i-1}, x_i y x_{i+1} , como se muestra en la figura (3).

Figura 2

Regla de Simpson $\frac{1}{3}$.



A continuación se busca obtener una fórmula para la regla de Simpson, incluyendo su expresión del error. Para ello, se parte del teorema fundamental del cálculo.

$$I_i = \int_{x_{i-1}}^{x_{i+1}} f(x) dx = F(x_{i+1}) - F(x_{i-1}) \quad (12)$$

donde F es una antiderivada de f .

La expansión de la serie de Taylor de $F(x)$ sobre x_i es

$$\begin{aligned} F(x_{i+1}) &= F(x_i + h) \\ &= F(x_i) + hF'(x_i) + \frac{h^2}{2!}F''(x_i) + \frac{h^3}{3!}F'''(x_i) \\ &\quad + \frac{h^4}{4!}F^{(4)}(x_i) + \frac{h^5}{5!}F^{(5)}(x_i) + \frac{h^6}{6!}F^{(6)}(x_i) + \dots \end{aligned}$$

$$\begin{aligned}
F(x_{i+1}) = & F(x_i) + hf(x_i) + \frac{h^2}{2!}f'(x_i) + \frac{h^3}{3!}f''(x_i) \\
& + \frac{h^4}{4!}f'''(x_i) + \frac{h^5}{5!}f^{(4)}(x_i) + \dots
\end{aligned} \tag{13}$$

Similar,

$$\begin{aligned}
F(x_{i-1}) = & F(x_i) - hf(x_i) + \frac{h^2}{2!}f'(x_i) - \frac{h^3}{3!}f''(x_i) \\
& + \frac{h^4}{4!}f'''(x_i) - \frac{h^5}{5!}f^{(4)}(x_i) + \dots
\end{aligned} \tag{14}$$

Sustituyendo (13) y (14) en (12), obtenemos

$$I_i = 2hf(x_i) + \frac{h^3}{3}f''(x_i) + \frac{h^5}{60}f^{(4)}(x_i) + \dots \tag{15}$$

teniendo el siguiente resultado

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2} - \frac{h^2}{12}f^{(4)}(x_i) + \dots \tag{16}$$

Finalmente, de las ecuaciones. (15) y (16) se obtiene

$$I_i = \frac{h}{3} [f(x_{i+1}) + 4f(x_i) + f(x_{i-1}))] - \frac{h^5}{90}f^{(4)}(x_i) + \dots \tag{17}$$

Por lo tanto, la expresión de Simpson se utiliza para estimar el área bajo la curva en el tramo comprendido entre x_{i-1} , x_i y x_{i+1} .

$$S_i = \frac{h}{3} [f(x_{i+1}) + 4f(x_i) + f(x_{i-1}))] \tag{18}$$

Podemos reescribir la regla de Simpson sobre el intervalo $[x_0, x_2]$ como

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90}f^{(4)}(\xi) \tag{19}$$

donde $x_0 < \xi < x_2$. En vista de las ecuaciones (17) y (19), el término de error para la

regla de Simpson viene dado por

$$E_i = -\frac{h^5}{90} f^{(4)}(x_i)$$

Nuevamente si $|f^{(4)}(x_i)| \leq M$ para todo x en $[a, b]$, entonces

$$|E_i| \leq \frac{h^5}{90} M \quad (20)$$

1.2.5 Método de Simpson 1/3 compuesta

Sea $f(x)$ una función continua en el intervalo cerrado $[a, b]$. La regla de Simpson $\frac{1}{3}$ compuesta se basa en aproximar $f(x)$ mediante polinomios de interpolación cuadráticos en pares consecutivos de subintervalos equidistantes. Para ello, se divide $[a, b]$ en $n = 2m$ subintervalos de igual longitud

$$h = \frac{b-a}{2m},$$

donde $m \in \mathbb{N}$.

La aproximación de la integral definida

$$I = \int_a^b f(x) dx$$

se expresa como:

$$S_n = \frac{h}{3} \left[f(x_0) + 2 \sum_{i=1}^{m-1} f(x_{2i}) + 4 \sum_{i=1}^m f(x_{2i-1}) + f(x_{2m}) \right],$$

donde $x_i = a + ih$, $i = 0, 1, \dots, 2m$.

El término de error de esta aproximación está dado por:

$$|E_T| \leq (b-a) \frac{h^4}{180} M,$$

donde $M = \max_{\xi \in [a, b]} |f^{(4)}(\xi)|$. Este error decrece con h^4 , lo que confiere a esta regla una

precisión superior a la de la regla del trapecio compuesta, cuyo error decrece con h^2 .

La regla de Simpson $\frac{1}{3}$ compuesta es exacta para todo polinomio de grado menor o igual a tres y es especialmente eficiente cuando $f(x)$ presenta un comportamiento suave en el intervalo de integración.

(Kharab y Guenther, 2012).

Ejemplo 2. Resolver la siguiente integral

$$\int_0^2 x^3 e^{x^2} dx$$

Solución.

$$\int_0^2 x^3 e^{x^2} dx = I$$

Resolviendo la siguiente integral por el método de integración por partes y cambio de variable se tiene:

$$u = x^2 \Rightarrow du = 2x dx \Rightarrow x dx = \frac{1}{2} du$$

$$x^3 = x \cdot x^2 = x \cdot u$$

Entonces:

$$I = \int_0^2 x^3 e^{x^2} dx = \frac{1}{2} \int_0^4 u e^u du$$

Aplicando integración por partes:

$$u = u \rightarrow du = du$$

$$dv = e^u du \rightarrow v = e^u$$

$$\int u e^u du = u e^u - \int e^u du = u e^u - e^u + C$$

Evaluando:

$$I = \frac{1}{2}[ue^u - e^u] \Big|_0^4 = \frac{1}{2}(3e^4 + 1) = \frac{3}{2}e^4 + \frac{1}{2}$$

$$I = \frac{3}{2}(54.59815003) + \frac{1}{2} = 81.89722505 + 0.5 = 82.39722505$$

Usando la ecuación (??) se tiene:

Para $n = 12$, $h = \frac{2-0}{12} = 0.1667$

$$\int_0^2 x^3 e^{x^2} dx \approx \frac{0.1667}{3} [f(0) + 4f(0.1667) + 2f(0.3333) + 4f(0.5) + 2f(0.6667) + 4f(0.8333) + 2f(1) + 4f(1.1667) + 2f(1.3333) + 4f(1.5) + 2f(1.6667) + 4f(1.8333) + f(2)]$$

Reemplazando los valores de $f(x_i)$ que se encuentran en la Tabla (1) se tiene:

$$\int_0^2 x^3 e^{x^2} dx \approx \frac{0.1667}{3} [0 + 4(0.0046 + 0.5530 + 7.3427 + 14.7762 + 72.3314 + 232.9824) + 2(0.0640 + 2.3797 + 2.7183 + 34.4387 + 134.5217) + 436.7852]$$

$$\begin{aligned} \int_0^2 x^3 e^{x^2} dx &\approx \frac{0.1667}{3} [0 + 4(327.9903) + 2(174.1224) + 436.7852] \\ &\approx \frac{0.1667}{3} [1311.9612 + 348.2448 + 436.7852] \\ &\approx \frac{0.1667}{3} (2096.9912) \\ &\approx 82.7079 \end{aligned}$$

Tabla 1

Valores de $f(x_i) = x_i^3 e^{x_i^2}$

i	x_i	$f(x_i)$
0	0	0
1	0.1667	0.0046
2	0.3333	0.0640
3	0.5	0.5530
4	0.6667	2.3797
5	0.8333	7.3427
6	1	2.7183
7	1.1667	14.7762
8	1.3333	34.4387
9	1.5	72.3314
10	1.6667	134.5217
11	1.8333	232.9824
12	2	436.7852

Como se observa, la regla de Simpson $\frac{1}{3}$ compuesta aproxima de manera excelente el valor exacto. Esta técnica resulta más precisa que la regla del trapecio para funciones como $x^3 e^{x^2}$.

1.2.6 Método de Simpson 3/8

Kharab y Guenther (2012)

Sea $f : [a, b] \rightarrow \mathbb{R}$ una función continua y lo suficientemente derivable en el intervalo de integración. La regla de Simpson $\frac{3}{8}$ se basa en aproximar la integral definida de $f(x)$ mediante un polinomio interpolante de grado tres que pasa por cuatro puntos igualmente espaciados: x_i, x_{i+1}, x_{i+2} y x_{i+3} .

Si el intervalo $[a, b]$ se divide en n subintervalos de igual longitud $h = \frac{b-a}{n}$, con n múltiplo de 3.

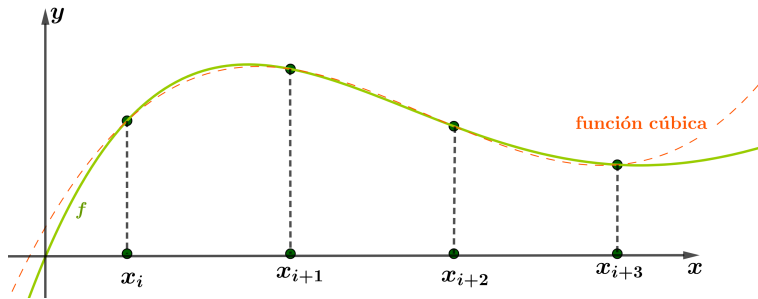


Figura 3

Regla de Simpson $\frac{3}{8}$.

Por el teorema fundamental del cálculo tenemos

$$I_i = \int_{x_i}^{x_{i+3}} f(x) dx = F(x_{i+3}) - F(x_i) \quad (21)$$

La expansión de la serie de Taylor de $F(x)$ sobre x_i es

$$\begin{aligned} F(x_{i+3}) &= F(x_i + 3h) \\ &= F(x_i) + 3hF'(x_i) + \frac{(3h)^2}{2!}F''(x_i) + \frac{(3h)^3}{3!}F'''(x_i) \\ &\quad + \frac{(3h)^4}{4!}F^{(4)}(x_i) + \frac{(3h)^5}{5!}F^{(5)}(x_i) + \frac{(3h)^6}{6!}F^{(6)}(x_i) + \dots \end{aligned}$$

$$\begin{aligned} F(x_{i+3}) &= F(x_i) + 3hf'(x_i) + \frac{9h^2}{2}f''(x_i) + \frac{9h^3}{2}f'''(x_i) \\ &\quad + \frac{27h^4}{8}f^{(4)}(x_i) + \frac{81h^5}{40}f^{(5)}(x_i) + \dots \end{aligned}$$

$$\begin{aligned} F(x_{i+3}) - F(x_i) &= 3hf'(x_i) + \frac{9h^2}{2}f''(x_i) + \frac{9h^3}{2}f'''(x_i) \\ &\quad + \frac{27h^4}{8}f^{(4)}(x_i) + \frac{81h^5}{40}f^{(5)}(x_i) + \dots \end{aligned} \tag{22}$$

Tener en cuenta que debemos aproximar I_i e función de $f(x_i)$, $f(x_{i+1})$, $f(x_{i+2})$ y $f(x_{i+3})$, entonces hacemos lo siguiente:

Tomamos la serie de Taylor para $x_i + 3h$ entorno a x_i , se tiene:

$$f(x_i + 3h) = f(x_i) + 3hf'(x_i) + \frac{9h^2}{2}f''(x_i) + \frac{9h^3}{2}f'''(x_i) + \frac{27h^4}{8}f^{(4)}(x_i) + \dots \tag{23}$$

Ahora tomamos la serie de Taylor para $x_i + 2h$ entorno a x_i , se tiene

$$f(x_i + 2h) = f(x_i) + 2hf'(x_i) + 2h^2f''(x_i) + \frac{4h^3}{3}f'''(x_i) + \frac{2h^4}{3}f^{(4)}(x_i) + \dots \tag{24}$$

luego tomamos la serie de Taylor para $x_i + h$ entorno a x_i , se tiene

$$f(x_i + h) = f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f'''(x_i) + \frac{h^4}{24}f^{(4)}(x_i) + \dots \tag{25}$$

multiplicando por $\lambda_1 h$ a (23), $\lambda_2 h$ a (24) y $\lambda_3 h$ a (2,26), se obtiene

$$\lambda_1 h f(x_i + 3h) = \lambda_1 h f(x_i) + 3\lambda_1 h^2 f'(x_i) + \frac{9\lambda_1 h^3}{2} f''(x_i) + \frac{9\lambda_1 h^4}{2} f'''(x_i) + \frac{27\lambda_1 h^5}{8} f^{(4)}(x_i) + \dots$$

(26)

$$\lambda_2 h f(x_i + 2h) = \lambda_2 h f(x_i) + 2\lambda_2 h^2 f'(x_i) + 2\lambda_2 h^3 f''(x_i) + \frac{4\lambda_2 h^4}{3} f'''(x_i) + \frac{2\lambda_2 h^5}{3} f^{(4)}(x_i) + \dots$$

(27)

$$\lambda_3 h f(x_i + h) = \lambda_3 h f(x_i) + \lambda_3 h^2 f'(x_i) + \frac{\lambda_3 h^3}{2} f''(x_i) + \frac{\lambda_3 h^4}{6} f'''(x_i) + \frac{\lambda_3 h^5}{24} f^{(4)}(x_i) + \dots$$

(28)

Esto hacemos para obtener la expresión $\frac{9h^2}{2} f'(x_i) + \frac{9h^3}{2} f''(x_i) + \frac{27h^4}{8} f'''(x_i)$ de la ecuación (22), en función de $f(x_i)$, $f(x_i + h)$, $f(x_i + 2h)$ y $f(x_i + 3h)$ entonces de las tres últimas ecuaciones y la expresión anterior se obtiene el siguiente sistema:

$$\begin{aligned} 3\lambda_1 + 2\lambda_2 + \lambda_3 &= \frac{9}{2} \\ \frac{9}{2}\lambda_1 + 2\lambda_2 + \frac{1}{2}\lambda_3 &= \frac{9}{2} \\ \frac{9}{2}\lambda_1 + \frac{4}{3}\lambda_2 + \frac{1}{6}\lambda_3 &= \frac{27}{8} \end{aligned}$$

resolviendo se obtiene: $\lambda_1 = \frac{3}{8}$, $\lambda_2 = \frac{9}{8}$ y $\lambda_3 = \frac{9}{8}$

reemplazando cada λ en las ecuaciones (26), (27) y (28) se obtiene

$$\frac{3}{8} h f(x_i + 3h) = \frac{3}{8} h f(x_i) + \frac{9}{8} h^2 f'(x_i) + \frac{27h^3}{16} f''(x_i) + \frac{27h^4}{16} f'''(x_i) + \frac{81h^5}{64} f^{(4)}(x_i) + \dots$$

(29)

$$\frac{9}{8}hf(x_i + 2h) = \frac{9}{8}hf(x_i) + \frac{9}{4}h^2f'(x_i) + \frac{9}{4}h^3f''(x_i) + \frac{3}{2}h^4f'''(x_i) + \frac{3h^5}{4}f^{(4)}(x_i) + \dots \quad (30)$$

$$\frac{9}{8}hf(x_i + h) = \frac{9}{8}hf(x_i) + \frac{9}{8}h^2f'(x_i) + \frac{9h^3}{16}f''(x_i) + \frac{3h^4}{16}f'''(x_i) + \frac{3h^5}{64}f^{(4)}(x_i) + \dots \quad (31)$$

sumando se tiene

$$\begin{aligned} \frac{3h}{8} [f(x_i + 3h) + 3f(x_i + 2h) + 3f(x_i + h)] &= \frac{21}{8}hf(x_i) + \frac{9h^2}{2}f'(x_i) + \frac{9h^3}{2}f''(x_i) + \frac{27h^4}{8}f'''(x_i) \\ &+ \frac{33}{16}h^5f^{(4)}(x_i) + \dots \end{aligned}$$

$$\begin{aligned} \frac{9h^2}{2}f'(x_i) + \frac{9h^3}{2}f''(x_i) + \frac{27h^4}{8}f'''(x_i) &= \frac{3h}{8} [f(x_i + 3h) + 3f(x_i + 2h) + 3f(x_i + h)] - \frac{21}{8}hf(x_i) \\ &- \frac{33}{16}h^5f^{(4)}(x_i) - \dots \end{aligned} \quad (32)$$

Reemplazando (32) en (22), se tiene

$$\begin{aligned} F(x_{i+3}) - F(x_i) &= 3hf(x_i) + \frac{3h}{8} [f(x_i + 3h) + 3f(x_i + 2h) + 3f(x_i + h)] - \frac{21}{8}hf(x_i) \\ &- \frac{33}{16}h^5f^{(4)}(x_i) + \frac{81h^5}{40}f^{(4)}(x_i) + \dots \end{aligned}$$

teniendo en cuenta que $x_{i+1} = x_i + h$, $x_{i+2} = x_i + 2h$ y $x_{i+3} = x_i + 3h$ se obtiene

$$F(x_{i+3}) - F(x_i) = \frac{3h}{8} [f(x_i) + 3f(x_{i+1}) + 3f(x_{i+2}) + f(x_{i+3})] - \frac{3}{80}h^5f^{(4)}(x_i) - \dots \quad (33)$$

Por lo tanto, la regla de Simpson para aproximar el área sobre dos subintervalos que se

encuentran entre x_i, x_{i+1}, x_{i+2} y x_{i+3} está dada por

$$S_i = \frac{3h}{8} [f(x_i) + 3f(x_{i+1}) + 3f(x_{i+2}) + f(x_{i+3})] \quad (34)$$

Podemos reescribir la regla de Simpson sobre el intervalo $[x_0, x_3]$ como

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3}{80}h^5 f^{(4)}(\xi) \quad (35)$$

donde $x_0 < \xi < x_3$, $h = \frac{b-a}{n}$, n es múltiplo de 3. En vista de las ecuaciones. (33) y (34), el término de error para la regla de Simpson viene dado por

$$E_i = -\frac{3h^5}{80} f^{(4)}(x_\xi)$$

Nuevamente si $|f^{(4)}(x_i)| \leq M$ para todo x en $[a, b]$, entonces

$$|E_i| \leq \frac{3h^5}{80} M \quad (36)$$

1.2.7 Método de Simpson 3/8 compuesta

La fórmula (34) proporciona una aproximación para la integral definida:

$$I = \int_{x_0}^{x_n} f(x) dx,$$

donde $f : [x_0, x_n] \rightarrow \mathbb{R}$ es una función continua y lo suficientemente derivable en el intervalo de integración.

La regla de Simpson $\frac{3}{8}$ aproxima $f(x)$ mediante un polinomio interpolante de grado tres que pasa por cuatro puntos equidistantes. En su versión simple, esta regla se aplica sobre tres subintervalos de igual longitud h . No obstante, cuando el tramo total $[x_0, x_n]$ es extenso, aplicar la fórmula en una sola sección puede disminuir la precisión de la estimación.

Para mejorar la exactitud, el intervalo se divide en n subintervalos de igual longitud

$h = \frac{x_n - x_0}{n}$, donde n es múltiplo de 3, es decir, $n = 3m$ con $m \in \mathbb{N}$. De esta manera, la regla de Simpson $\frac{3}{8}$ se aplica de forma repetida sobre cada grupo de tres subintervalos consecutivos, logrando así una aproximación más precisa de la integral.

$$I = \int_{x_0}^{x_n} f(x)dx \approx \int_{x_0}^{x_3} f(x)dx + \int_{x_3}^{x_6} f(x)dx + \int_{x_6}^{x_9} f(x)dx + \cdots + \int_{x_{n-3}}^{x_n} f(x)dx$$

$$I \approx \frac{3h}{8} \left[\left(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3) \right) + \left(f(x_3) + 3f(x_4) + 3f(x_5) + f(x_6) \right) + \right. \\ \left. \left(f(x_6) + 3f(x_7) + 3f(x_8) + f(x_9) \right) + \cdots + \left(f(x_{n-3}) + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n) \right) \right]$$

Entonces usando (35), obtenemos:

$$I = \int_{x_0}^{x_n} f(x)dx \approx \sum_{i=1}^m \left[\frac{3h}{8} [f(x_{3i-3}) + 3f(x_{3i-2}) + 3f(x_{3i-1}) + f(x_{3i})] \right] - \sum_{i=1}^{3m} \frac{3h^5}{80} f^{(4)}(x_\xi) \quad (37)$$

En (37) ambas sumatorias son de $i=1$ hasta m . Donde $\xi \in [x_0, x_n]$.

Al despreciar el término de error, obtenemos:

$$I = \int_{x_0}^{x_n} f(x)dx \approx S_n = \sum_{i=1}^m \left[\frac{3h}{8} [f(x_{3i-3}) + 3f(x_{3i-2}) + 3f(x_{3i-1}) + f(x_{3i})] \right] \quad (38)$$

donde

$$h = \frac{b-a}{3m}, i = 1, 2, \dots, m$$

La fórmula (38) se llama regla de **Simpson $\frac{3}{8}$ compuesta**.

Ejemplo 3. Resolver la siguiente integral

$$\int_0^2 x^3 e^{x^2} dx$$

Solución.

Sea:

$$\int_0^2 x^3 e^{x^2} dx = I$$

Aplicamos el cambio de variable:

$$u = x^2 \Rightarrow du = 2x dx \Rightarrow x dx = \frac{1}{2} du$$

$$x^3 = x \cdot x^2 = x \cdot u$$

Sustituyendo en la integral:

$$I = \int_0^2 x^3 e^{x^2} dx = \int_0^2 x \cdot u \cdot e^u dx$$

Transformamos completamente a la variable u :

$$I = \int_{u=0}^{u=4} ue^u \cdot \frac{1}{2} du = \frac{1}{2} \int_0^4 ue^u du$$

Aplicamos integración por partes:

$$u = u \rightarrow du = du$$

$$dv = e^u du \rightarrow v = e^u$$

$$\int ue^u du = ue^u - \int e^u du = ue^u - e^u + C$$

Entonces:

$$I = \frac{1}{2} [ue^u - e^u] \Big|_0^4 = \frac{1}{2} [(4e^4 - e^4) - (0 - e^0)] = \frac{1}{2}(3e^4 + 1)$$

$$I = \frac{3}{2}e^4 + \frac{1}{2} \approx \frac{3}{2}(54.59815003) + \frac{1}{2} = 81.89722505 + 0.5 = 82.39722505$$

Resolución numérica por Simpson 3/8 compuesta

Usando $n = 4$, $h = \frac{2-0}{12} = 0.1667$

$$\int_0^2 x^3 e^{x^2} dx \approx \sum_{i=1}^4 \left[\frac{3h}{8} (f(x_{3i-3}) + 3f(x_{3i-2}) + 3f(x_{3i-1}) + f(x_{3i})) \right]$$

$$\begin{aligned} \int_0^2 x^3 e^{x^2} dx &\approx \sum_{i=1}^{12} \left[\frac{3h}{8} [f(x_{3i-3}) + 3f(x_{3i-2}) + 3f(x_{3i-1}) + f(x_{3i})] \right] \\ &\approx \frac{3(0.25)}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] + \\ &\quad \frac{3(0.25)}{8} [f(x_3) + 3f(x_4) + 3f(x_5) + f(x_6)] + \\ &\quad \frac{3(0.25)}{8} [f(x_6) + 3f(x_7) + 3f(x_8) + f(x_9)] + \\ &\quad \frac{3(0.25)}{8} [f(x_9) + 3f(x_{10}) + 3f(x_{11}) + f(x_{12})] \end{aligned}$$

Tabla 2

Tabulando la función $f(x) = x^3 e^{x^2}$ para $h = 0.1667$ en el intervalo $[0, 2]$

x_i	$f(x_i)$
0.00000000	0.00000000
0.16666667	0.00476003
0.33333333	0.04138959
0.50000000	0.16050318
0.66666667	0.46211067
0.83333333	1.15890982
1.00000000	2.71828183
1.16666667	6.19388904
1.33333333	14.02475495
1.50000000	32.02110845
1.66666667	74.45944883
1.83333333	177.59727805
2.00000000	436.78520027

$$\begin{aligned} &= \frac{3(0.1667)}{8} [0 + 3(0.00476003) + 3(0.04138959) + 0.16050318] + \\ &\quad \frac{3(0.1667)}{8} [0.16050318 + 3(0.46211067) + 3(1.15890982) + 2.71828183] + \\ &\quad \frac{3(0.1667)}{8} [2.71828183 + 3(6.19388904) + 3(14.02475495) + 32.02110845] + \\ &\quad \frac{3(0.1667)}{8} [32.02110845 + 3(74.45944883) + 3(177.59727805) + 436.78520027] \end{aligned}$$

$$\begin{aligned}
&= 0.031383 + 0.592129 + 6.466927 + 75.931588 \\
&= \boxed{83.0258}
\end{aligned}$$

Comparación

- Solución exacta: ≈ 82.3972
- Solución numérica: ≈ 83.0258

Tabla 3

Comparación de resultado de $\int_0^2 x^3 e^{x^2} dx$

Método	Subintervalos	Solución analítica	Solución numérica	error
Trapecio n=12	12	82.39722505	87.8725	5.47527495
Simpson 1/3 m=6	12	82.39722505	82.7079	0.31067495
Simpson 3/8 m=4	12	82.39722505	83.0258	0.62857495

En la Tabla (3) se presenta una comparación entre los resultados analíticos y numéricos obtenidos para la integral definida $\int_0^2 x^3 e^{x^2} dx$, utilizando los métodos del trapecio, Simpson compuesto $\frac{1}{3}$ y Simpson compuesto $\frac{3}{8}$, cada uno con 12 subintervalos. Se observa que el método de Simpson compuesto $\frac{1}{3}$ proporciona la aproximación más precisa, con un error de tan solo 0.3107, en contraste con el método del trapecio que presenta un error significativamente mayor (5.4753). El método de Simpson compuesto $\frac{3}{8}$ también ofrece una buena aproximación, aunque con un error ligeramente superior (0.6286) al del método de Simpson compuesto $\frac{1}{3}$. Por lo tanto, se concluye que el método de Simpson compuesto $\frac{1}{3}$ resulta ser el más eficiente y preciso para este caso particular.

1.2.8 Objetos de Diseño en App Designer de Matlab R2021a

1. **UIFigure** Es la ventana principal de la aplicación. Todos los demás componentes se colocan dentro de esta figura. Representa la base del diseño visual.
2. **Label (Etiqueta)** Muestra texto estático. Se usa para indicar nombres de campos, títulos, instrucciones o resultados.

3. **Edit Field (Campo de texto)** Permite al usuario ingresar datos, como números o funciones. Hay campos de texto simples, numéricos y de texto largo.
4. **Numeric Edit Field** Específicamente diseñado para ingresar y mostrar valores numéricos con validación automática.
5. **Text Area** Se utiliza para mostrar o ingresar bloques de texto más largos, útil para resultados extensos o comentarios.
6. **Button (Botón)** Al hacer clic, ejecuta una acción programada (por ejemplo: calcular, graficar, limpiar). Es el componente interactivo más usado.
7. **Drop Down (Lista desplegable)** Muestra una lista de opciones. El usuario puede seleccionar una entre varias (por ejemplo, método de integración).
8. **Check Box (Casilla de verificación)** Permite activar o desactivar una opción específica. Útil para seleccionar múltiples condiciones.
9. **Radio Button (Botón de opción)** Similar a la casilla, pero solo permite seleccionar una opción entre un grupo.
10. **Switch (Interruptor)** Cambia entre dos estados, como ON/OFF o sí/no. Representa decisiones binarias.
11. **Slider (Deslizador)** Permite seleccionar un valor dentro de un rango moviendo un control horizontal o verticalmente. Ideal para ajustar valores dinámicamente.
12. **Knob (Perilla)** Representa un control circular para seleccionar valores continuos o discretos, visualmente atractivo.
13. **Lamp (Luz indicadora)** Muestra un color para señalar un estado, como verde para correcto o rojo para error. No se puede hacer clic, solo es visual.
14. **Axes (Ejes de gráfico)** Área dedicada a mostrar gráficas, como funciones, áreas integradas o resultados visuales. Soporta múltiples tipos de gráficos.

15. **Gauge (Indicador tipo medidor)** Muestra valores numéricos en forma circular o semicircular (como un velocímetro). Útil para representar resultados dinámicos.
16. **Progress Bar (Barra de progreso)** Indica el avance de un proceso, como una carga o una ejecución prolongada. Es solo visual.
17. **Table (Tabla de datos)** Muestra y permite editar datos en filas y columnas. Se usa para mostrar resultados intermedios, iteraciones o conjuntos de datos.
18. **Panel** Contenedor que agrupa varios componentes. Ayuda a organizar la interfaz en secciones, como entradas, resultados y controles.
19. **Tab Group (Pestañas)** Permite dividir la GUI en páginas o secciones organizadas por pestañas, sin crear varias ventanas.
20. **Image** Inserta una imagen (como un logo, fondo o figura). Se usa con fines visuales o estéticos. Spinner Componente con flechas arriba/abajo para aumentar o disminuir un valor numérico.

1.3 Bases Conceptuales

1.3.1 Integración Numérica

La integración numérica permite encontrar una aproximación del valor de una integral definida cuando no se puede calcular exactamente. Se usa frecuentemente en ingeniería y ciencias aplicadas cuando la función es compleja o cuando los datos provienen de mediciones experimentales. El principio básico es aproximar el área bajo una curva dividiendo el intervalo en partes pequeñas y sumando áreas de figuras simples, como trapecios o parábolas. La forma general se representa como:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n A_i$$

Donde A_i es el área aproximada de cada subintervalo. La precisión mejora si se incrementa el número de subintervalos o si se elige un método más avanzado. (Ribeiro, 2023).

1.3.2 Método del Trapecio

El método del trapecio es uno de los métodos más simples de integración numérica. Se basa en aproximar el área bajo la curva mediante trapecios. Se divide el intervalo $[a, b]$ en n partes iguales y se calcula el área de cada trapecio formado entre puntos consecutivos.

La fórmula compuesta es:

$$\int_a^b f(x)dx \approx \frac{h}{2} \left[f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right]$$

Donde $h = \frac{b-a}{n}$ y x_i son los puntos equidistantes. (Tineo & Torres, 2022)

1.3.3 Regla de Simpson 1/3

La regla de Simpson 1/3 mejora la precisión del método del trapecio al usar parábolas para aproximar la función. Divide el intervalo $[a, b]$ en un número par de subintervalos y utiliza una combinación de los valores de la función en puntos específicos.

Su fórmula compuesta es:

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[f(x_0) + 4 \sum_{\text{impares}} f(x_i) + 2 \sum_{\text{pares}} f(x_i) + f(x_n) \right]$$

Donde $h = \frac{b-a}{n}$, y n debe ser par.

Este método es más preciso cuando la función es continua y suave. Se utiliza ampliamente en problemas de física e ingeniería. Es ideal para funciones cuadráticas o cercanas a una parábola, logrando resultados muy cercanos al valor real. (Tineo & Torres, 2022)

1.3.4 Regla de Simpson 3/8

La regla de Simpson 3/8 es una extensión del método de Simpson que utiliza polinomios de tercer grado para mejorar la aproximación. Se requiere que el número de subintervalos sea múltiplo de tres. Este método ajusta la curva con una función cúbica pasando por cuatro puntos.

La fórmula compuesta es:

$$\int_a^b f(x)dx \approx \frac{3h}{8} \left[f(x_0) + 3 \sum f(x_i) + 2 \sum f(x_j) + f(x_n) \right]$$

Donde $h = \frac{b-a}{n}$, y los índices i y j corresponden a posiciones específicas de los subintervalos. Se aplica cuando la función presenta mayor curvatura, logrando más precisión que Simpson 1/3 en ciertos casos. Aunque es menos conocido, se usa como complemento para lograr exactitud cuando el número de intervalos no es par. (Tineo & Torres, 2022)

1.3.5 Interfaces gráficas de usuario (GUI) en App Designer de Matlab R2021a

Una interfaz gráfica de usuario (GUI) es una ventana interactiva que contiene elementos gráficos (botones, campos de texto, menús desplegables, gráficas, etc.) con los cuales el usuario puede interactuar. Su propósito es facilitar la comunicación entre el usuario y los cálculos o procesos complejos, brindando un entorno visual e intuitivo en lugar de depender únicamente de comandos textuales o código. En el contexto de MATLAB, una GUI permite

al usuario manipular parámetros, ingresar datos y visualizar resultados de manera amigable, todo dentro de una aplicación unificada. Por ejemplo, en una GUI para integración numérica, el usuario podría introducir la función a integrar, los límites de integración y el método numérico deseado mediante controles gráficos, y el programa mostraría el resultado calculado y quizás la gráfica del área bajo la curva. (MathWorks, 2024).

1.3.6 Matlab R2021a

Matlab R2021a es una plataforma ampliamente utilizada en ambientes académicos y profesionales debido a su capacidad para desarrollar algoritmos, visualizar datos y crear aplicaciones interactivas. Una de sus ventajas es que permite construir interfaces gráficas de usuario (GUIs) de manera sencilla, sin necesidad de conocimientos avanzados en programación visual. En versiones recientes, destaca el uso de App Designer, una herramienta integrada que facilita el diseño de interfaces mediante un entorno gráfico con componentes arrastrables, combinando diseño y programación en un solo espacio.

App Designer ha reemplazado progresivamente al antiguo GUIDE, ofreciendo una experiencia más moderna, clara y adecuada para futuras versiones del software. Uno de sus principales beneficios es la posibilidad de generar aplicaciones ejecutables de manera independiente mediante Matlab R2021a Compiler, lo que permite que la herramienta funcione sin necesidad de tener Matlab R2021a instalado en el equipo del usuario. Esto es especialmente útil en proyectos educativos o profesionales, ya que las aplicaciones pueden distribuirse como archivos .exe y utilizarse sin licencias adicionales.

En el contexto de la integración numérica, App Designer permite crear programas gráficos interactivos donde el usuario puede elegir métodos de cálculo, ingresar funciones y visualizar resultados de forma accesible, lo que amplía el alcance y la utilidad de Matlab R2021a en múltiples disciplinas. (MathWorks, 2024)

II. DISEÑO METODOLÓGICO

2.1 Tipo de Investigación

La investigación es de tipo aplicada, ya que se enfoca en el desarrollo de una solución práctica mediante la implementación de métodos de integración numérica (Trapezio, Simpson 1/3, Simpson 3/8) en una interfaz gráfica desarrollada con App Designer de Matlab R2021a. Este tipo de investigación busca resolver problemas reales de cálculo aproximado de integrales definidas, facilitando su comprensión y aplicación mediante herramientas tecnológicas.

2.2 Método de Investigación

El método de investigación es cuantitativo-experimental. Se obtendrán datos numéricos mediante simulaciones controladas en la interfaz diseñada, con el propósito de evaluar la precisión y eficiencia de los métodos de integración numérica implementados. Los resultados serán comparados con soluciones analíticas conocidas o calculadas con software especializado, permitiendo validar la utilidad del ejecutable en contextos educativos.

2.3 Diseño de Contrastación

El diseño de contrastación adopta un enfoque iterativo de desarrollo, prueba y validación. Las fases son:

- **Diseño de la Interfaz Gráfica (GUI) en App Designer:** Se estructura una interfaz intuitiva que permita seleccionar el método de integración, ingresar la función, los límites de integración, y visualizar el resultado numérico y gráfico.
- **Implementación de Métodos Numéricos:** Se codifican en Matlab R2021a los algoritmos de integración numérica, tales como el método del trapecio, Simpson 1/3, Simpson 3/8.
- **Pruebas Iterativas:** Se prueban diferentes funciones con soluciones conocidas. Se analizan los errores y se ajusta el diseño para mejorar la precisión y la experiencia del

usuario.

- **Validación:** Se contrastan los resultados numéricos obtenidos por la app con los resultados analíticos o de otros softwares, garantizando su correcto funcionamiento.

2.4 Población, Muestra y Muestreo

Esta investigación no contempla el uso de población ni muestra, ya que se centra en el desarrollo y validación de un software académico.

2.5 Técnicas, Instrumentos, Equipos y Materiales de Recolección de Datos

- **Técnicas:** Simulación computacional, análisis numérico y comparación de resultados.
- **Instrumentos:** Aplicación ejecutable desarrollada con App Designer, código Matlab R2021a, herramientas de depuración y visualización.
- **Equipos:** Computadora con capacidad suficiente para ejecutar Matlab R2021a y compilar aplicaciones (.exe).
- **Materiales:** Funciones matemáticas seleccionadas, guías de experimentación, referencias analíticas y documentación técnica de los métodos de integración numérica.

2.6 Procesamiento y Análisis de Datos

- **Procesamiento de Datos:** Los valores obtenidos mediante el uso de la aplicación serán procesados automáticamente por Matlab R2021a, permitiendo evaluar numéricamente las integrales introducidas por el usuario.
- **Análisis de Datos:** Se realizarán comparaciones entre los resultados obtenidos por cada método de integración numérica en la interfaz y los resultados exactos calculados analíticamente. Se analizará el error absoluto de cada aproximación.
- **Validación de Resultados:** La validación se realizará mediante pruebas repetidas con funciones conocidas. Se observará la convergencia del resultado con el incremento del número de subintervalos, y se verificarán los errores cometidos en cada método.

III. RESULTADOS

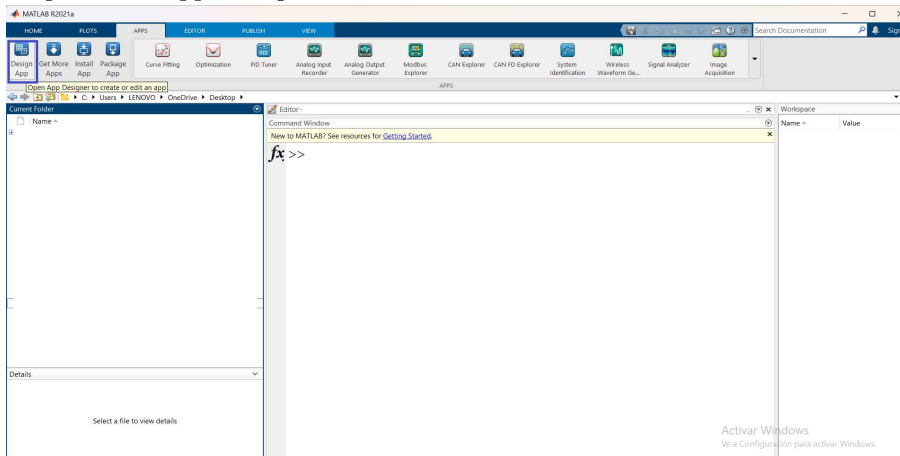
3.1 Diseño de la Interfaz Gráfica de Usuario (GUI) en App Designer.

3.1.1 Diseño de la GUI del Método del Trapecio.

Paso 1: La Figura (4) muestra como se debe ingresar en App Designer.

Figura 4

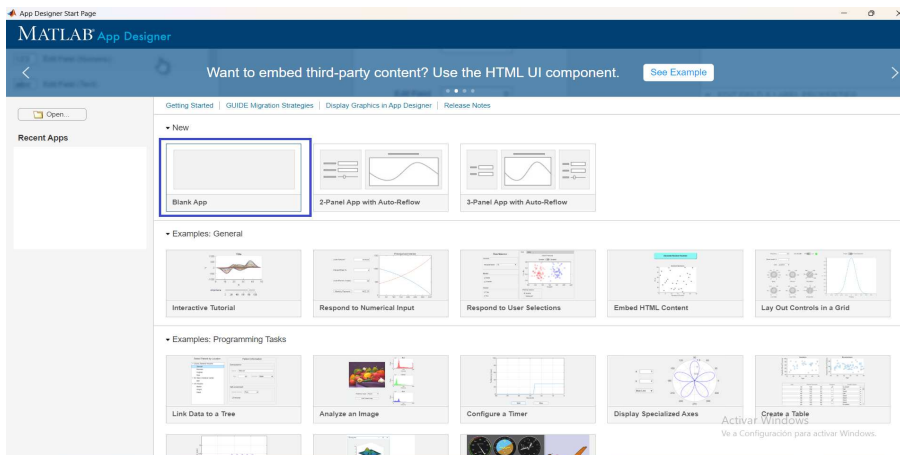
Ingreso en App Designer.



Paso 2: La Figura (5) muestra para seleccionar una ventana en blanco para ingresar en App Designer.

Figura 5

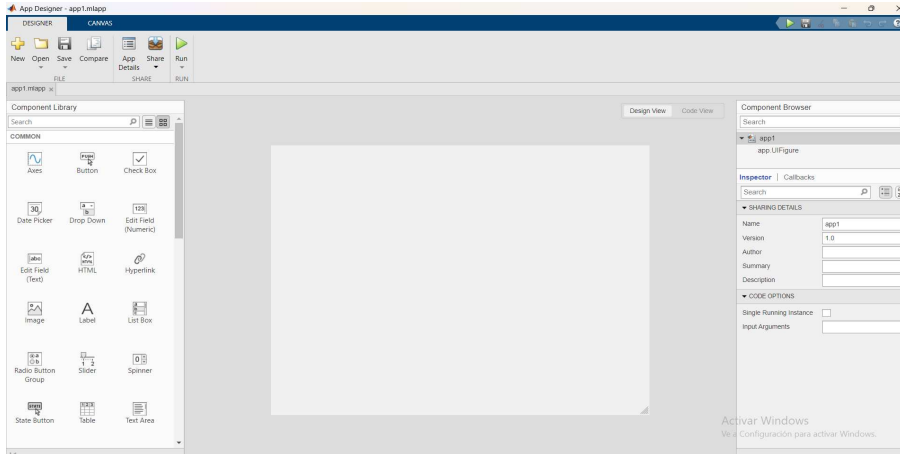
Se selecciona una ventana en blanco.



Paso 3: La Figura (6) muestra la ventana de diseño de la interfaz gráfica de usuario en App Designer.

Figura 6

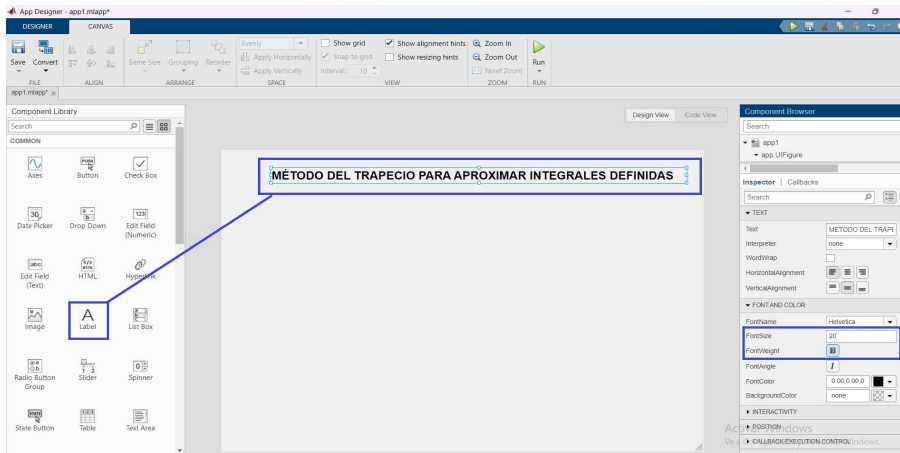
Interfaz gráfica de usuario.



Paso 4: La Figura (7) muestra como se debe ingresar un Label para colocar el título del método del trapecio para aproximar integrales definidas.

Figura 7

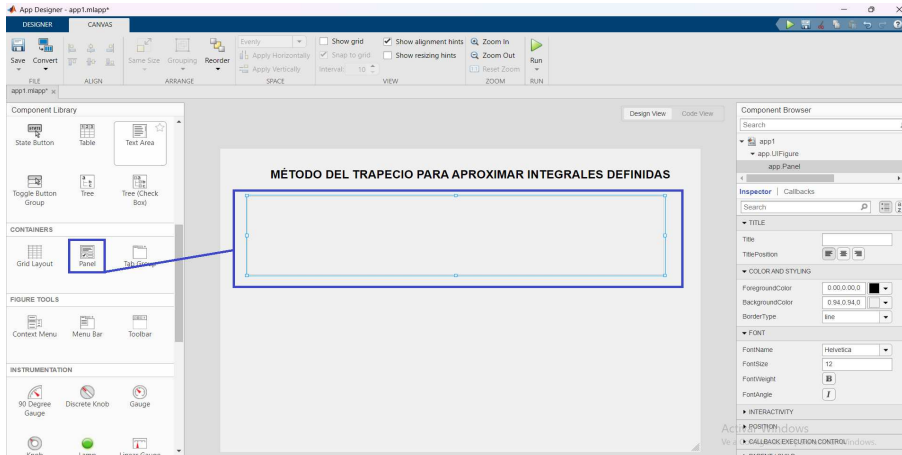
Se coloca un Label para el título.



Paso 5: La Figura (8) muestra como se debe ingresar un panel 1.

Figura 8

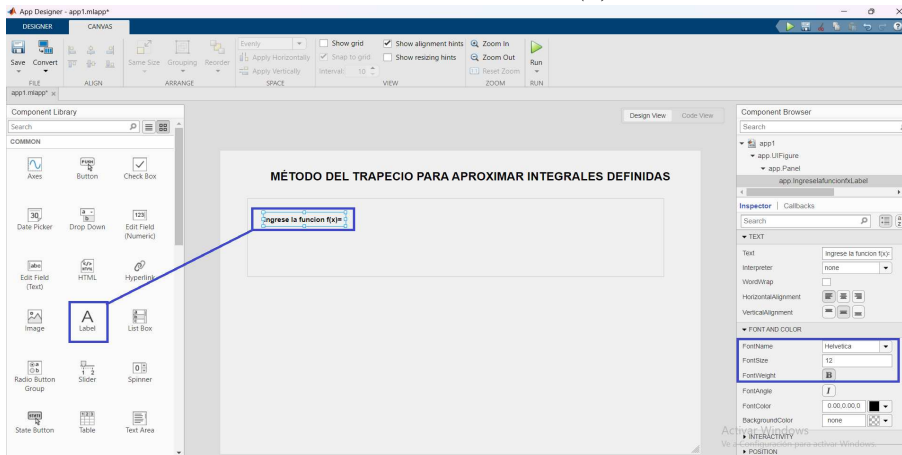
Ingresar un panel 1.



Paso 6: La Figura (9) muestra como se debe ingresar un Label para colocar el nombre de "Ingrese la funcion f(x)=:".

Figura 9

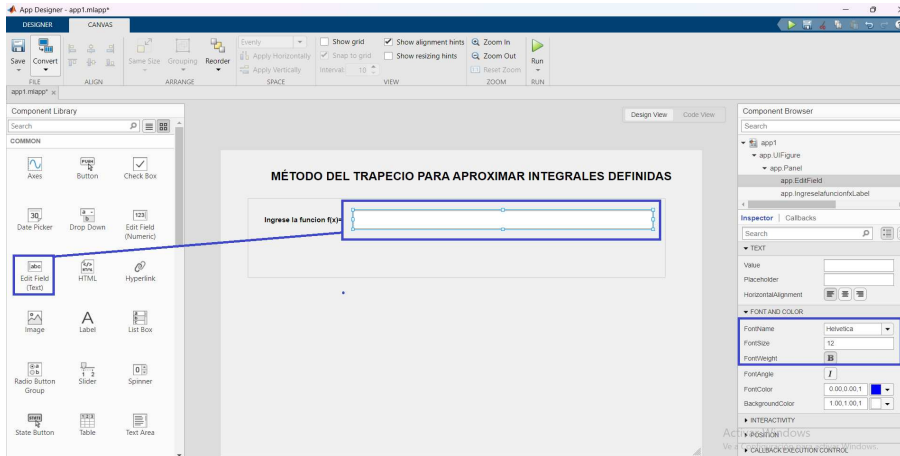
Se coloca el nombre de "Ingrese la funcion f(x)=:".



Paso 7: La Figura (10) muestra como se debe ingresar un Edit Text Fiel, para poder ingresar la función.

Figura 10

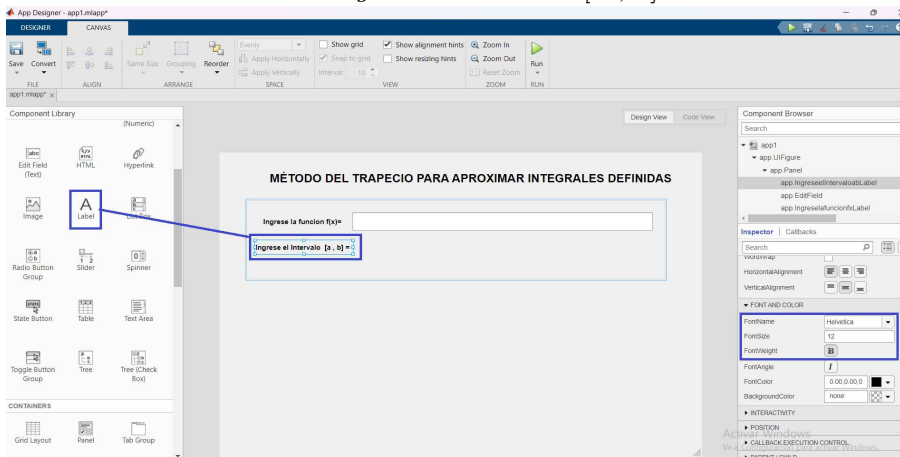
Edit Text Fiel, para ingresar $f(x)$.



Paso 8: La Figura (11) muestra como se debe ingresar un Label para colocar el nombre de "Ingrese el Intervalo [a , b] =".

Figura 11

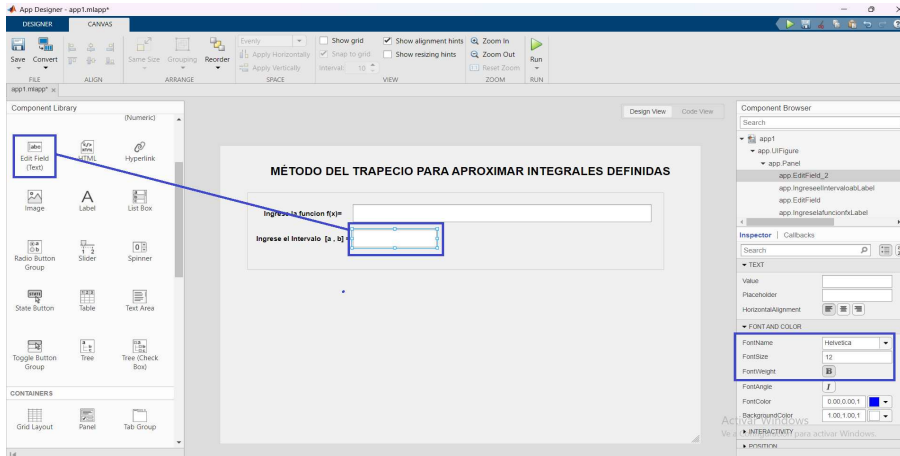
Se coloca el nombre de "Ingrese el Intervalo [a , b] =".



Paso 9: La Figura (12) muestra como se debe ingresar un Edit Text Fiel, para poder ingresar el intervalo $[a,b]$.

Figura 12

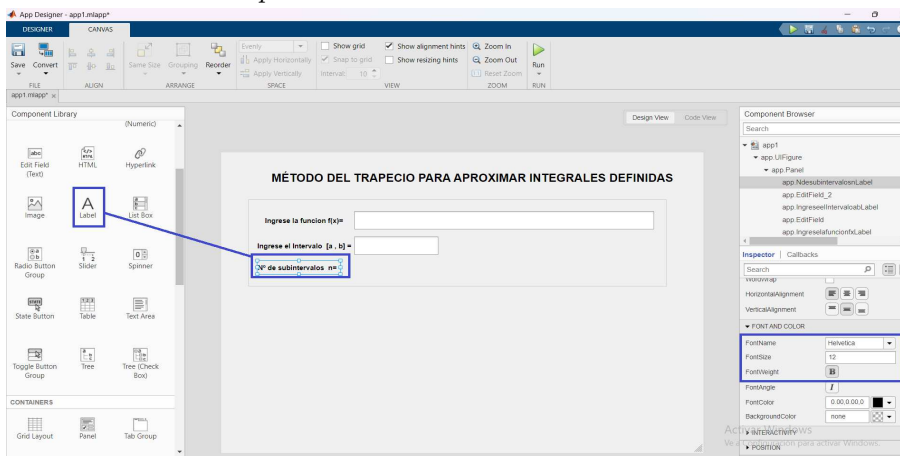
Edit Text Fiel, para ingresar $[a,b]$.



Paso 10: La Figura (13) muestra como se debe ingresar un Label para colocar el nombre de “Nº de subintervalos n=”.

Figura 13

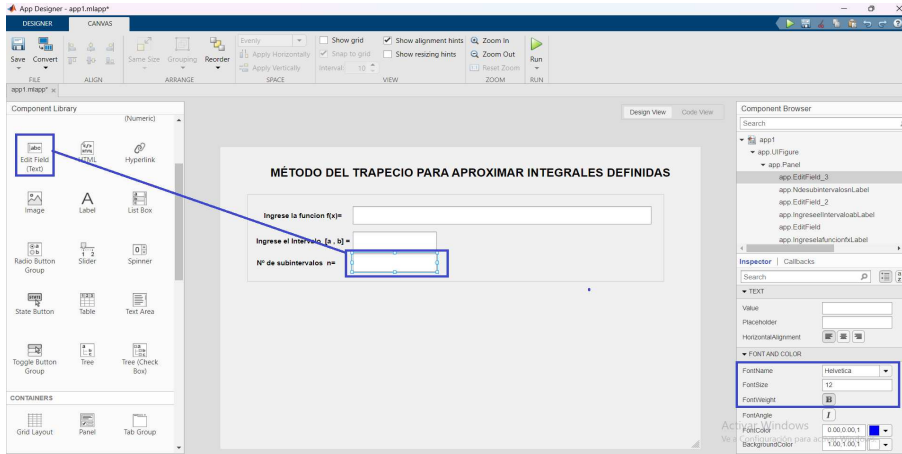
Se coloca un Label para “Nº de subintervalos n=”.



Paso 11: La Figura (14) muestra como se debe ingresar un Edit Text Fiel, para poder ingresar el número de subintervalos.

Figura 14

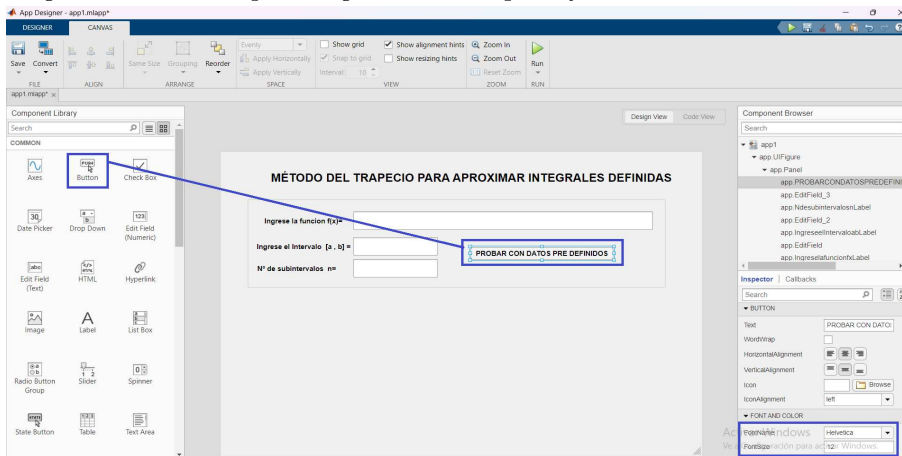
Edit Text Fiel, para ingresar el número de subintervalos.



Paso 12: La Figura (15) muestra como se debe ingresar un Pushbutton, para colocar el nombre de “PROBAR CON DATOS PRE DEFINIDOS”.

Figura 15

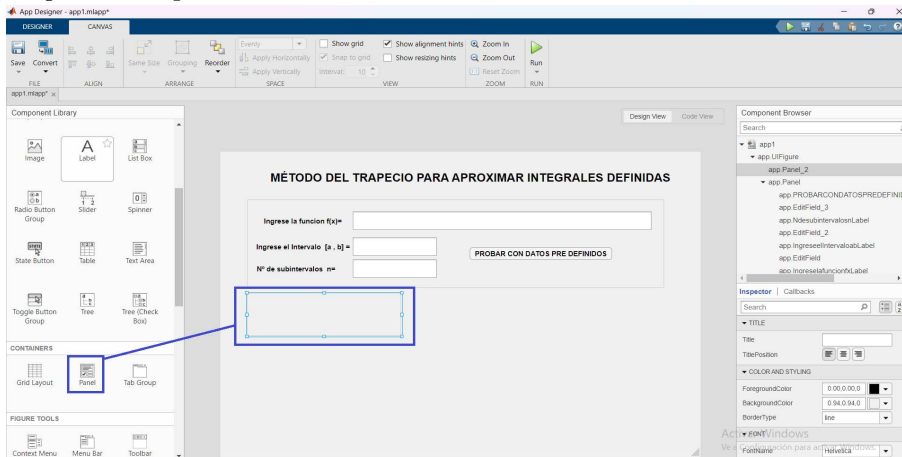
Ingresar un botón para ingresar datos predefinidos.



Paso 13: La Figura (16) muestra como se debe ingresar un panel 2

Figura 16

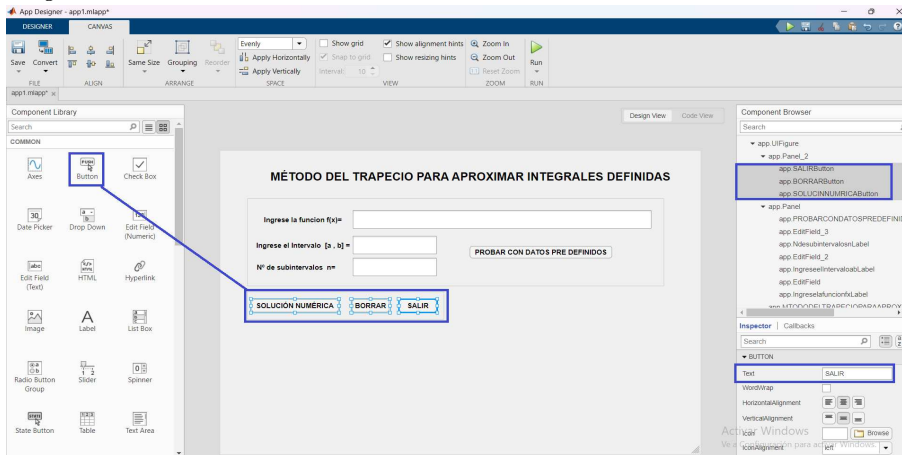
Ingresar un panel 2.



Paso 14: La Figura (17) muestra como se debe ingresar 3 Pushbutton para los botones SOLUCIÓN NUMÉRICA, BORRAR Y SALIR.

Figura 17

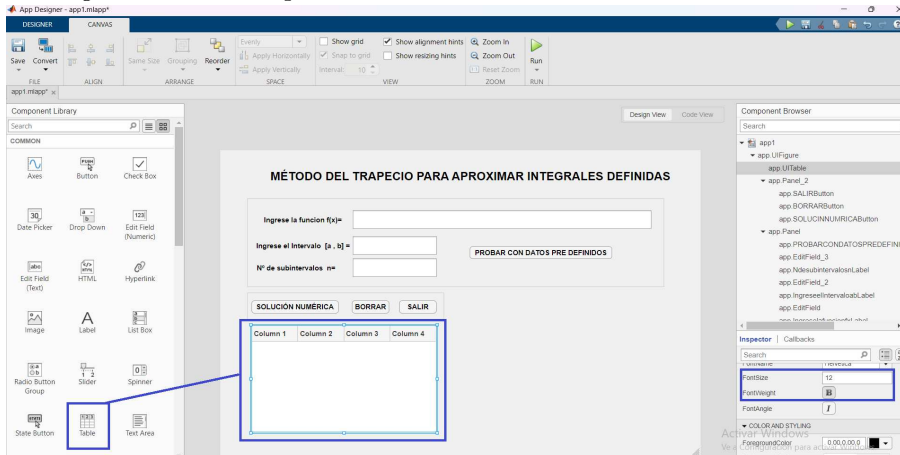
Ingresar 3 botones.



Paso 15: La Figura (18) muestra como se debe ingresar una Table para que se muestre los resultados de la función.

Figura 18

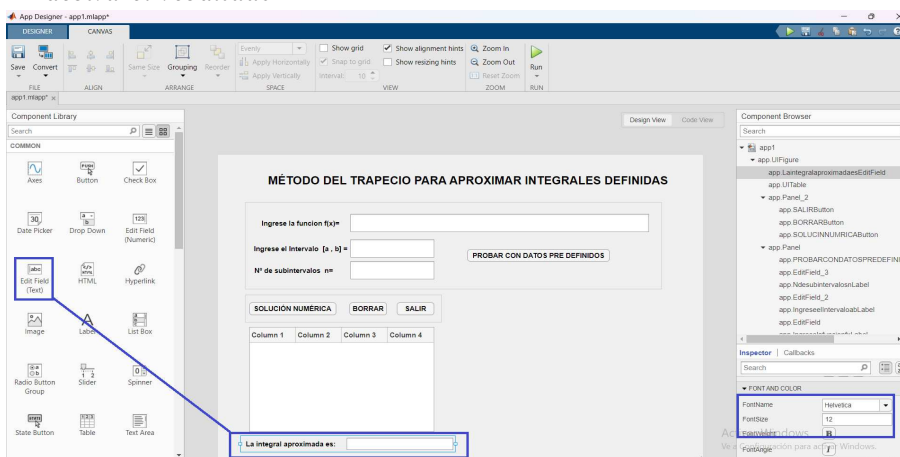
Se ingresa una tabla para mostrar resultados.



Paso 16: La Figura (19) muestra como se debe ingresar un Edit Text Field, para colocar el nombre de "La integral aproximada es:" y a la vez me muestre el resultado de la solución numérica del método del trapezio.

Figura 19

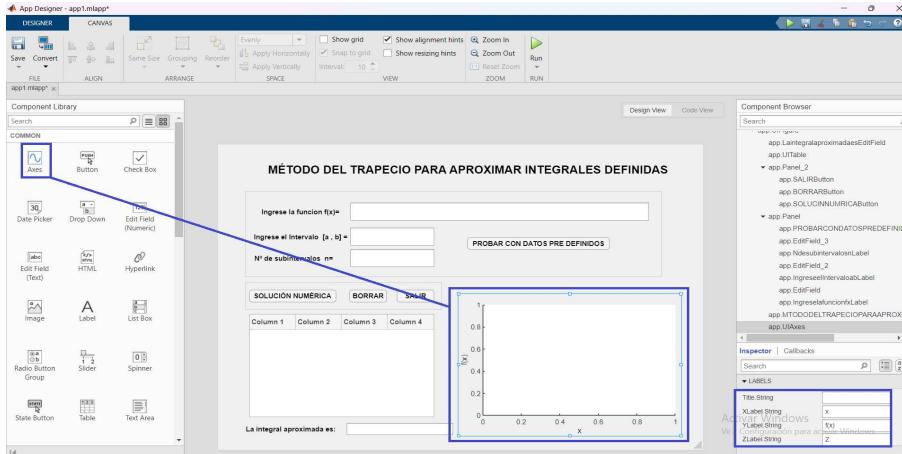
Muestra el resultado.



Paso 17: La Figura (20) muestra como se debe ingresar un Axes, para mostrar la gráfica de la función.

Figura 20

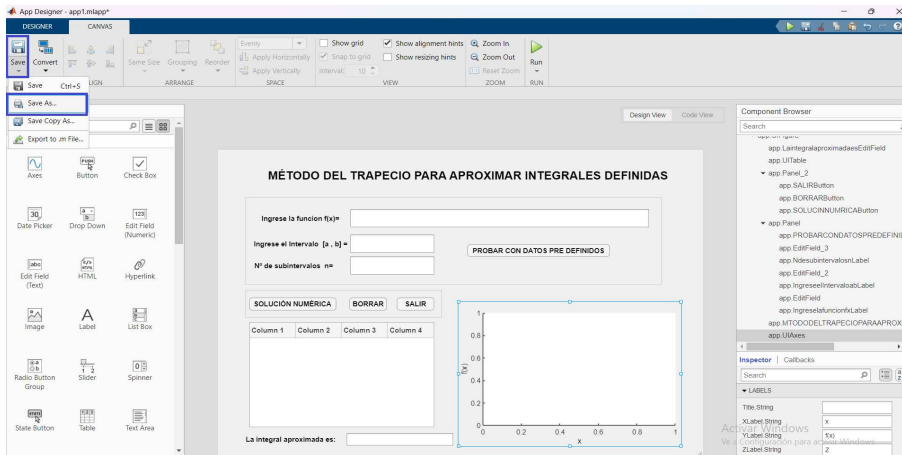
Se muestra la gráfica de la función.



Paso 18: La Figura (21) muestra como se procede a guardar la interfaz grafica de usuario en App designer.

Figura 21

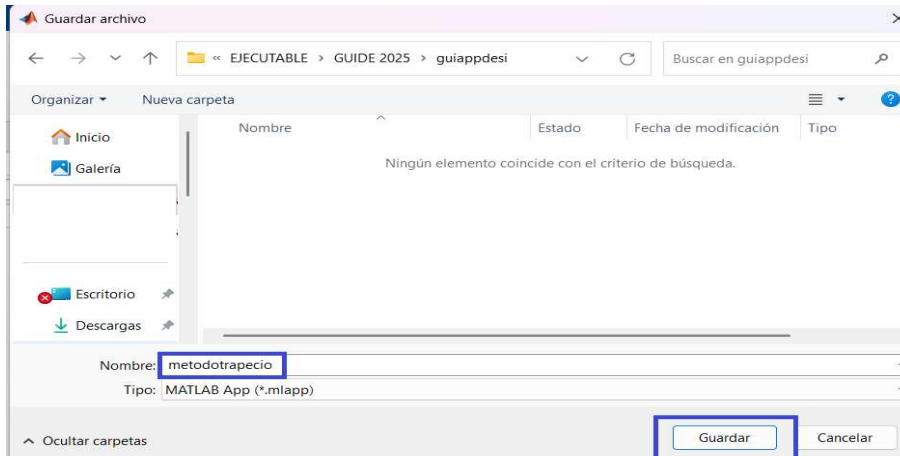
Guardar la GUI en App Designer.



Paso 19: La Figura (22) se guarda con el nombre de metodotrapecio y clic en guardar.

Figura 22

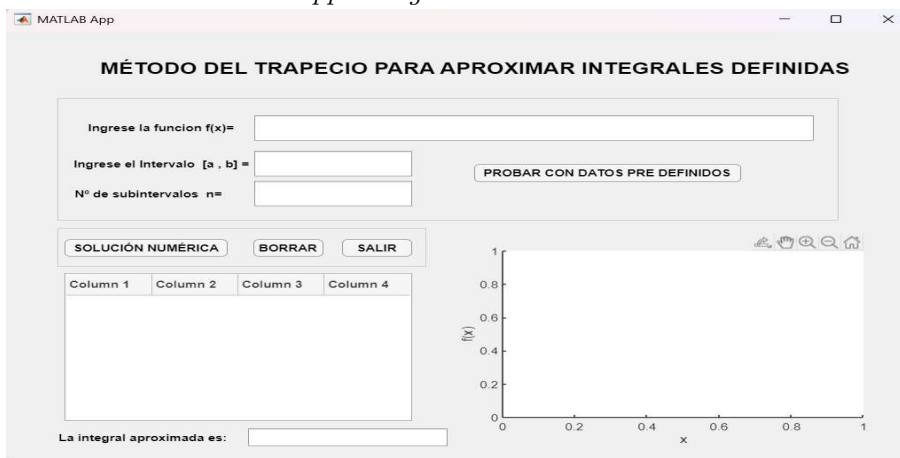
Colocar el nombre de la GUI en Designer.



Paso 20: La Figura (23) clic en el botón run para que muestre el diseño de la GUI en App Designer.

Figura 23

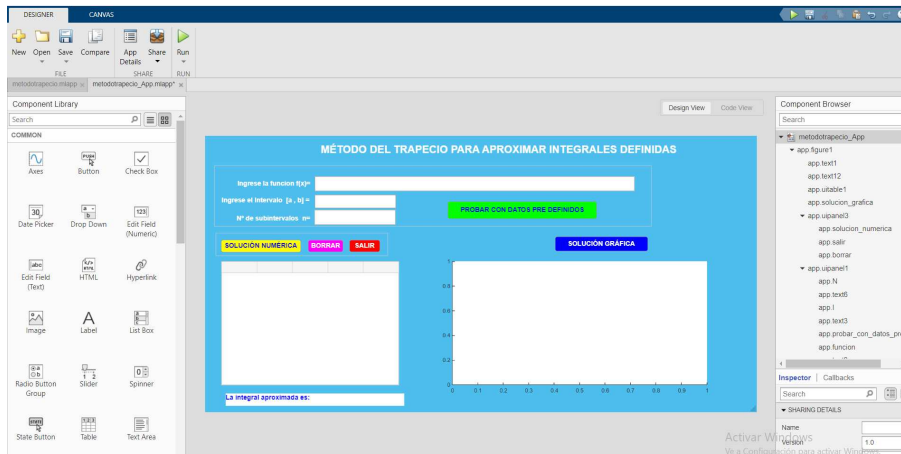
Diseño de la GUI en App Designer.



Paso 21: La Figura (24) diseño final de la GUI en App Designer.

Figura 24

Diseño final de la GUI en App Designer.



Descripción de la Figura (24) y su utilidad

La Figura (24) muestra una interfaz desarrollada para resolver integrales definidas utilizando el **método del trapecio**, un procedimiento numérico ampliamente usado para aproximar el área bajo una curva. Esta herramienta está diseñada con fines educativos y prácticos, facilitando el aprendizaje y la aplicación del método.

- **Ingreso de la función:** El usuario puede escribir cualquier función $f(x)$ que desee integrar.
- **Límites de integración:** Se deben ingresar los valores del intervalo $[a, b]$, donde se calculará la integral.
- **Subintervalos:** Se solicita la cantidad de divisiones n , necesarias para aplicar el método con mayor precisión.
- **Botón de solución numérica:** Al hacer clic en este botón, el sistema realiza automáticamente el cálculo y muestra el valor aproximado de la integral.

- **Visualización gráfica:** La interfaz ofrece una gráfica que representa la función y los trapecios generados, lo cual permite al usuario comprender mejor el proceso de aproximación.
- **Otros controles:** Incluye botones para borrar los datos ingresados, cerrar el programa o probar el método con datos predefinidos.

Esta aplicación sirve como una herramienta interactiva para el estudio del método del trapecio, siendo útil tanto en el aula como en proyectos de investigación aplicada.

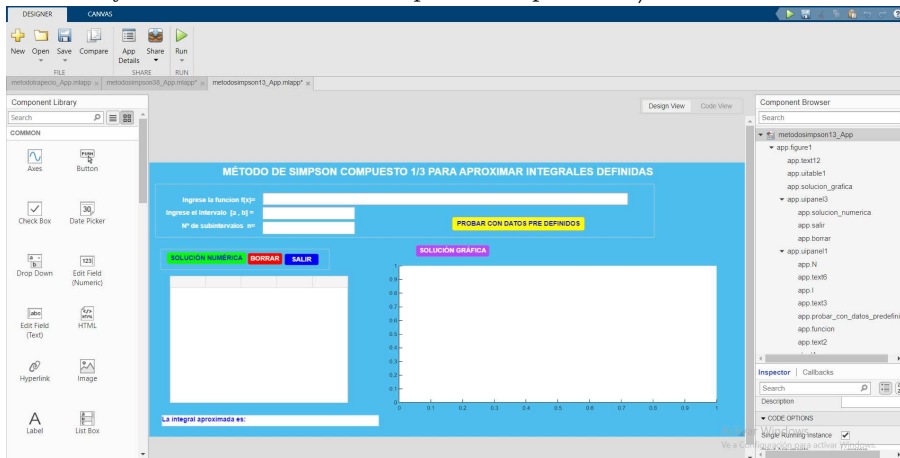
3.1.2 Diseño de la GUI del Método de Simpson 1/3

El proceso de diseño de la Interfaz Gráfica de Usuario (GUI) en App Designer de Matlab R2021a para implementar el método de Simpson 1/3 compuesto ha seguido una estructura similar a la utilizada para el método del trapecio, aprovechando la modularidad y la re-utilización de componentes del entorno visual. Ambos métodos comparten una lógica de entrada, procesamiento y salida muy parecida, lo que facilita la implementación paralela de sus algoritmos en una misma plataforma interactiva.

La Figura (25) muestra el diseño final de la GUI de App Designer del método de Simpson compuesto 1/3.

Figura 25

Diseño final del método de Simpson compuesto 1/3.



Descripción de la Figura (25) y su utilidad

La Figura (25) muestra una interfaz gráfica desarrollada en Matlab R2021a para aplicar el **método de Simpson 1/3 compuesto**, el cual se emplea para aproximar el valor de integrales definidas mediante un enfoque numérico.

- **Encabezado informativo:** En la parte superior de la ventana, se observa un título en letras blancas sobre fondo celeste que indica que la herramienta está diseñada para

aproximar integrales definidas utilizando el método de Simpson compuesto 1/3.

- **Campos de entrada:** El usuario dispone de tres cuadros de texto para ingresar los siguientes datos:
 - La función matemática $f(x)$ a integrar.
 - El intervalo de integración $[a, b]$.
 - El número de subintervalos n , que debe ser par, conforme a los requerimientos del método.
- **Botón de prueba automática:** A la derecha de los campos de entrada, aparece un botón amarillo que permite “PROBAR CON DATOS PRE DEFINIDOS”, facilitando el uso de la herramienta sin necesidad de escribir los valores manualmente.
- **Botones principales:** En la parte inferior izquierda se encuentran tres botones:
 - Verde: ejecuta la **solución numérica** y muestra el resultado aproximado.
 - Rojo: **borra** todos los datos introducidos.
 - Azul: permite **cerrar** el programa.
- **Gráfica del resultado:** En el lado derecho se observa un espacio reservado para la gráfica, donde se representa visualmente la función junto a las parábolas que aproximan el área bajo la curva.
- **Visualización de datos tabulados:** En la parte izquierda se incluye una tabla (inicialmente vacía) que probablemente muestra los valores utilizados y generados durante el proceso de integración.
- **Resultado final:** En la parte inferior central se muestra el campo “La integral aproximada es:”, seguido de un espacio donde aparece el valor obtenido por el método.

Esta herramienta tiene el objetivo de facilitar el cálculo de integrales definidas usando el método de Simpson 1/3 compuesto, combinando una interfaz amigable con una representación gráfica que refuerza el aprendizaje visual del procedimiento.

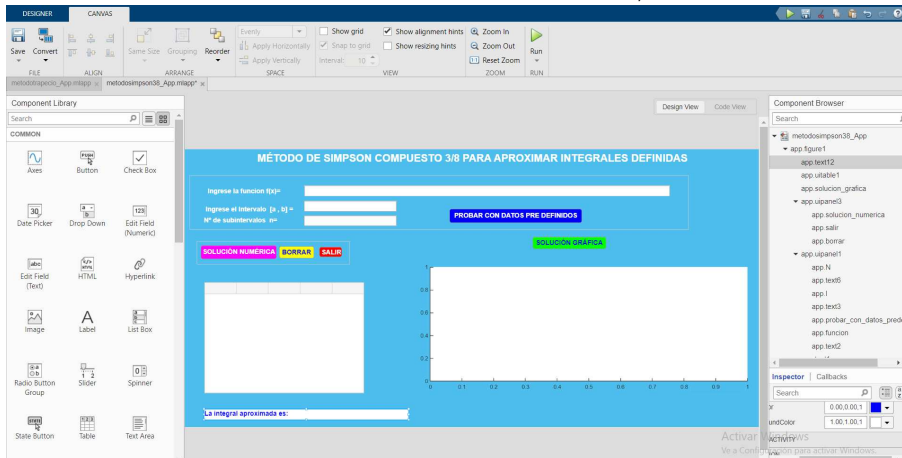
3.1.3 Diseño de la GUI del Método de Simpson 3/8

El proceso de diseño de la Interfaz Gráfica de Usuario (GUI) en App Designer de Matlab R2021a para el método de Simpson compuesto 3/8 se ha desarrollado utilizando la misma estructura funcional y visual implementada previamente para el método del trapecio. Esto se realizó con el objetivo de mantener una experiencia de usuario uniforme, aprovechar componentes reutilizables y facilitar la integración de múltiples métodos de integración numérica dentro de una sola aplicación.

La Figura (26) muestra el diseño final de la GUI de App Designer del método de Simpson compuesto 3/8.

Figura 26

Diseño final del método de Simpson compuesto 3/8.



Descripción de la Figura (26) y su utilidad

La Figura (26) presenta una interfaz gráfica diseñada en Matlab R2021a para resolver integrales definidas aplicando el **método de Simpson 3/8 compuesto**, el cual permite aproximar el valor de un área bajo la curva mediante la combinación de polinomios cúbicos.

- **Título principal:** En la parte superior se encuentra el texto “MÉTODO DE SIMPSON COMPUESTO 3/8 PARA APROXIMAR INTEGRALES DEFINIDAS”, es-

crito en letras blancas sobre un fondo celeste que destaca la finalidad del aplicativo.

- **Ingreso de datos:** Justo debajo del título, la interfaz ofrece tres campos para que el usuario pueda escribir:
 - La función matemática $f(x)$.
 - El intervalo de integración definido por los valores a y b .
 - El número de subintervalos n , que debe ser múltiplo de tres para que el método funcione correctamente.

- **Botón de datos predefinidos:** A la derecha, se encuentra un botón azul con el texto “PROBAR CON DATOS PRE DEFINIDOS”, el cual permite al usuario trabajar directamente con un ejemplo sin necesidad de llenar los campos manualmente.

- **Controles principales:** En la parte inferior izquierda hay tres botones bien diferenciados:
 - Rosado: activa la **solución numérica**, que realiza el cálculo aproximado de la integral.
 - Amarillo: permite **borrar** los datos ingresados.
 - Rojo: sirve para **cerrar** la aplicación.

- **Zona gráfica:** En el lado derecho de la interfaz, hay un área destinada a mostrar una representación gráfica de la función y de los segmentos aplicados por el método de Simpson 3/8.

- **Tabla de datos:** A la izquierda de la gráfica se encuentra un espacio donde se mostrará una tabla con los valores numéricos utilizados en el cálculo.

- **Resultado final:** En la parte inferior, un texto en azul claro indica: “La integral aproximada es:”, seguido de un recuadro donde aparecerá el resultado obtenido.

Esta aplicación permite al usuario ejecutar de forma visual e interactiva el método de Simpson 3/8 compuesto. Es una herramienta ideal para docentes, estudiantes e investigadores que deseen experimentar y comprender este proceso de integración numérica.

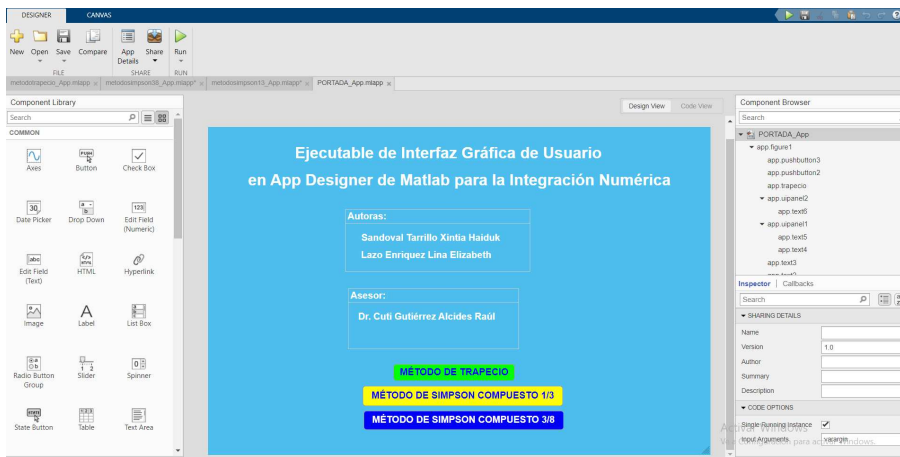
3.1.3 Diseño de la Portada

La portada de la aplicación desarrollada en App Designer de MATLAB fue diseñada con el propósito de ofrecer una presentación clara, profesional e intuitiva sobre nuestra investigación titulada “Ejecutable de Interfaz Gráfica de Usuario en App Designer de Matlab R2021a para la Integración Numérica”. Esta pantalla de bienvenida cumple una doble función: presentar el objetivo principal de la aplicación y servir como menú principal de navegación hacia los diferentes métodos de integración implementados.

La Figura (27) muestra la portada de nuestra investigación.

Figura 27

Portada.



3.1.4 Convertir a Ejecutable

Después de terminar el diseño completo de la aplicación en App Designer, se llevó a cabo la creación de un archivo ejecutable para que pueda usarse en cualquier computadora sin necesidad de tener Matlab R2021a instalado. Para eso, se usó la herramienta que Matlab R2021a ofrece llamada MATLAB Compiler, a la cual se accedió desde el comando `deploytool`.

Se eligió como archivo principal el que contiene la portada de la aplicación (`PORTADA_App.mlapp`), ya que desde ahí el usuario puede acceder a los tres métodos desarrollados: el método del trapecio, Simpson 1/3 y Simpson 3/8. Por esta razón, también se agregaron los archivos de cada uno de estos métodos para que el ejecutable funcione correctamente.

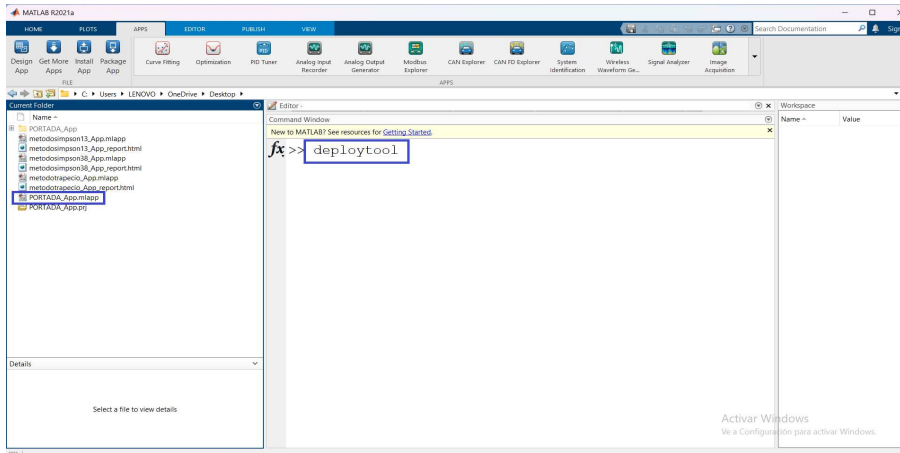
En el proceso de configuración, se colocó el nombre de la aplicación, se agregaron los archivos necesarios y se generó un instalador que incluye todo lo que se necesita para ejecutarla. Este instalador también permite que el usuario descargue e instale automáticamente el MATLAB Runtime, que es un programa gratuito que reemplaza a Matlab R2021a para poder abrir este tipo de aplicaciones.

Con este procedimiento, se logró obtener una versión del proyecto que funciona de forma independiente, permitiendo su uso en otros equipos de forma sencilla y sin depender del software original. Además, se conservó la estructura de la aplicación con su diseño gráfico, botones y resultados, tal como se desarrolló dentro del entorno de Matlab R2021a.

Paso 1: La Figura (28) se observa que se ingresa la palabra `deploytool` en el command window.

Figura 28

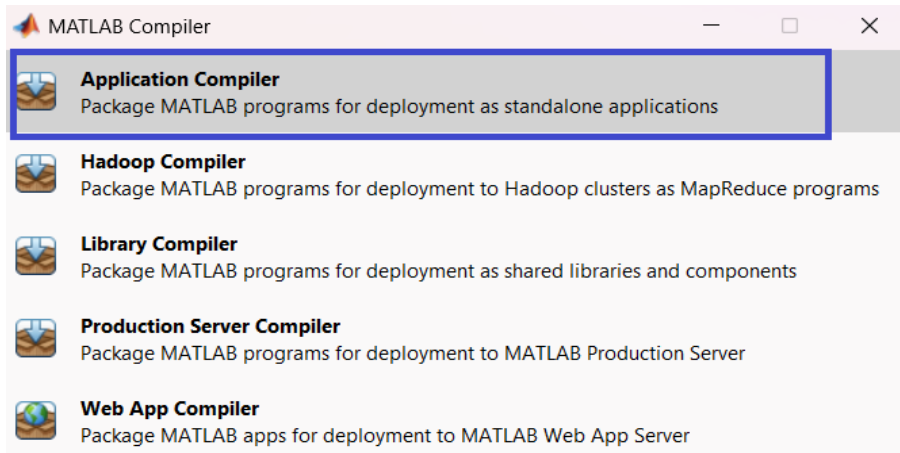
Deploytool.



Paso 2: La Figura (29) muestra que se debe elegir Application Compiler.

Figura 29

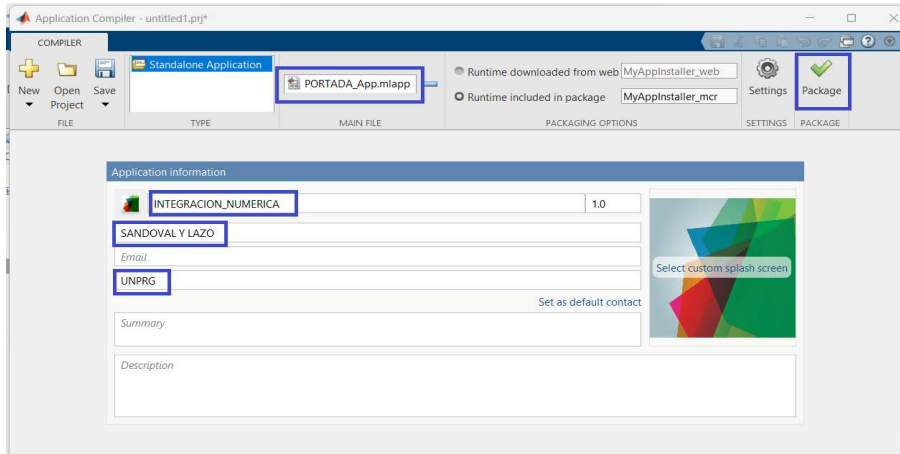
Application compiler.



Paso 3: La Figura (30) muestra como se llenan los datos y se agrega la portada cargando los métodos para que se conviertan en ejecutable dando clic en Package.

Figura 30

Cargar datos.



Paso 4: La Figura (31), muestra que se han activado los 4 chek, lo cual significa que se ha convertido en ejecutable.

Figura 31

Activación de los 4 chek.



3.2 Aplicación 1

Ejemplo 4. Resolver la siguiente integral

$$\int_0^2 x^3 e^{x^2} dx$$

Solución.

La siguiente integral a sido solucionada de forma analítica y de forma numérica por los métodos de Trapecio, Simpson compuesto 1/3 y Simpson compuesto 3/8. Ahora utilizaremos el ejecutable para comparar los resultados obtenidos de forma iterativa.

Método del Trapecio

Figura 32

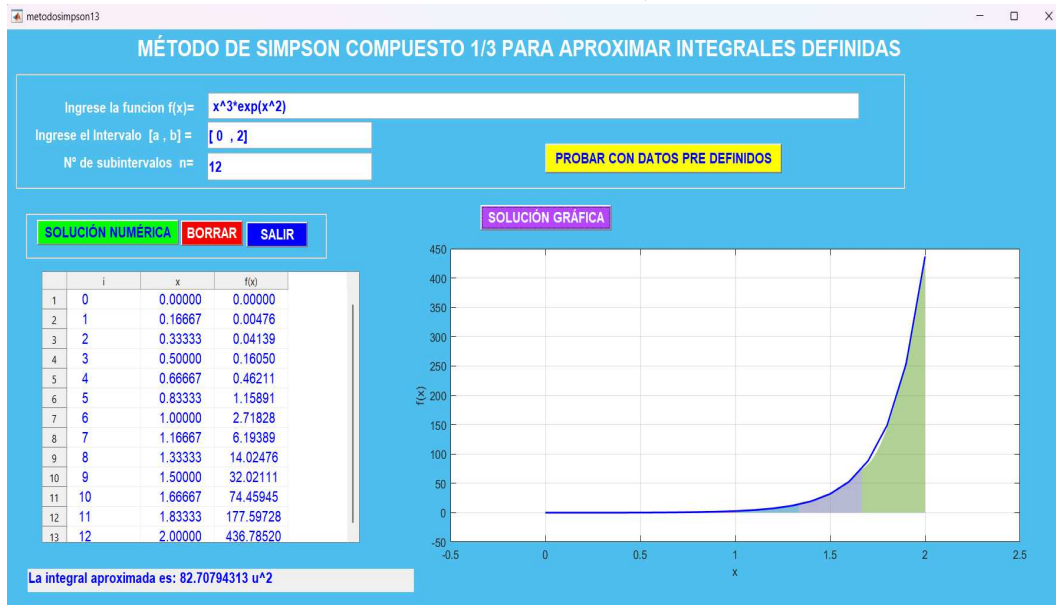
Resultado de la aplicación 1 método del Trapecio, para $n=12$.



Método de Simpson compuesto 1/3

Figura 33

Resultado de la aplicación 1 método de Simpson 1/3, para $n=12$.



Método de Simpson compuesto 3/8

Figura 34

Resultado de la aplicación 1 método de Simpson 3/8, para $n=12$.

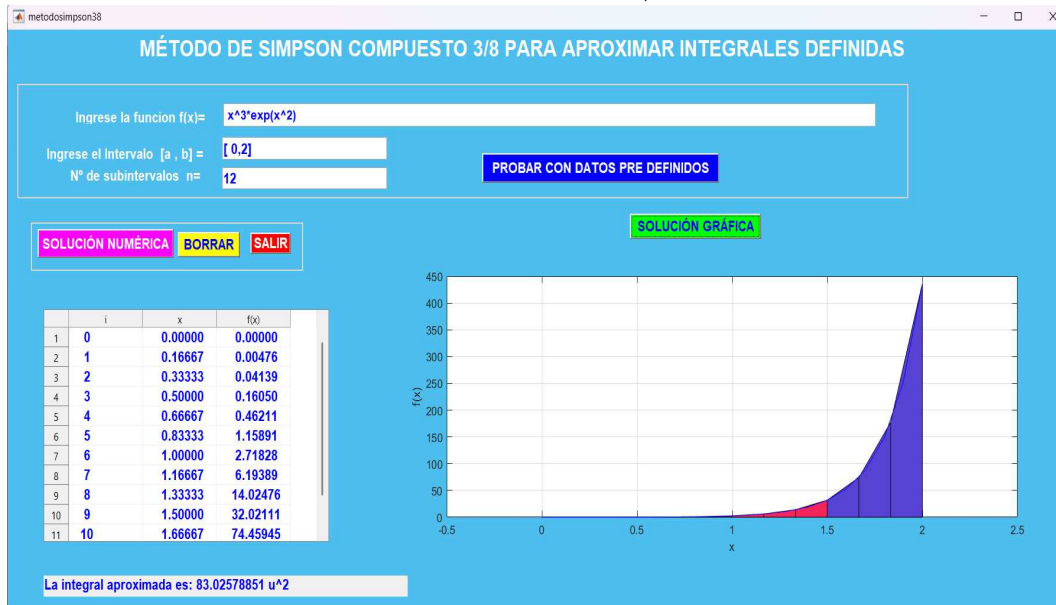


Tabla 4*Comparando los resultados de la aplicación 1, para $n=12$*

n	Solución analítica	Trapezio	Simpson compuesto 1/3	Simpson compuesto 3/8
12	82.39722505	87.87250609739430	82.70794313476460	83.025788510454
		errotrapecio	errors1/3	errors3/8
		5.47528104739429	0.3107180847646020	0.62856346045400

De la Tabla (4) y de las Figuras (32), (33) y (35), al analizar los resultados obtenidos en la Aplicación 1 para $n = 12$, se puede observar que el valor de la solución analítica es de 82.39722505. El método del Trapecio arrojó un resultado de 87.87250609739430, lo cual generó un error absoluto de 5.47528104739429, siendo este el mayor error entre los tres métodos evaluados. Esto demuestra que, si bien este método es fácil de implementar, su precisión disminuye considerablemente al trabajar con un número moderado de subintervalos.

En cambio, el método de Simpson compuesto 1/3 generó un valor de 82.70794313476460, con un error de solo 0.3107180847646020, lo que lo convierte en el método con mayor precisión y más cercano a la solución analítica. Por su parte, el método de Simpson compuesto 3/8 ofreció un resultado de 83.02578851045400, cuyo error fue de 0.62856346045400, lo que representa una mejora respecto al método del trapecio, pero sigue siendo más impreciso que el método de Simpson 1/3.

Con base en estos resultados, se concluye que el método de Simpson compuesto 1/3 es el que proporciona la mejor aproximación al valor real de la integral para $n = 12$. Esto justifica su elección como el método más adecuado para ser implementado en el aplicativo desarrollado.

3.3 Aplicación 2

Debasish y Sanjit (2024), plantearon la siguiente integral

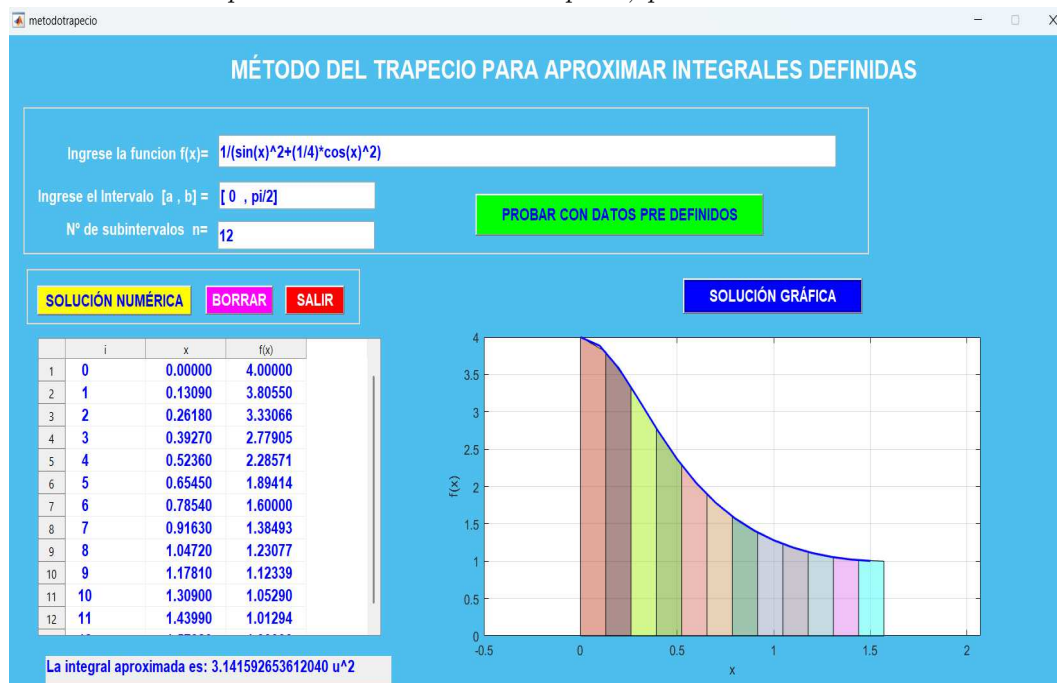
$$\int_0^{\frac{\pi}{2}} \frac{1}{\sin^2 x + \frac{1}{4} \cos^2 x} dx$$

cuya solución analítica es: $\pi \cong 3.14159265358979$ La siguiente integral se va a solucionar mediante el ejecutable, para $n=12$.

Método del Trapecio

Figura 35

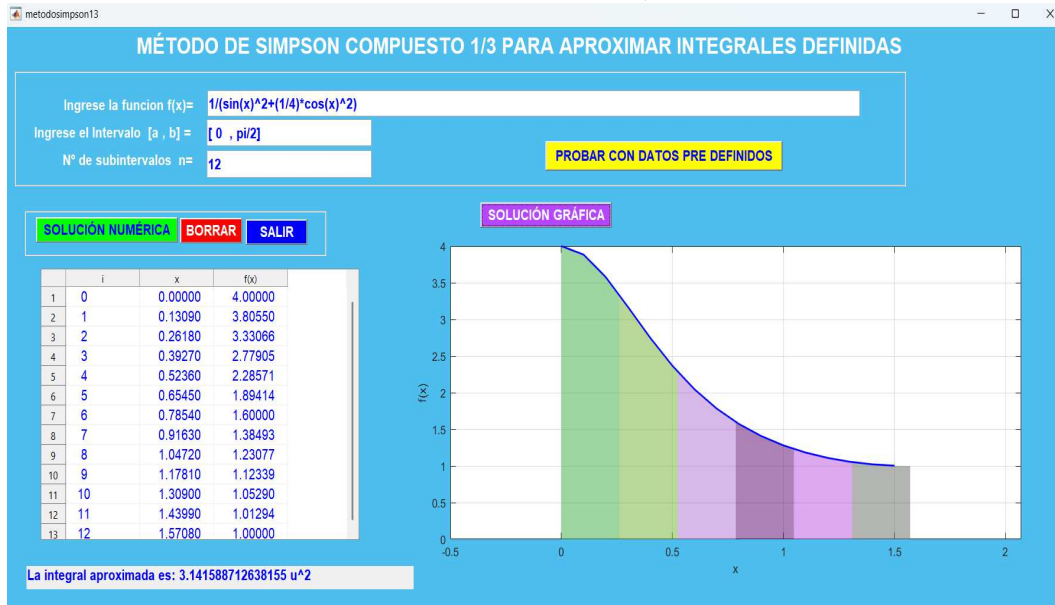
Resultado de la aplicación 2 método del Trapecio, para $n=12$.



Método de Simpson compuesto 1/3

Figura 36

Resultado de la aplicación 2 método de Simpson 1/3, para $n=12$.



Método de Simpson compuesto 3/8

Figura 37

Resultado de la aplicación 2 método de Simpson 3/8, para $n=12$.

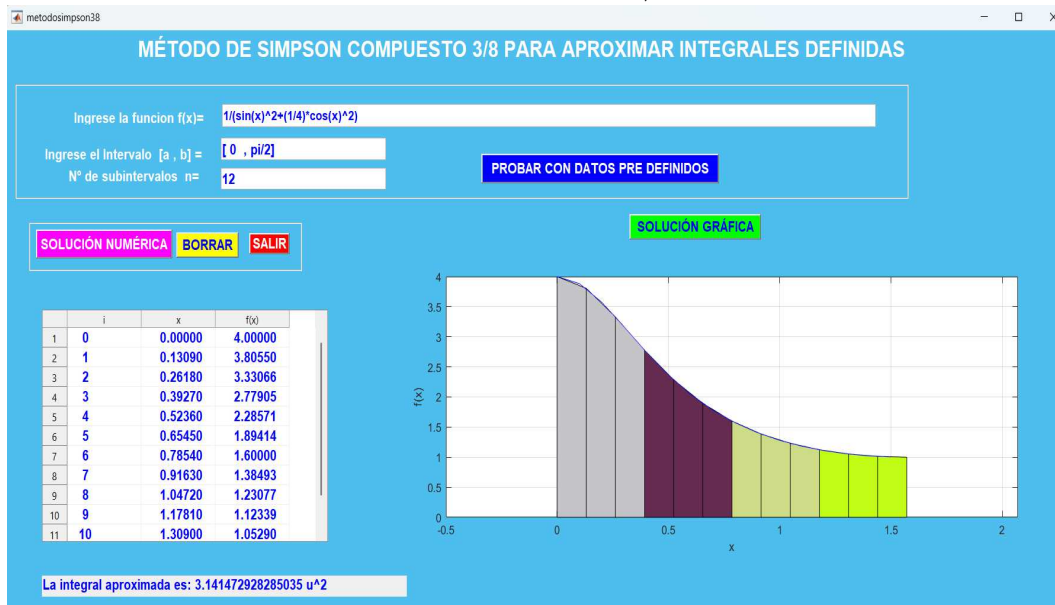


Tabla 5*Comparando los resultados de la aplicación 2, para $n=12$*

n	Solución analítica	Trapecio	Simpson compuesto 1/3	Simpson compuesto 3/8	$Q_{S_3}(f)$	$Q_{aS_4}(f)$
12	3.14159265358979	3.14159265361204	3.14158871263815	3.14147292828503	2.98451302090592	3.30078360295501
		error	errors1/3	errors3/8	error$Q_{S_3}(f)$	error$Q_{aS_4}(f)$
		0.0000000002225	0.0000039410	0.000119725305	0.157079632684	0.159190949365

De la Tabla (5) y de las Figuras (35), (36) y (37), se concluyó que el método del trapecio fue el más preciso en esta aplicación específica con $n = 12$, superando incluso a métodos más complejos. Aunque el método de Simpson 1/3 también presentó una muy buena aproximación, el trapecio ofreció el menor error absoluto, lo cual demuestra su eficiencia y estabilidad para determinados tipos de funciones y particiones. Por tanto, se recomienda destacar el método del trapecio como el más efectivo para esta implementación.

3.4 Aplicación 3

Tineo y Torres (2022) plantearon la siguiente integral

$$\int_0^1 e^{x^2} dx$$

Plantearon su solución utilizando su GUI y obtuvieron el siguiente resultado.

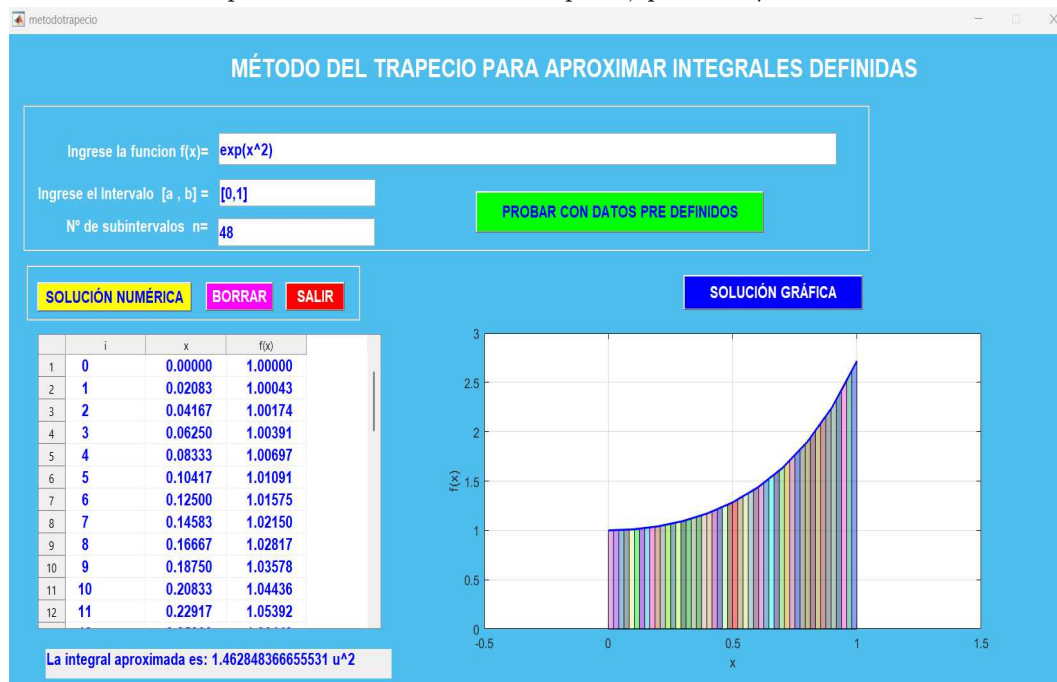
$$\int_0^1 e^{x^2} dx = -\frac{1}{2} * I * \operatorname{erf}(I) * \sqrt{\pi} = 1.462651746$$

Ahora para poder validar nuestro ejecutable, vamos a comparar el resultado para $n=48$:

Método del Trapecio

Figura 38

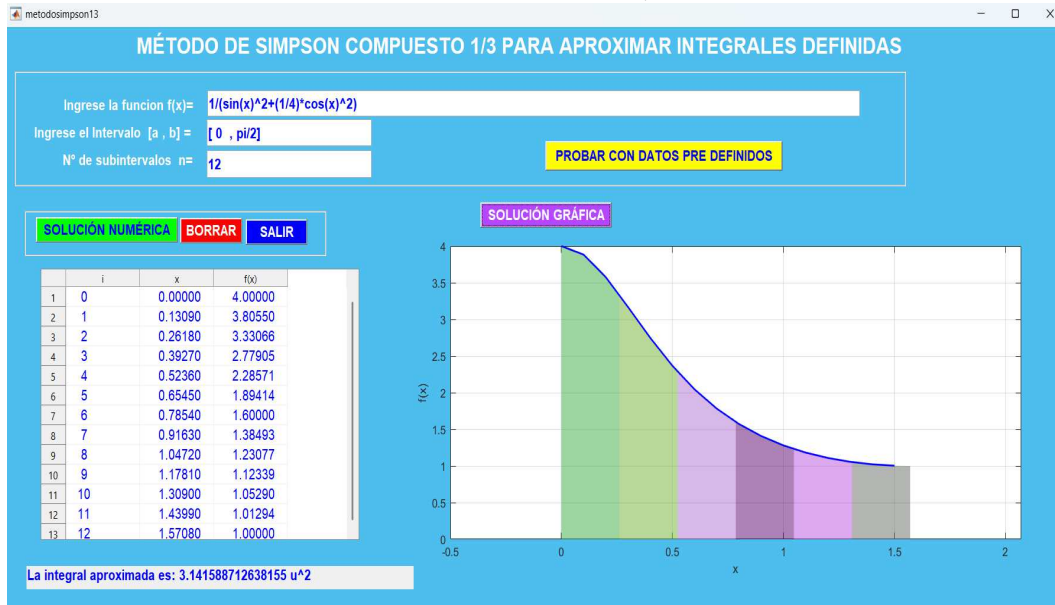
Resultado de la aplicación 3 método del Trapecio, para $n=48$.



Método de Simpson compuesto 1/3

Figura 39

Resultado de la aplicación 2 método de Simpson 1/3, para $n=48$.



Método de Simpson compuesto 3/8

Figura 40

Resultado de la aplicación 2 método de Simpson 3/8, para $n=48$.

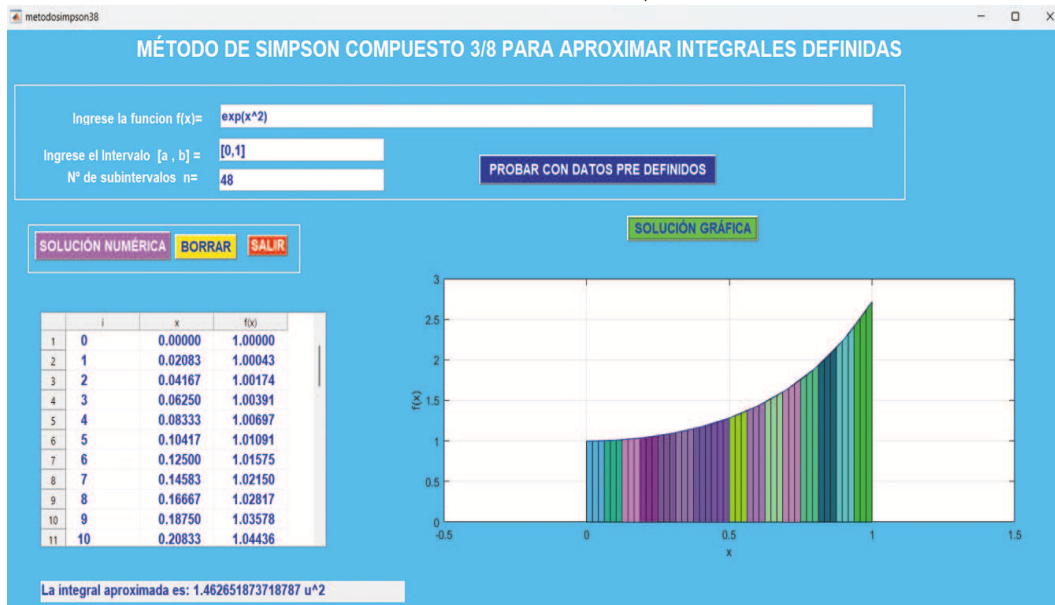


Tabla 6*Comparando los resultados de la aplicación 3, para $n=48$*

n	Solución analítica	Trapezio	Simpson compuesto 1/3	Simpson compuesto 3/8
48	1.462651746	1.46284836665553	1.46265180275805	1.462651874
		errortrapecio	errors1/3	errors3/8
		0.00019662065553	0.0000000567580500	0.00000012771878

De la Tabla (6), Figuras (38), (39) y (40), al comparar los resultados obtenidos con los métodos numéricos implementados en el ejecutable, se identificó que el valor más próximo a la solución analítica fue generado por el método de Simpson compuesto 1/3, seguido muy de cerca por el método de Simpson compuesto 3/8. En contraste, el método del trapezio presentó un error considerablemente mayor. Esto evidencia que los métodos de Simpson ofrecen una mayor precisión para el caso analizado con $n = 48$ subintervalos.

Además, los resultados encontrados en Tineo y Torres (2022) coinciden con los obtenidos en esta investigación, lo que permite validar el correcto funcionamiento del ejecutable desarrollado. Es importante señalar que, dentro de los algoritmos internos, se consideró un valor de $m = 24$ para el método de Simpson compuesto 1/3 y $m = 16$ para el de Simpson compuesto 3/8, asegurando una distribución adecuada de los subintervalos para el cálculo numérico.

Con base en esta comparación, se concluye que el método de Simpson compuesto 1/3 es el más preciso y eficiente, siendo la mejor opción para su aplicación dentro del desarrollo de este trabajo.

IV. DISCUSIÓN DE RESULTADOS

Analizar los métodos numéricos del trapecio, Simpson 1/3 y Simpson 3/8

Los métodos fueron evaluados en tres aplicaciones distintas para determinar su precisión. En la primera aplicación, se resolvió la integral $\int_0^2 x^3 e^{x^2} dx$, donde el método del trapecio presentó el mayor error (5.47528), mientras que Simpson 1/3 mostró el menor (0.31071), seguido por Simpson 3/8 (0.62856). Esto demostró que el método de Simpson 1/3 ofreció una aproximación mucho más cercana a la solución exacta.

En la segunda aplicación, al resolver la integral propuesta por Debasish y Sanjit (2024), $\int_0^{\pi/2} \frac{1}{\sin^2 x + \frac{1}{4} \cos^2 x} dx$, cuya solución analítica es π , el método del trapecio sorprendió con el menor error absoluto (2.225 e-11), superando a Simpson 1/3 y 3/8. Este caso confirma lo expuesto por Guo (2023), quien señaló que el método del trapecio puede ser muy eficiente en funciones específicas y con ciertas configuraciones de subintervalos.

En la tercera aplicación, con la integral $\int_0^1 e^{x^2} dx$, la solución analítica propuesta por Tineo y Torres (2022) fue 1.462651746. Aquí, el método de Simpson 1/3 tuvo el mejor rendimiento (error: 0.0000000567), seguido muy de cerca por Simpson 3/8 (error: 0.0000001277), mientras que el trapecio presentó un error de 0.0001966207. Esto se alinea con estudios como el de Md. et al. (2019), que identificaron a Simpson 1/3 como el más preciso ante funciones no lineales o suavemente crecientes.

Así, al comparar los tres métodos, se concluye que el método de Simpson 1/3 fue, en promedio, el más preciso, validando lo afirmado por Ahmed et al. (2021), quienes observaron que este método presenta una mejor relación entre exactitud y esfuerzo computacional. La evidencia sugiere que, si bien cada método puede tener ventajas en situaciones específicas, Simpson 1/3 es el más confiable para integrales definidas con funciones suaves y particiones adecuadas.

Diseñar y programar la interfaz gráfica de usuario en App Designer de Matlab R2021a para cada método de integración

Se logró construir una interfaz intuitiva que permite al usuario ingresar la función, los límites de integración y el número de subintervalos. El diseño facilitó la visualización del procedimiento numérico, gráfico y los resultados obtenidos. Esto representa una mejora significativa respecto a lo reportado por Solano (2023), quien usó plataformas educativas para reforzar la comprensión, pero no integró el desarrollo de una herramienta visual autónoma para cálculos numéricos.

Convertir en un ejecutable la interfaz gráfica de usuario en App Designer de Matlab R2021a para cada método de integración

Una de las principales contribuciones de esta investigación fue la transformación de la interfaz en un archivo ejecutable independiente del entorno Matlab R2021a, lo cual no había sido alcanzado por Tineo y Torres (2022). En su estudio, ellos desarrollaron una GUI funcional en Matlab R2021a, pero recomendaron la necesidad de crear un ejecutable para ampliar el acceso a usuarios sin licencia de Matlab R2021a. En esta tesis, se logró cumplir con esa recomendación, convirtiendo la interfaz en un recurso autónomo y accesible para entornos educativos y profesionales, mejorando así la aplicabilidad del recurso creado.

Identificar qué método tiene mejor aproximación

Los resultados muestran que Simpson 1/3 ofreció la mejor aproximación general en dos de las tres aplicaciones, con errores mínimos incluso con un número moderado de subintervalos. Esto respalda la propuesta de Eidous y Ananbeh (2024), quienes al aplicar reglas como la de Simpson en estimaciones integrales obtuvieron sesgos más bajos y mayor eficiencia. Asimismo, este hallazgo coincide con lo descrito por Aflook y Al-Rammahi (2021), quienes mejoraron Simpson 3/8 con aceleraciones exponenciales, lo que sugiere que la familia de

métodos de Simpson es la más sólida numéricamente.

Utilizar el ejecutable para resolver las integrales definidas sin necesidad de tener instalado MATLAB

Gracias a la conversión del entorno gráfico a ejecutable, los usuarios podrán resolver las integrales definidas en las tres aplicaciones usando solo el archivo ejecutable, sin requerir instalación adicional. Esta innovación fortalece el uso de métodos numéricos en aulas de ingeniería y matemáticas, tal como se promueve en el trabajo de Uddin et al. (2019), quienes destacaron la importancia del uso de software interactivo (como GeoGebra) para comprender visualmente la integración. En este caso, se combinó el poder de cálculo numérico con la visualización dinámica sin depender del entorno Matlab R2021a.

V. CONCLUSIONES

1. En la presente investigación se analizaron 3 aplicaciones, en la primera y tercera , se concluyó que el método de Simpson 1/3 es el más preciso, al obtener resultados muy cercanos a la solución analítica, superando a los métodos del trapecio y Simpson 3/8. Esto evidenció su efectividad al trabajar con funciones exponenciales y polinómicas. En la aplicación 2, el método del trapecio logró una aproximación casi exacta al valor de π con solo 12 subintervalos, mostrando su eficiencia en funciones racionales. Sin embargo, al incrementar los subintervalos a 36, el método de Simpson 1/3 igualó la precisión alcanzada por el trapecio en dicha aplicación. Estos resultados permitieron concluir que Simpson compuesto 1/3 fue el método más preciso en la mayoría de los casos evaluados, especialmente cuando se emplea una mayor cantidad de particiones. Finalmente, se determinó que la elección del método depende del tipo de función, pero Simpson compuesto 1/3 representa la mejor opción cuando se requiere mayor exactitud.
2. Se diseñó una interfaz gráfica funcional que permite al usuario ingresar de manera sencilla la función a integrar, el intervalo $[a, b]$ y el número de subintervalos n . El aplicativo presenta los resultados de manera clara, tanto en forma de tabla como mediante gráficos segmentados por colores, lo cual favorece la comprensión visual del proceso de integración. Además, se programaron validaciones internas que aseguran el uso correcto de cada método. Por ejemplo, el método de Simpson 1/3 solo acepta valores pares de n y, en caso contrario, muestra el mensaje: “Para aplicar Simpson 1/3, n debe ser un número par”. De forma similar, el método de Simpson 3/8 exige que n sea múltiplo de 3 , y si no se cumple esta condición, aparece la advertencia: “El valor de n debe ser múltiplo de 3 para aplicar el método de Simpson 3/8”. Estas validaciones automáticas permiten evitar errores de ejecución y educan al usuario en el uso adecuado de los métodos numéricos. Además, una de las principales mejoras logradas fue la creación de un ejecutable independiente, lo que permitió que el pro-

grama se ejecute sin necesidad de contar con Matlab R2021a instalado. Esto amplía las posibilidades de uso, tanto en entornos educativos como profesionales, facilitando el acceso a una herramienta didáctica completa y práctica.

3. El ejecutable permitió resolver diversas integrales definidas de forma rápida, precisa y visual, mostrando en pantalla tanto los resultados numéricos como gráficos. Las validaciones internas ayudaron a evitar errores comunes en la selección de subintervalos, y los mensajes informativos orientaron al usuario sobre cómo corregirlos. Gracias a su independencia del entorno Matlab R2021a, el ejecutable puede utilizarse en distintas computadoras sin restricciones técnicas, lo que lo convierte en una herramienta educativa útil, didáctica y de fácil acceso para el aprendizaje de los métodos de integración numérica.

RECOMENDACIONES

Se recomienda seguir desarrollando ejecutables educativos similares en otras áreas del cálculo numérico, como derivación, interpolación o resolución de ecuaciones diferenciales. Asimismo, se sugiere implementar mejoras en el aplicativo creado, como la exportación automática de resultados y gráficos en formato PDF o Excel, lo que permitiría ampliar su utilidad práctica. También sería beneficioso adaptar la herramienta a plataformas móviles o web, con el fin de facilitar el acceso desde distintos dispositivos. Finalmente, se aconseja integrar este recurso en cursos universitarios, promoviendo el uso de herramientas visuales e interactivas que fortalezcan el aprendizaje autónomo.

Referencias Bibliográficas

- Aflook, S., & Al-Rammahi, A. (2021). Exponential Acceleration To Improve Simpson's 3/8 rule. *Al Qadisiyah Journal of Pure Science*, 26(1), 104-111. <https://doi.org/10.29350/qjps.2021.26.1.1240>
- Ahmed, T., Islam, T., Abdullah, A., & Md, B. (2021). The Periodicity of the Accuracy of Numerical Integration Methods for the Solution of Different Engineering Problems. *ResearchGate*, 2(4), 203-216. <https://doi.org/10.38032/jea.2021.04.006>
- Burden, R. & Douglas, J. (2002). *Análisis Numérico*. 9^a ed. Cengage.
- Debasish, D., & Sanjit, M. (2024). Anti-Simpson's quadrature formula and its extension for evaluation of elliptic and other integrals in adaptive environment. *South East Asian Journal of Mathematics and Mathematical Sciences*, 20(02), 463-476. <https://doi.org/10.56827/SEAJMMS.2024.2002.33>
- Eidous, O., & Ananbeh, E. (2024). Kernel method for estimating overlapping coefficient using numerical integration methods. *Applied Mathematics and Computation*, 462, 128339. <https://doi.org/10.1016/j.amc.2023.128339>
- Guo, W. (2023). Solving problems involving numerical integration (I): Incorporating different techniques. *STEM Education*, 3(2), Article steme-03-02-009. <https://doi.org/10.3934/steme.2023009>
- Kharab, A. & Guenther, R. (2012). *An introduction to numerical methods a Matlab approach*. CRC Pres, New York, USA.
- Madrid, P. (2023). Acerca del Orden de Convergencia de las Reglas de Integración del Trapecio y Simpson para Cierta Clase de Funciones No Diferenciables. *Revista de la Escuela de Física*, 11(1), Article 1. <https://doi.org/10.5377/ref.v11i1.16823>
- Mallma, R., Montoya, O., Huanca, A., & Figueroa, H. (2022). A Measurement Application to Surfaces using Non Linear Interpolation and Numerical Integration Methods. *International Journal of Engineering Trends and Technology*, 70(6), 237-251. <https://doi.org/10.14445/22315381/IJETT-V70I6P225>
- MathWorks, (2024). *GUI de MATLAB*. Recuperado de <https://n9.cl/2m35g>

- Md., N., Mohammad, F., & Sadiya, U. (2019). Comparison on Trapezoidal and Simpson's Rule for Unequal Data Space. ResearchGate, 4(1). <https://doi.org/10.5815/ijmsc.2019.04.03>
- Navarro, I. (2020). *Análisis de Métodos de Integración Numérica para Problemas Dinámicos. Implementación y Validación Mediante una Aplicación de Matlab* [Tesis de maestría, Universidad Politécnica de Valencia]. Repositorio Institucional de la Universidad Politécnica de Valencia. <https://n9.cl/yy01q>
- Solano, C. (2023). Nonlinear numerical techniques for the processing of data with discontinuities [Universidad Politécnica de Cartagena]. <https://doi.org/10.31428/10317/13128>
- Tineo, H., & Torres, P. (2022). Métodos de Integración Numérica Asistido por la Interfaz Gráfica de Matlab [Tesis de Licenciatura, Universidad Nacional Pedro Ruiz Gallo]. <http://repositorio.unprg.edu.pe/handle/20.500.12893/12284>

Anexos

Archivo del método del trapecio

```
classdef metodotrapecio_App < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        figure1      matlab.ui.Figure
    text1      matlab.ui.control.Label
        text12
        matlab.ui.control.Label
        uitable1
        matlab.ui.control.Table
        solucion_grafica
        matlab.ui.control.Button
        uipanel3
        matlab.ui.container.Panel
        solucion_numerica
        matlab.ui.control.Button
        salir
        matlab.ui.control.Button
        borrar
        matlab.ui.control.Button
        uipanel1
        matlab.ui.container.Panel
        N
        matlab.ui.control.EditField
        text6
        matlab.ui.control.Label
    end
end
```

```

    matlab.ui.control.EditField
    text3
    matlab.ui.control.Label
    probar_con_datos_predefinidos
    matlab.ui.control.Button
    funcion
    matlab.ui.control.EditField
    text2
    matlab.ui.control.Label
    axes1
    matlab.ui.control.UIAxes
end

methods (Access = private)
    % Update components that require runtime
    %configuration
    function addRuntimeConfigurations(app)

        % Set component properties that require
        %runtime configuration
        app.uitable1.BackgroundColor = [1 1 1];
        app.uitable1.ColumnFormat = {[ ] [ ] [ ] [ ] [ ]};
    end
end

% Callbacks that handle component events
methods (Access = private)
% Code that executes after component creation

```

```

function metodotrapecio_OpeningFcn(app, varargin)
    % Add runtime required configuration - Added
    %by Migration Tool
    addRuntimeConfigurations(app);
    % Ensure that the app appears on screen
    % when run
    movegui(app.figure1, 'onscreen');

    % Create GUIDE-style callback args - Added by
% Migration Tool
    [hObject, eventdata, handles] =
        convertToGUIDECallbackArguments(app);
    %#ok<ASGLU>

    set(handles.uitable1,'data', [ ])
    %set(handles.uitable1,'ColumnName',
    {'T','y1','y2','y3'})
    set(handles.text12,'string','La integral
    aproximada es: ')
    plot(1,1)
    handles.output = hObject;
    guidata(hObject, handles);
end

% Button pushed function: borrar
function borrar_Callback(app, event)
    % Create GUIDE-style callback args - Added by
%Migration Tool

```

```

[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
    %#ok<ASGLU>

set(handles.text12,'string','La integral
aproximada es: ')
set(handles.funcion,'String',' ');
set(handles.I,'String',' ');
set(handles.N,'String',' ');
set(handles.uitable1,'Data', [ ]);
set(handles.uitable1,'ColumnName',{' ',' ','
',' '})
plot(1,1)
end

% Button pushed function:
%probar_con_datos_predefinidos
function
probar_con_datos_predefinidos_Callback(app,
event)
    % Create GUIDE-style callback args - Added by
Migration Tool
[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
    %#ok<ASGLU>

plot(1,1)
set(handles.uitable1,'data',{ })

```

```

set(handles.text12,'string','La integral
aproximada es: ')
fun=' x^3*exp(x^2) ';
int=      [ 0 , 2]      ;
n=      12      ;
set(handles.funcion,'String',fun);
set(handles.I,'String', '[ 0 , 2] ');
set(handles.N,'String',n);
end

% Button pushed function: salir
function salir_Callback(app, event)
    % Create GUIDE-style callback args - Added by
    Migration Tool
    [hObject, eventdata, handles] =
    convertToGUIDECallbackArguments(app, event);
    %#ok<ASGLU>

    opc=questdlg('¿Desea salir del
    programa?', 'SALIR', 'Si', 'No', 'No');
    if strcmp(opc, 'No')
    return;

    else
        msgbox({'Gracias por usar nuestra
        Aplicación',...
        '
        Desarrollado

```

```

        por:',...
        'Sandoval Tarrillo Xintia
        Haiduk','Lazo Enriquez Lina Elizabeth',
        'Dr. Cuti Gutiérrez Alcides
        Raúl'},'Acerca de');
    end
    clear,clc,close(gcf)
end

% Button pushed function: solucion_grafica
function solucion_grafica_Callback(app, event)
% Create GUIDE-style callback args - Added by
Migration Tool
[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

plot(1,1) % Inicializa la figura

a = handles.a;
b = handles.b;
f = handles.f;
F = handles.F;
n = handles.n;
X = handles.x;

XX = a:0.1:b; % Puntos para curva suave

```

```

for i = 1:length(XX)
    FF(i) = f(XX(i));
end

color = rand(n, 3); % Genera un color
%diferente para cada trapecio

hold on
for i = 1:n
    xx = [X(i), X(i+1), X(i+1), X(i), X(i)];
    yy = [0, 0, F(i+1), F(i), 0];
    fill(xx, yy, color(i,:), 'EdgeColor', 'k',
        'FaceAlpha', 0.5); % color por trapecio
end

plot(XX, FF, 'b', 'LineWidth', 1.5)

grid on
xlabel('x')
ylabel('f(x)')
xlim([a - 0.5, b + 0.5])
hold off
end

% Button pushed function: solucion_numerica
function solucion_numerica_Callback(app, event)
    % Create GUIDE-style callback args - Added by
    Migration Tool

```

```

[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

set(handles.text12,'string','La integral
aproximada es: ')
fun=get(handles.funcion,'String');
int=str2num(get(handles.I,'String'));
n=str2double(get(handles.N,'String'));
f=str2func(['@(x)',fun]) ;
a=int(1);
b=int(2);
h=(b-a)/n;
x(1)=a;          x(n+1)=b;
F(1)=f(x(1));   F(n+1)=f(x(n+1));
s=abs(F(1))+abs(F(n+1));
I(1)=0;
    for i=2:n
        I(i)=i;
        x(i)=a+(i-1)*h;
        F(i)=f(x(i));
        s=s+2*abs(F(i));
    end
Integral=(h/2)*(s);
I(n+1)=n+1;

    for i=1:n+1
        M{i,1}=sprintf('%5d',i-1);
    end

```

```

        M{i,2}=sprintf('%12.5f',x(i));
        M{i,3}=sprintf('%12.5f',F(i));
    end
    set(handles.uitable1,'ColumnName',
        {'i','x','f(x)'})
handles.a=a;
handles.b=b;
handles.f=f;
handles.n=n;
handles.x=x;
handles.F=F;
texto=get(handles.text12,'string');
Integral=sprintf('%10.8f',Integral);
set(handles.text12,'string',[texto,Integral,
    ' u^2'])
    guidata(hObject, handles);
    set(handles.uitable1,'data', M)
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create figure1 and hide until all
%components are created

```

```

app.figure1 = uifigure('Visible', 'off');
app.figure1.Color = [0.301960784313725
0.745098039215686 0.933333333333333];
colormap(app.figure1, 'parula');
app.figure1.Position = [582 146 1327 661];
app.figure1.Name = 'metodotrapeccio';
app.figure1.Resize = 'off';
app.figure1.HandleVisibility = 'callback';
app.figure1.Tag = 'figure1';

% Create axes1
app.axes1 = uiaxes(app.figure1);
app.axes1.FontSize = 13.3333333333333;
app.axes1.NextPlot = 'replace';
app.axes1.Tag = 'axes1';
app.axes1.Position = [571 45 647 325];

% Create uipanel1
app.uipanel1 = uipanel(app.figure1);
app.uipanel1.BorderColor = [1 1 1];
app.uipanel1.HighlightColor = [1 1 1];
app.uipanel1.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.uipanel1.Tag = 'uipanel1';
app.uipanel1.FontWeight = 'bold';
app.uipanel1.FontSize = 10.6666666666667;
app.uipanel1.Position = [22 445 1053 146];

```

```

% Create text2
app.text2 = uilabel(app.uipanel1);
app.text2.Tag = 'text2';
app.text2.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.text2.HorizontalAlignment = 'center';
app.text2.VerticalAlignment = 'top';
app.text2.WordWrap = 'on';
app.text2.FontSize = 16;
app.text2.FontWeight = 'bold';
app.text2.FontColor = [1 1 1];
app.text2.Position = [51 91 187 22];
app.text2.Text = 'Ingrese la funcion f(x)=';

% Create funcion
app.funcion = uieditfield(app.uipanel1,
'text');
app.funcion.Tag = 'funcion';
app.funcion.FontSize = 16;
app.funcion.FontWeight = 'bold';
app.funcion.FontColor = [0 0 1];
app.funcion.Position = [243 85 769 35];

% Create probar_con_datos_predefinidos
app.probar_con_datos_predefinidos =
uibutton(app.uipanel1, 'push');

```

```

app.probar_con_datos_predefinidos.ButtonPushedFcn
    = createCallbackFcn(app,
        @probar_con_datos_predefinidos_Callback,
        true);
app.probar_con_datos_predefinidos.Tag =
    'probar_con_datos_predefinidos';
app.probar_con_datos_predefinidos.BackgroundColor
    = [0 1 0];
app.probar_con_datos_predefinidos.FontSize =
    16;
app.probar_con_datos_predefinidos.FontWeight
    = 'bold';
app.probar_con_datos_predefinidos.FontColor =
    [0 0 1];
app.probar_con_datos_predefinidos.Position =
    [563 18 358.4 41.6];
app.probar_con_datos_predefinidos.Text =
    'PROBAR CON DATOS PRE DEFINIDOS';

% Create text3
app.text3 = uilabel(app.uipanel1);
app.text3.Tag = 'text3';
app.text3.BackgroundColor =
    [0.301960784313725 0.745098039215686
    0.933333333333333];
app.text3.HorizontalAlignment = 'center';
app.text3.VerticalAlignment = 'top';
app.text3.WordWrap = 'on';

```

```

app.text3.FontSize = 16;
app.text3.FontWeight = 'bold';
app.text3.FontColor = [1 1 1];
app.text3.Position = [13 44 222.4 28.8];
app.text3.Text = 'Ingrese el Intervalo [a ,
b] =';

% Create I
app.I = uieditfield(app.uipanel1, 'text');
app.I.Tag = 'I';
app.I.FontSize = 16;
app.I.FontWeight = 'bold';
app.I.FontColor = [0 0 1];
app.I.Position = [243 44 195 32];

% Create text6
app.text6 = uilabel(app.uipanel1);
app.text6.Tag = 'text6';
app.text6.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.text6.HorizontalAlignment = 'center';
app.text6.VerticalAlignment = 'top';
app.text6.WordWrap = 'on';
app.text6.FontSize = 16;
app.text6.FontWeight = 'bold';
app.text6.FontColor = [1 1 1];
app.text6.Position = [43 3 200 27];

```

```

app.text6.Text = 'Nº de subintervalos n=';

% Create N
app.N = uieditfield(app.uipanel1, 'text');
app.N.Tag = 'N';
app.N.FontSize = 16;
app.N.FontWeight = 'bold';
app.N.FontColor = [0 0 1];
app.N.Position = [242 4 195 36];

% Create uipanel3
app.uipanel3 = uipanel(app.figure1);
app.uipanel3.BorderColor = [1 1 1];
app.uipanel3.HighlightColor = [1 1 1];
app.uipanel3.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.uipanel3.Tag = 'uipanel3';
app.uipanel3.FontSize = 10.6666666666667;
app.uipanel3.Position = [26 375 415 55];

% Create borrar
app.borrar = uibutton(app.uipanel3, 'push');
app.borrar.ButtonPushedFcn =
createCallbackFcn(app, @borrar_Callback,
true);

app.borrar.Tag = 'borrar';

```

```

app.borrar.BackgroundColor = [1 0 1];
app.borrar.FontSize = 16;
app.borrar.FontWeight = 'bold';
app.borrar.FontColor = [1 1 1];
app.borrar.Position = [223 9 84 28.8];
app.borrar.Text = 'BORRAR';

% Create salir
app.salir = uibutton(app.uipanel3, 'push');
app.salir.ButtonPushedFcn =
createCallbackFcn(app, @salir_Callback,
true);
app.salir.Tag = 'salir';
app.salir.BackgroundColor = [1 0 0];
app.salir.FontSize = 16;
app.salir.FontWeight = 'bold';
app.salir.FontColor = [1 1 1];
app.salir.Position = [322 9 73.6 28.8];
app.salir.Text = 'SALIR';

% Create solucion_numerica
app.solucion_numerica =
uibutton(app.uipanel3, 'push');
app.solucion_numerica.ButtonPushedFcn =
createCallbackFcn(app,
@solucion_numerica_Callback, true);
app.solucion_numerica.Tag =
'solucion_numerica';

```

```

app.solucion_numerica.BackgroundColor = [1 1
0];
app.solucion_numerica.FontSize = 16;
app.solucion_numerica.FontWeight = 'bold';
app.solucion_numerica.FontColor = [0 0 1];
app.solucion_numerica.Position = [13 8 192
29.6];
app.solucion_numerica.Text = 'SOLUCIÓN
NUMÉRICA';

% Create solucion_grafica
app.solucion_grafica = uibutton(app.figure1,
'push');
app.solucion_grafica.ButtonPushedFcn =
createCallbackFcn(app,
@solucion_grafica_Callback, true);
app.solucion_grafica.Tag =
'solucion_grafica';
app.solucion_grafica.BackgroundColor = [0 0
1];
app.solucion_grafica.FontSize = 16;
app.solucion_grafica.FontWeight = 'bold';
app.solucion_grafica.FontColor = [1 1 1];
app.solucion_grafica.Position = [843.4 385.8
223.2 35.2];
app.solucion_grafica.Text = 'SOLUCIÓN
GRÁFICA';

```

```

% Create uitable1
app.uitable1 = uitable(app.figure1);
app.uitable1.ColumnName = {''; ''; ''; ''';
''};
app.uitable1.ColumnWidth = {100, 100, 100,
100, 100};
app.uitable1.ColumnEditable = [false false
false false false];
app.uitable1.ForegroundColor = [0 0 1];
app.uitable1.Tag = 'uitable1';
app.uitable1.FontWeight = 'bold';
app.uitable1.FontSize = 16;
app.uitable1.Position = [40 66 427 296];

% Create text12
app.text12 = uilabel(app.figure1);
app.text12.Tag = 'text12';
app.text12.BackgroundColor = [1 1 1];
app.text12.VerticalAlignment = 'top';
app.text12.WordWrap = 'on';
app.text12.FontSize = 16;
app.text12.FontWeight = 'bold';
app.text12.FontColor = [0 0 1];
app.text12.Position = [49.8 16.2 430.4 29.6];
app.text12.Text = 'La integral aproximada es:
';

% Create text1

```

```

app.text1 = uilabel(app.figure1);
app.text1.Tag = 'text1';
app.text1.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.text1.HorizontalAlignment = 'center';
app.text1.VerticalAlignment = 'top';
app.text1.WordWrap = 'on';
app.text1.FontSize = 25.3333333333333;
app.text1.FontWeight = 'bold';
app.text1.FontColor = [1 1 1];
app.text1.Position = [259.4 611.4 896.8
33.6];
app.text1.Text = 'MÉTODO DEL TRAPECIO PARA
APROXIMAR INTEGRALES DEFINIDAS';

% Show the figure after all components are
created
app.figure1.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = metodotrapecio_App(varargin)

```

```

runningApp = getRunningApp(app);

% Check for running singleton app
if isempty(runningApp)

    % Create UIFigure and components
    createComponents(app)

    % Register the app with App Designer
    registerApp(app, app.figure1)

    % Execute the startup function
    runStartupFcn(app,
        @(app)metodotrapeccio_OpeningFcn(app,
            varargin{:}))
else

    % Focus the running singleton app
    figure(runningApp.figure1)

    app = runningApp;
end

if nargin == 0
    clear app
end
end
end

```

```
% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.figure1)
end
end
end
```

Archivo del método de Simpson compuesto 1/3

```
classdef metodosimpson13_App < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)

        figure1                matlab.ui.Figure
        text12
        matlab.ui.control.Label
        uitable1
        matlab.ui.control.Table
        solucion_grafica
        matlab.ui.control.Button
        uipanel3
        matlab.ui.container.Panel
        solucion_numerica
        matlab.ui.control.Button
        salir
        matlab.ui.control.Button
        borrar
        matlab.ui.control.Button
        uipanel1
        matlab.ui.container.Panel
        N
        matlab.ui.control.EditField
        text6
        matlab.ui.control.Label
        I
```

```

matlab.ui.control.EditField
text3
matlab.ui.control.Label
probar_con_datos_predefinidos
matlab.ui.control.Button
funcion
matlab.ui.control.EditField
text2
matlab.ui.control.Label
text1
matlab.ui.control.Label
axes1
matlab.ui.control.UIAxes
end

methods (Access = private)
    % Update components that require runtime
    %configuration
    function addRuntimeConfigurations(app)

        % Set component properties that require
        %runtime configuration
        app.uitable1.BackgroundColor = [1 1 1];
app.uitable1.ColumnFormat = {[[] [] [] [] []]};
    end
end

```

```

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function metodosimpson13_OpeningFcn(app,
varargin)
    % Add runtime required configuration - Added
    by Migration Tool
    addRuntimeConfigurations(app);

% Ensure that the app appears on screen when
run
movegui(app.figure1, 'onscreen');

% Create GUIDE-style callback args - Added by
Migration Tool
[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app);
%#ok<ASGLU>

set(handles.uitable1,'data', [ ])
%set(handles.uitable1,'ColumnName',
{'T','y1','y2','y3'})
set(handles.text12,'string','La integral
aproximada es: ')
plot(1,1)
handles.output = hObject;
guidata(hObject, handles);

```

```

end

% Button pushed function: borrar
function borrar_Callback(app, event)
    % Create GUIDE-style callback args - Added by
    Migration Tool
    [hObject, eventdata, handles] =
    convertToGUIDECallbackArguments(app, event);
    %#ok<ASGLU>

    set(handles.text12, 'string', 'La integral
    aproximada es: ')
    set(handles.funcion, 'String', ' ');
    set(handles.I, 'String', ' ');
    set(handles.N, 'String', ' ');
    set(handles.uitable1, 'Data', [ ]);
    set(handles.uitable1, 'ColumnName', {' ', ' ', ' ',
    ' ', ' '})
    plot(1,1)
end

% Button pushed function:
probar_con_datos_predefinidos
function
probar_con_datos_predefinidos_Callback(app,
event)
    % Create GUIDE-style callback args - Added by
    Migration Tool

```

```

[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

plot(1,1)
set(handles.uitable1,'data',{ })
set(handles.text12,'string','La integral
aproximada es: ')
fun=' x^3*exp(x^2) ';
int= [ 0 , 2] ;
n= 12 ;
set(handles.funcion,'String',fun);
set(handles.I,'String', '[ 0 , 2] ');
set(handles.N,'String',n);

end

% Button pushed function: salir
function salir_Callback(app, event)
% Create GUIDE-style callback args - Added by
Migration Tool
[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

opc=questdlg('¿Desea salir del
programa?', 'SALIR', 'Si', 'No', 'No');
if strcmp(opc, 'No')

```

```

return;

else
    msgbox({'Gracias por usar nuestra
Aplicación',...
        '
Desarrollado
por:',...
        'Sandoval Tarrillo Xintia
Haiduk','Lazo Enriquez Lina
Elizabeth', 'Dr. Cuti Gutiérrez
Alcides Raúl'},'Acerca de');
end
clear,clc,close(gcf)
end

% Button pushed function: solucion_grafica
function solucion_grafica_Callback(app, event)
    % Create GUIDE-style callback args - Added by
Migration Tool
[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

plot(1,1)
a = handles.a;
b = handles.b;
f = handles.f;
F = handles.F;

```

```

n = handles.n;
X = handles.x;
XX = a:0.1:b;

% Calcular valores para la curva suave
for i = 1:length(XX)
    FF(i) = f(XX(i));
end

color = rand(n/2, 3); % un color por cada
par (n debe ser par)
l = 1;
hold on

% Rellenar con áreas curvas para cada par de
%subintervalos
for i = 1:2:n-1
    xi = X(i);
    xmid = X(i+1);
    xi2 = X(i+2);

    % Interpolación cuadrática para sombrear
    %el área
    px = linspace(xi, xi2, 50);
    yi = f(xi);
    ymid = f(xmid);
    yi2 = f(xi2);

```

```

    % Polinomio de Lagrange de grado 2
    L = @(x) yi*((x - xmid).*(x - xi2))/((xi
    - xmid)*(xi - xi2)) + ...
        ymid*((x - xi).*(x - xi2))/
    ((xmid - xi)*(xmid - xi2)) + ...
    yi2*((x - xi).*(x - xmid))/((xi2 - xi)*(xi2
    - xmid));

    yy = L(px);
    fill([px fliplr(px)], [yy
    zeros(size(yy))], color(1,:),
    'EdgeColor', 'none', 'FaceAlpha', 0.5)

    l = l + 1;
end

% Dibujar la curva de la función original
plot(XX, FF, 'b', 'LineWidth', 1.5)
grid on
xlabel('x')
ylabel('f(x)')
xlim([a-0.5, b+0.5])
hold off
zoom on
end

% Button pushed function: solucion_numerica
function solucion_numerica_Callback(app, event)

```

```

% Create GUIDE-style callback args - Added by
Migration Tool

[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

set(handles.text12, 'String', 'La integral
aproximada es: ');

% Obtener función como texto y corregir
operadores
fun = get(handles.funcion, 'String');
fun = strrep(fun, '*', '.*');
fun = strrep(fun, '/', './');
fun = strrep(fun, '^', '.^');
f = str2func(['@(x)', fun]);

% Obtener intervalos y subintervalos
int = str2num(get(handles.I, 'String')); %
debe tener dos valores: [a b]
n = str2double(get(handles.N, 'String'));

% Validar valor de n
if isnan(n) || n <= 0 || rem(n,1)~=0
    msgbox('Debe ingresar un número entero
positivo para n.', 'Error', 'error');
    set(handles.N, 'String', '');
    set(handles.uitable1, 'Data', []);

```

```

        set(handles.uitable1, 'ColumnName', {' ',
        ' ', ' '});
        cla; % Limpiar gráficos sin crear nuevos
        return;
end

% Verificar si n es par (requisito de Simpson
1/3)
if rem(n,2) ~= 0
    msgbox('Para aplicar Simpson 1/3, n debe
    ser un número par.', 'Advertencia',
    'warn');
    set(handles.N, 'String', '');
    set(handles.uitable1, 'Data', []);
    set(handles.uitable1, 'ColumnName', {' ',
    ' ', ' '});
    cla; % Limpiar sin plot(1,1)
    return;
end

% Definiciones
a = int(1);
b = int(2);
h = (b - a) / n;

x = a:h:b;           % n+1 puntos
fx = f(x);           % Evaluar función en los
puntos

```

```

% Cálculo Simpson 1/3
suma_pares = sum(fx(3:2:end-1));
suma_impares = sum(fx(2:2:end));
Integral = (h/3)*(fx(1) + 2*suma_pares +
4*suma_impares + fx(end));

% Mostrar en la tabla
for i = 1:n+1
    M{i,1} = sprintf('%5d', i-1);
    M{i,2} = sprintf('%12.5f', x(i));
    M{i,3} = sprintf('%12.5f', fx(i));
end

set(handles.uitable1, 'ColumnName', {'i',
'x', 'f(x)'});
set(handles.uitable1, 'Data', M);

% Guardar valores para graficar
handles.a = a;
handles.b = b;
handles.f = f;
handles.n = n;
handles.x = x;
handles.F = fx;

texto = get(handles.text12, 'String');
Integral_str = sprintf('%10.8f', Integral);

```

```

        set(handles.text12, 'String', [texto,
        Integral_str, 'u^2']);
        guidata(hObject, handles);
    end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create figure1 and hide until all
components are created
app.figure1 = uifigure('Visible', 'off');
app.figure1.Color = [0.301960784313725
0.745098039215686 0.933333333333333];
colormap(app.figure1, 'parula');
app.figure1.Position = [582 143 1438 664];
app.figure1.Name = 'metodosimpson13';
app.figure1.HandleVisibility = 'callback';
app.figure1.Tag = 'figure1';

% Create axes1
app.axes1 = uiaxes(app.figure1);
app.axes1.FontSize = 13.3333333333333;
app.axes1.NextPlot = 'replace';
app.axes1.Tag = 'axes1';

```

```

app.axes1.Position = [580 63 801 361];

% Create text1
app.text1 = uilabel(app.figure1);
app.text1.Tag = 'text1';
app.text1.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.text1.HorizontalAlignment = 'center';
app.text1.VerticalAlignment = 'top';
app.text1.WordWrap = 'on';
app.text1.FontSize = 25.3333333333333;
app.text1.FontWeight = 'bold';
app.text1.FontColor = [1 1 1];
app.text1.Position = [4.84942016057092
623.179948586118 1385.79125780553
36.6992287917737];
app.text1.Text = 'MÉTODO DE SIMPSON COMPUESTO
1/3 PARA APROXIMAR INTEGRALES DEFINIDAS';

% Create uipanel1
app.uipanel1 = uipanel(app.figure1);
app.uipanel1.BorderColor = [1 1 1];
app.uipanel1.HighlightColor = [1 1 1];
app.uipanel1.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.uipanel1.Tag = 'uipanel1';

```

```

app.uipanel1.FontWeight = 'bold';
app.uipanel1.FontSize = 10.66666666666667;
app.uipanel1.Position = [15 484 1206 130];

% Create text2
app.text2 = uilabel(app.uipanel1);
app.text2.Tag = 'text2';
app.text2.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.text2.HorizontalAlignment = 'center';
app.text2.VerticalAlignment = 'top';
app.text2.WordWrap = 'on';
app.text2.FontSize = 16;
app.text2.FontWeight = 'bold';
app.text2.FontColor = [1 1 1];
app.text2.Position = [62 80 183.204725379246
20.8];
app.text2.Text = 'Ingrese la funcion f(x)=';

% Create funcion
app.funcion = uieditfield(app.uipanel1,
'text');
app.funcion.Tag = 'funcion';
app.funcion.FontSize = 16;
app.funcion.FontWeight = 'bold';
app.funcion.FontColor = [0 0 1];
app.funcion.Position = [260 78

```

```

881.622739597943 29.6];

% Create probar_con_datos_predefinidos
app.probar_con_datos_predefinidos =
uibutton(app.uipanel1, 'push');
app.probar_con_datos_predefinidos.ButtonPushedFcn
= createCallbackFcn(app,
@probar_con_datos_predefinidos_Callback,
true);
app.probar_con_datos_predefinidos.Tag =
'probar_con_datos_predefinidos';
app.probar_con_datos_predefinidos.BackgroundColor
= [1 1 0];
app.probar_con_datos_predefinidos.FontSize =
16;
app.probar_con_datos_predefinidos.FontWeight
= 'bold';
app.probar_con_datos_predefinidos.FontColor =
[0 0 1];
app.probar_con_datos_predefinidos.Position =
[717 16 320.808274572391 33.6];
app.probar_con_datos_predefinidos.Text =
'PROBAR CON DATOS PRE DEFINIDOS';

% Create text3
app.text3 = uilabel(app.uipanel1);
app.text3.Tag = 'text3';
app.text3.BackgroundColor =

```

```

[0.301960784313725 0.745098039215686
0.933333333333333];
app.text3.HorizontalAlignment = 'center';
app.text3.VerticalAlignment = 'top';
app.text3.WordWrap = 'on';
app.text3.FontSize = 16;
app.text3.FontWeight = 'bold';
app.text3.FontColor = [1 1 1];
app.text3.Position = [19 49 225.605819025971
20.8];
app.text3.Text = 'Ingrese el Intervalo [a ,
b] =';

% Create I
app.I = uieditfield(app.uipanel1, 'text');
app.I.Tag = 'I';
app.I.FontSize = 16;
app.I.FontWeight = 'bold';
app.I.FontColor = [0 0 1];
app.I.Position = [259 44 223.205757121439
30.4];

% Create text6
app.text6 = uilabel(app.uipanel1);
app.text6.Tag = 'text6';
app.text6.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];

```

```

app.text6.HorizontalAlignment = 'center';
app.text6.VerticalAlignment = 'top';
app.text6.WordWrap = 'on';
app.text6.FontSize = 16;
app.text6.FontWeight = 'bold';
app.text6.FontColor = [1 1 1];
app.text6.Position = [61 16 183.204725379246
22.4];
app.text6.Text = 'Nº de subintervalos n=';

% Create N
app.N = uieditfield(app.uipanel1, 'text');
app.N.Tag = 'N';
app.N.FontSize = 16;
app.N.FontWeight = 'bold';
app.N.FontColor = [0 0 1];
app.N.Position = [259 8 223.205757121439
31.2];

% Create uipanel3
app.uipanel3 = uipanel(app.figure1);
app.uipanel3.BorderColor = [1 1 1];
app.uipanel3.HighlightColor = [1 1 1];
app.uipanel3.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.uipanel3.Tag = 'uipanel3';
app.uipanel3.FontSize = 10.66666666666667;

```

```

app.uipanel3.Position = [28 405 408 51];

% Create borrar
app.borrar = uibutton(app.uipanel3, 'push');
app.borrar.ButtonPushedFcn =
createCallbackFcn(app, @borrar_Callback,
true);
app.borrar.Tag = 'borrar';
app.borrar.BackgroundColor = [1 0 0];
app.borrar.FontSize = 16;
app.borrar.FontWeight = 'bold';
app.borrar.FontColor = [1 1 1];
app.borrar.Position = [210 13 84 28.8];
app.borrar.Text = 'BORRAR';

% Create salir
app.salir = uibutton(app.uipanel3, 'push');
app.salir.ButtonPushedFcn =
createCallbackFcn(app, @salir_Callback,
true);
app.salir.Tag = 'salir';
app.salir.BackgroundColor = [0 0 1];
app.salir.FontSize = 16;
app.salir.FontWeight = 'bold';
app.salir.FontColor = [1 1 1];
app.salir.Position = [299 11 83.2 28.8];
app.salir.Text = 'SALIR';

```

```

% Create solucion_numerica
app.solucion_numerica =
uibutton(app.uipanel3, 'push');
app.solucion_numerica.ButtonPushedFcn =
createCallbackFcn(app,
@solucion_numerica_Callback, true);
app.solucion_numerica.Tag =
'solucion_numerica';
app.solucion_numerica.BackgroundColor = [0 1
0];
app.solucion_numerica.FontSize = 16;
app.solucion_numerica.FontWeight = 'bold';
app.solucion_numerica.FontColor = [0 0 1];
app.solucion_numerica.Position = [15 14 192
28.8];
app.solucion_numerica.Text = 'SOLUCIÓN
NUMÉRICA';

% Create solucion_grafica
app.solucion_grafica = uibutton(app.figure1,
'push');
app.solucion_grafica.ButtonPushedFcn =
createCallbackFcn(app,
@solucion_grafica_Callback, true);
app.solucion_grafica.Tag =
'solucion_grafica';
app.solucion_grafica.BackgroundColor =
[0.717647058823529 0.274509803921569 1];

```

```

app.solucion_grafica.FontSize = 16;
app.solucion_grafica.FontWeight = 'bold';
app.solucion_grafica.FontColor = [1 1 1];
app.solucion_grafica.Position = [645 437
177.6 28.8];
app.solucion_grafica.Text = 'SOLUCIÓN
GRÁFICA';

% Create uitable1
app.uitable1 = uitable(app.figure1);
app.uitable1.ColumnName = {''; ''; ''; ''};
''};
app.uitable1.ColumnWidth = {100, 100, 100,
100, 100};
app.uitable1.ColumnEditable = [false false
false false false];
app.uitable1.ForegroundColor = [0 0 1];
app.uitable1.Tag = 'uitable1';
app.uitable1.FontSize = 16;
app.uitable1.Position = [50 83 431 308];

% Create text12
app.text12 = uilabel(app.figure1);
app.text12.Tag = 'text12';
app.text12.BackgroundColor = [1 1 1];
app.text12.VerticalAlignment = 'top';
app.text12.WordWrap = 'on';
app.text12.FontSize = 16;

```

```

    app.text12.FontWeight = 'bold';
    app.text12.FontColor = [0 0 1];
    app.text12.Position = [30.6 26.6 524 26.4];
    app.text12.Text = 'La integral aproximada es:
    ';

    % Show the figure after all components are
    created
    app.figure1.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = metodosimpson13_App(varargin)

        runningApp = getRunningApp(app);

        % Check for running singleton app
        if isempty(runningApp)

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.figure1)

```

```

        % Execute the startup function
        runStartupFcn(app,
            @(app)metodosimpson13_OpeningFcn(app,
                varargin{:}))
    else

        % Focus the running singleton app
        figure(runningApp.figure1)

        app = runningApp;
    end

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.figure1)
end
end
end
end

```

Archivo del método de Simpson compuesto 3/8

```
classdef metodosimpson38_App < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)

        figure1                matlab.ui.Figure
        text12
        matlab.ui.control.Label
        uitable1
        matlab.ui.control.Table
        solucion_grafica
        matlab.ui.control.Button
        uipanel3
        matlab.ui.container.Panel
        solucion_numerica
        matlab.ui.control.Button
        salir
        matlab.ui.control.Button
        borrar
        matlab.ui.control.Button
        uipanel1
        matlab.ui.container.Panel
        N
        matlab.ui.control.EditField
        text6
        matlab.ui.control.Label
        I
```

```

matlab.ui.control.EditField
text3
matlab.ui.control.Label
probar_con_datos_predefinidos
matlab.ui.control.Button
funcion
matlab.ui.control.EditField
text2
matlab.ui.control.Label
text1
matlab.ui.control.Label
axes1
matlab.ui.control.UIAxes
end

methods (Access = private)
    % Update components that require runtime
    %configuration
    function addRuntimeConfigurations(app)

        % Set component properties that require
        %runtime configuration
        app.uitable1.BackgroundColor = [1 1 1];
        app.uitable1.ColumnFormat = {[[] [] [] [] []]};
    end
end
end

```

```

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function metodosimpson38_OpeningFcn(app,
varargin)
    % Add runtime required configuration - Added
    %by Migration Tool
    addRuntimeConfigurations(app);

    % Ensure that the app appears on screen when
    %run
    movegui(app.figure1, 'onscreen');

    % Create GUIDE-style callback args - Added by
    %Migration Tool
    [hObject, eventdata, handles] =
    %convertToGUIDECallbackArguments(app);
    %#ok<ASGLU>

    set(handles.uitable1,'data', [ ])
    %set(handles.uitable1,'ColumnName',
    {'T','y1','y2','y3'})
    set(handles.text12,'string','La integral
    aproximada es: ')
    plot(1,1)
    handles.output = hObject;
    guidata(hObject, handles);

```

```

end

% Button pushed function: borrar
function borrar_Callback(app, event)
    % Create GUIDE-style callback args - Added by
    Migration Tool
    [hObject, eventdata, handles] =
    convertToGUIDECallbackArguments(app, event);
    %#ok<ASGLU>

    set(handles.text12,'string','La integral
    aproximada es: ')
    set(handles.funcion,'String',' ');
    set(handles.I,'String',' ');
    set(handles.N,'String',' ');
    set(handles.uitable1,'Data', [ ]);
    set(handles.uitable1,'ColumnName',{' ',' ',' '
    ',' '})
    plot(1,1)
end

% Button pushed function:
probar_con_datos_predefinidos
function
probar_con_datos_predefinidos_Callback(app,
event)
    % Create GUIDE-style callback args - Added by
    Migration Tool

```

```

[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

plot(1,1)
set(handles.uitable1,'data',{ })
set(handles.text12,'string','La integral
aproximada es: ')
fun=' x^3*exp(x^2) ';
int=      [ 0 , 2] ;
n=      12 ;
set(handles.funcion,'String',fun);
set(handles.I,'String', '[ 0,2] ');
set(handles.N,'String',n);

end

% Button pushed function: salir
function salir_Callback(app, event)
% Create GUIDE-style callback args - Added by
%Migration Tool
[hObject, eventdata, handles] =
convertToGUIDECallbackArguments(app, event);
%#ok<ASGLU>

opc=questdlg('¿Desea salir del
programa?', 'SALIR', 'Si', 'No', 'No');
if strcmp(opc, 'No')

```

```

return;

else
    msgbox({'Gracias por usar nuestra
    Aplicación',...
           '
           Desarrollado
por: ',...
           'Sandoval Tarrillo Xintia
Haiduk', 'Lazo Enriquez Lina
Elizabeth', 'Dr. Cuti Gutiérrez
Alcides Raúl'}, 'Acerca de');
end
clear,clc,close(gcf)
end

% Button pushed function: solucion_grafica
function solucion_grafica_Callback(app, event)
    % Create GUIDE-style callback args - Added by
    %Migration Tool
    [hObject, eventdata, handles] =
    convertToGUIDECallbackArguments(app, event);
    %#ok<ASGLU>

    plot(1,1)
    a=handles.a;
    b=handles.b;
    f=handles.f;
    F=handles.F;

```

```

n=handles.n;
X=handles.x;
XX=a:0.1:b;
for i=1:length(XX)
    FF(i)=f(XX(i));
end
color=rand(n,3);
l=1;
hold on
for i=1:n
    xx=[X(i)    X(i+1)                X(i+1)
        X(i)                X(i)  ] ;
    yy=[ 0          0                F(i+1)
        F(i)          0];
    fill(xx,yy,color(l,:))
    if rem(i,3)==0
        l=l+1;
    end
end

end
plot(XX,FF,'b')
grid on
xlabel('x')
ylabel('f(x)')
xlim([a-0.5 b+0.5])
hold off
end

```

```

% Button pushed function: solucion_numerica
function solucion_numerica_Callback(app, event)
    % Create GUIDE-style callback args - Added by
    Migration Tool
    [hObject, eventdata, handles] =
    convertToGUIDECallbackArguments(app, event);
    %#ok<ASGLU>

    set(handles.text12, 'String', 'La integral
    aproximada es: ')
    fun = get(handles.funcion, 'String');
    int = str2num(get(handles.I, 'String'));
    n = str2double(get(handles.N, 'String'));

    if rem(n,3) ~= 0
        msgbox('El valor de n debe ser múltiplo
        de 3 para aplicar el método de Simpson
        3/8.', ...
            'Advertencia', 'warn');
        set(handles.N, 'String', '');
        set(handles.uitable1, 'Data', []);
        set(handles.uitable1, 'ColumnName', {' ',
        ' ', ' ', ' '});
        cla; % limpia el gráfico si existe
    else
        f = str2func(['@(x)', fun]);
        a = int(1);
        b = int(2);
    end

```

```

h = (b - a) / n;

x(1) = a;
x(n+1) = b;
F(1) = f(x(1));
F(n+1) = f(x(n+1));

s = abs(F(1)) + abs(F(n+1));
I(1) = 0;

for i = 2:n
    I(i) = i;
    x(i) = a + (i - 1)*h;
    F(i) = f(x(i));

    if rem(i,3) == 0
        s = s + 3*abs(F(i));
    elseif rem(i,3) == 2
        s = s + 3*abs(F(i));
    elseif rem(i,3) == 1
        s = s + 2*abs(F(i));
    end
end

Integral = (3*h/8) * s;
I(n+1) = n+1;

for i = 1:n+1

```

```

        M{i,1} = sprintf('%5d', i-1);
        M{i,2} = sprintf('%12.5f', x(i));
        M{i,3} = sprintf('%12.5f', F(i));
    end

    set(handles.uitable1, 'ColumnName', {'i',
    'x', 'f(x)'})
    set(handles.uitable1, 'Data', M)

    handles.a = a;
    handles.b = b;
    handles.f = f;
    handles.n = n;
    handles.x = x;
    handles.F = F;

    texto = get(handles.text12, 'String');
    Integral = sprintf('%10.8f', Integral);
    set(handles.text12, 'String', [texto,
    Integral, ' u^2'])

    guidata(hObject, handles);
end
end
end

% Component initialization
methods (Access = private)

```

```

% Create UIFigure and components
function createComponents(app)

    % Create figure1 and hide until all
    %components are created
    app.figure1 = uifigure('Visible', 'off');
    app.figure1.Color = [0.301960784313725
    0.745098039215686 0.933333333333333];
    colormap(app.figure1, 'parula');
    app.figure1.Position = [582 143 1438 664];
    app.figure1.Name = 'metodosimpson38';
    app.figure1.HandleVisibility = 'callback';
    app.figure1.Tag = 'figure1';

    % Create axes1
    app.axes1 = uiaxes(app.figure1);
    app.axes1.FontSize = 13.3333333333333;
    app.axes1.NextPlot = 'replace';
    app.axes1.Tag = 'axes1';
    app.axes1.Position = [571 89 801 303];

    % Create text1
    app.text1 = uilabel(app.figure1);
    app.text1.Tag = 'text1';
    app.text1.BackgroundColor =
    [0.301960784313725 0.745098039215686
    0.933333333333333];

```

```

app.text1.HorizontalAlignment = 'center';
app.text1.VerticalAlignment = 'top';
app.text1.WordWrap = 'on';
app.text1.FontSize = 25.33333333333333;
app.text1.FontWeight = 'bold';
app.text1.FontColor = [1 1 1];
app.text1.Position = [4.84942016057092
623.179948586118 1385.79125780553
36.6992287917737];
app.text1.Text = 'MÉTODO DE SIMPSON COMPUESTO
3/8 PARA APROXIMAR INTEGRALES DEFINIDAS';

% Create uipanel1
app.uipanel1 = uipanel(app.figure1);
app.uipanel1.BorderColor = [1 1 1];
app.uipanel1.HighlightColor = [1 1 1];
app.uipanel1.BackgroundColor =
[0.301960784313725 0.745098039215686
0.9333333333333333];
app.uipanel1.Tag = 'uipanel1';
app.uipanel1.FontWeight = 'bold';
app.uipanel1.FontSize = 10.66666666666667;
app.uipanel1.Position = [14 473 1206 130];

% Create text2
app.text2 = uilabel(app.uipanel1);
app.text2.Tag = 'text2';
app.text2.BackgroundColor =

```

```

[0.301960784313725 0.745098039215686
0.933333333333333];
app.text2.HorizontalAlignment = 'center';
app.text2.VerticalAlignment = 'top';
app.text2.WordWrap = 'on';
app.text2.FontSize = 16;
app.text2.FontWeight = 'bold';
app.text2.FontColor = [1 1 1];
app.text2.Position = [20 84 224.65811965812
17.8147651006711];
app.text2.Text = 'Ingrese la funcion f(x)=';

% Create funcion
app.funcion = uieditfield(app.uipanel1,
'text');
app.funcion.Tag = 'funcion';
app.funcion.FontSize = 14.6666666666667;
app.funcion.FontWeight = 'bold';
app.funcion.FontColor = [0 0 1];
app.funcion.Position = [278 80
881.94358974359 26.2979865771812];

% Create probar_con_datos_predefinidos
app.probar_con_datos_predefinidos =
uibutton(app.uipanel1, 'push');
app.probar_con_datos_predefinidos.ButtonPushedFcn
= createCallbackFcn(app,
@probar_con_datos_predefinidos_Callback,

```

```

true);
app.probar_con_datos_predefinidos.Tag =
    'probar_con_datos_predefinidos';
app.probar_con_datos_predefinidos.BackgroundColor
    = [0 0 1];
app.probar_con_datos_predefinidos.FontSize =
    16;
app.probar_con_datos_predefinidos.FontWeight
    = 'bold';
app.probar_con_datos_predefinidos.FontColor =
    [1 1 1];
app.probar_con_datos_predefinidos.Position =
    [628 16 320.586560212907 33.6];
app.probar_con_datos_predefinidos.Text =
    'PROBAR CON DATOS PRE DEFINIDOS';

% Create text3
app.text3 = uilabel(app.uipanel1);
app.text3.Tag = 'text3';
app.text3.BackgroundColor =
    [0.301960784313725 0.745098039215686
    0.933333333333333];
app.text3.HorizontalAlignment = 'center';
app.text3.VerticalAlignment = 'top';
app.text3.WordWrap = 'on';
app.text3.FontSize = 16;
app.text3.FontWeight = 'bold';
app.text3.FontColor = [1 1 1];

```

```

app.text3.Position = [33 37 224.65811965812
22.0563758389262];
app.text3.Text = 'Ingrese el Intervalo [a ,
b] =';

% Create I
app.I = uieditfield(app.uipanel1, 'text');
app.I.Tag = 'I';
app.I.FontSize = 16;
app.I.FontWeight = 'bold';
app.I.FontColor = [0 0 1];
app.I.Position = [277 43 223.051497005988
25.6];

% Create text6
app.text6 = uilabel(app.uipanel1);
app.text6.Tag = 'text6';
app.text6.BackgroundColor =
[0.301960784313725 0.745098039215686
0.933333333333333];
app.text6.HorizontalAlignment = 'center';
app.text6.VerticalAlignment = 'top';
app.text6.WordWrap = 'on';
app.text6.FontSize = 16;
app.text6.FontWeight = 'bold';
app.text6.FontColor = [1 1 1];
app.text6.Position = [32 11 183.577777777778
22.0563758389262];

```

```

app.text6.Text = 'Nº de subintervalos n=';

% Create N
app.N = uieditfield(app.uipanel1, 'text');
app.N.Tag = 'N';
app.N.FontSize = 16;
app.N.FontWeight = 'bold';
app.N.FontColor = [0 0 1];
app.N.Position = [277 9 223.051497005988
24.8];

% Create uipanel3
app.uipanel3 = uipanel(app.figure1);
app.uipanel3.BorderColor = [1 1 1];
app.uipanel3.HighlightColor = [1 1 1];
app.uipanel3.BackgroundColor =
[0.301960784313725 0.745098039215686
0.9333333333333333];
app.uipanel3.Tag = 'uipanel3';
app.uipanel3.FontSize = 10.66666666666667;
app.uipanel3.Position = [32 390 369 55];

% Create borrar
app.borrar = uibutton(app.uipanel3, 'push');
app.borrar.ButtonPushedFcn =
createCallbackFcn(app, @borrar_Callback,
true);
app.borrar.Tag = 'borrar';

```

```

app.borrar.BackgroundColor = [1 1 0];
app.borrar.FontSize = 16;
app.borrar.FontWeight = 'bold';
app.borrar.FontColor = [0 0 1];
app.borrar.Position = [199 14 84 28.8];
app.borrar.Text = 'BORRAR';

% Create salir
app.salir = uibutton(app.uipanel3, 'push');
app.salir.ButtonPushedFcn =
createCallbackFcn(app, @salir_Callback,
true);
app.salir.Tag = 'salir';
app.salir.BackgroundColor = [1 0 0];
app.salir.FontSize = 16;
app.salir.FontWeight = 'bold';
app.salir.FontColor = [1 1 1];
app.salir.Position = [296 18 54.4 24];
app.salir.Text = 'SALIR';

% Create solucion_numerica
app.solucion_numerica =
uibutton(app.uipanel3, 'push');
app.solucion_numerica.ButtonPushedFcn =
createCallbackFcn(app,
@solucion_numerica_Callback, true);
app.solucion_numerica.Tag =
'solucion_numerica';

```

```

app.solucion_numerica.BackgroundColor = [1 0
1];
app.solucion_numerica.FontSize = 16;
app.solucion_numerica.FontWeight = 'bold';
app.solucion_numerica.FontColor = [1 1 1];
app.solucion_numerica.Position = [9 13
183.932773109244 32.8];
app.solucion_numerica.Text = 'SOLUCIÓN
NUMÉRICA';

% Create solucion_grafica
app.solucion_grafica = uibutton(app.figure1,
'push');
app.solucion_grafica.ButtonPushedFcn =
createCallbackFcn(app,
@solucion_grafica_Callback, true);
app.solucion_grafica.Tag =
'solucion_grafica';
app.solucion_grafica.BackgroundColor = [0 1
0];
app.solucion_grafica.FontSize = 16;
app.solucion_grafica.FontWeight = 'bold';
app.solucion_grafica.FontColor = [0 0 1];
app.solucion_grafica.Position = [843.4 426.6
177.6 28.8];
app.solucion_grafica.Text = 'SOLUCIÓN
GRÁFICA';

```

```

% Create uitable1
app.uitable1 = uitable(app.figure1);
app.uitable1.ColumnName = {''; ''; ''; ''';
''};
app.uitable1.ColumnWidth = {100, 100, 100,
100, 100};
app.uitable1.ColumnEditable = [false false
false false false];
app.uitable1.ForegroundColor = [0 0 1];
app.uitable1.Tag = 'uitable1';
app.uitable1.FontWeight = 'bold';
app.uitable1.FontSize = 16;
app.uitable1.Position = [50 82 386 265];

% Create text12
app.text12 = uilabel(app.figure1);
app.text12.Tag = 'text12';
app.text12.VerticalAlignment = 'top';
app.text12.WordWrap = 'on';
app.text12.FontSize = 16;
app.text12.FontWeight = 'bold';
app.text12.FontColor = [0 0 1];
app.text12.Position = [49.8 19.4 492.8 24];
app.text12.Text = 'La integral aproximada es:
';

% Show the figure after all components are
%created

```

```

        app.figure1.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = metodosimpson38_App(varargin)

        runningApp = getRunningApp(app);

        % Check for running singleton app
        if isempty(runningApp)

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.figure1)

            % Execute the startup function
            runStartupFcn(app,
                @(app)metodosimpson38_OpeningFcn(app,
                    varargin{:}))
        else

            % Focus the running singleton app

```

```
        figure(runningApp.figure1)

        app = runningApp;
    end

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.figure1)
end
end
end
```



Recibo digital

Este recibo confirma que su trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: Sandoval Tarrillo Xintia Haiduk, Lazo Enriquez Lina Elizabeth
Título del ejercicio: Quick Submit
Título de la entrega: Ejecutable de Interfaz Gráfica de Usuario en App Designer de ...
Nombre del archivo: TESIS_XINTIA-LINA_2025.pdf
Tamaño del archivo: 5.78M
Total páginas: 153
Total de palabras: 20,966
Total de caracteres: 110,883
Fecha de entrega: 09-dic-2025 01:34p. m. (UTC-0500)
Identificador de la entrega: 2841452786



Dr. Cuti Gutiérrez Alcides Raúl
Departamento de Matemática
Asesor

Ejecutable de Interfaz Gráfica de Usuario en App Designer de Matlab para la Integración Numérica

INFORME DE ORIGINALIDAD

15%	13%	6%	10%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad Nacional Pedro Ruiz Gallo		4%
	Trabajo del estudiante		
2	gredos.usal.es		2%
	Fuente de Internet		
3	repositorio.unprg.edu.pe		1%
	Fuente de Internet		
4	Submitted to Asia Pacific University College of Technology and Innovation (UCTI)		1%
	Trabajo del estudiante		
5	stackoverflow.com		1%
	Fuente de Internet		
6	dokumen.pub	 Dr. Cuti Gutiérrez Alcides Raúl Departamento de Matemática Asesor	1%
	Fuente de Internet		
7	dspace.unitru.edu.pe		1%
	Fuente de Internet		
8	www.mathworks.com		<1%
	Fuente de Internet		

9	Submitted to Florida International University Trabajo del estudiante	<1 %
10	idoc.pub Fuente de Internet	<1 %
11	www.codenong.com Fuente de Internet	<1 %
12	okeanis.lib.teipir.gr Fuente de Internet	<1 %
13	Ruiz Lizama, Edgar Cruz. "Integralab: Un software para integracion de funciones y solucion de ecuaciones diferenciales por metodos numericos.", Pontificia Universidad Catolica del Peru - CENTRUM Catolica (Peru), 2020 Publicación	<1 %
14	repositorio.uss.edu.pe Fuente de Internet	<1 %
15	Submitted to University of Pretoria Trabajo del estudiante	<1 %
16	dspace.esPOCH.edu.ec Fuente de Internet	<1 %
17	repositorio.upct.es Fuente de Internet	<1 %
18	Camargo Caysahuana, Andres. "Aplicación de la metodología BIM para mejorar el contenido	<1 %



Dr. Cuti Gutiérrez Alcides Raúl
Departamento de Matemática
Asesor

de aprendizaje en estudiantes de Ingeniería Civil de la Universidad Católica Los Ángeles de Chimbote - 2021", Universidad Católica los Ángeles de Chimbote (Peru)

Publicación

19 Submitted to University of Osijek - Faculty of Electrical Engineering, Computer Science and Information Technology <1 %

Trabajo del estudiante

20 Pedro Madrid. "Acerca del Orden de Convergencia de las Reglas de Integración del Trapecio y Simpson para Cierta Clase de Funciones No Diferenciables", Revista de la Escuela de Física, 2023 <1 %

Publicación

Dr. Cuti Gutiérrez Alcides Raúl
Departamento de Matemática
Asesor

21 vbook.pub <1 %

Fuente de Internet

22 Guillermo Albuja-Proaño, Miguel Murillo. "A generalization of the Newton-Kantorovich method", Revista de la Academia Colombiana de Ciencias Exactas, Físicas y Naturales, 2025 <1 %

Publicación

23 Submitted to University of Northampton <1 %

Trabajo del estudiante

24 silo.pub <1 %

Fuente de Internet

25 hdl.handle.net
Fuente de Internet

<1%

26 repositorij.unios.hr
Fuente de Internet

<1%

Excluir citas

Activo

Excluir coincidencias

< 15 words

Excluir bibliografía

Activo



Dr. Cuti Gutiérrez Alcides Raúl
Departamento de Matemática
Asesor